ISSN 2007-9737

Leveraging Machine Learning to Unveil the Critical Role of Geographic Factors in COVID-19 Mortality in Mexico

Christian E. Maldonado-Sifuentes[1], Mariano Vargas-Santiago[*,1], Diana A. Leon-Velasco[2],
M. Cristina Ortega-García[3], Yoel Ledo-Mezquita[2], Francisco A. Castillo-Velasquez[4]

[1] Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCYT),
Mexico

[2] Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM),
Campus Ciudad de México,
Mexico

[3] Transdisciplinary Research for Augmented Innovation Laboratory (TRAI-L),
Mexico

[4] Universidad Politécnica de Querétaro,
Mexico

{christian.maldonado,mariano.vargas}@conahcyt.mx, assaely.leon@tec.mx,
cristina.ortega@trai-l.com, yledo@tec.mx, francisco.castillo@upq.edu.mx

Abstract. In this paper, we present an in-depth analysis leveraging several renowned machine learning techniques, including Snap Random Forest, XGBoost, Extra Trees, and Snap Decision Trees, to characterize comorbidity factors influencing the Mexican population. Distinct from existing literature, our study undertakes a comprehensive exploration of algorithms within a defined search space, conducting experiments ranging from coarse to fine granularity. This approach, coupled with machine learning-driven feature enhancement, enables us to deeply characterize the factors most significantly affecting COVID-19 mortality rates within the Mexican demographic. Contrary to other studies, which obscure the identification of primary factors for local populations, our findings reveal that geographical factors such as residence location hold greater significance than even comorbidities, indicating that socioeconomic factors play a pivotal role in the survival outcomes of the Mexican population. This research not only contributes to the targeted understanding of COVID-19 mortality drivers in Mexico but also highlights the critical influence of socioeconomic determinants, offering valuable insights for public health strategies and policy formulation.

Keywords. Diabetes, COVID-19, machine learning, SARS CoV-2, Cox, RMST.

1 Introduction

The advent of COVID-19 has instigated a global health crisis of unparalleled magnitude, prompting a concerted effort across healthcare systems worldwide to counteract its ramifications [2, 10, 12]. This pandemic has underscored the critical need for advanced medical research and data analytics to dissect and mitigate the virus's impacts efficiently. Central to this effort is the analysis of vast datasets to identify patterns and predictors of COVID-19 outcomes, with particular emphasis on the significance of patient comorbidities and geographic statistics in influencing mortality rates [2, 7, 8, 10, 12].

This scenario has propelled the development of an Automated Machine Learning (AutoML) framework, crafted to harness the latest in machine learning innovation to expedite the evaluation of COVID-19 mortality risks. By optimizing the model development process, AutoML aims to enrich our comprehension of the medical and societal variables influencing

doi: 10.13053/CyS-28-1-4908

COVID-19 mortality, offering crucial insights to both healthcare practitioners and researchers. The dynamic and evolving nature of COVID-19 data renders the adaptability and automation features of AutoML exceptionally valuable. Such capabilities facilitate rapid algorithmic adjustments and hyperparameter optimization to assimilate new findings and data, establishing AutoML as an essential asset in combating COVID-19.

Our investigation leverages a comprehensive dataset provided by the Mexican Federal Government, chronicling the pandemic's impact on the Mexican populace from January 1, 2023, to August 8, 2023. This dataset encompasses detailed information on 1,021,380 patients, including demographic, clinical outcomes, and mortality data, thereby offering a unique lens through which to examine the multifaceted influences on COVID-19 mortality.

Utilizing various validated methodologies for COVID-19 diagnosis, including antigen testing and clinical epidemiological association, our study utilizes AutoML to dissect this dataset, aiming to unearth pivotal patterns and predictors of mortality. This endeavor aligns with the urgent global requirement for innovative analytical tools capable of pacing with the swiftly evolving pandemic landscape, marking a significant stride in applying AutoML for comprehensive data analytics in confronting the COVID-19 health crisis.

Emerging studies highlight the utility of machine learning in scrutinizing COVID-19 data and AutoML's potential to revolutionize this analysis by enhancing the accessibility and adaptability of advanced data analytics [5, 9, 11]. Building upon these insights, our research endeavors to offer valuable perspectives on the determinants of COVID-19 mortality, showcasing AutoML's utility in pandemic response and preparedness. This paper makes the following contributions:

1. We establish a comprehensive experimental framework that divides into two core components: traditional statistical analysis and a machine learning-based approach. This framework, detailed in Sections 4.1 and 4.2 for statistical analysis and Section 4.4 for machine learning, facilitates a nuanced exploration of the comorbidity factors affecting COVID-19 mortality in the Mexican population.

2. Through our rigorous analysis employing cutting-edge machine learning techniques—specifically Snap Random Forest, XGBoost, Extra Trees, and Snap Decision Trees—we provide an in-depth characterization of the comorbidity factors influencing COVID-19 mortality.

   Our study distinguishes itself by executing a comprehensive exploration of algorithms within a defined search space, ranging from coarse to fine granularity. This method, enhanced by machine learning-driven feature selection, allows for a deepened understanding of the critical factors affecting mortality rates.

3. Contrary to prevailing studies that predominantly focus on comorbidities as the key mortality determinants, our findings reveal the greater significance of residential location, pointing to socioeconomic factors as pivotal in determining survival outcomes in the Mexican context.

   This novel insight emphasizes the need for public health strategies and policy formulation to consider socioeconomic determinants alongside medical factors.

4. By leveraging a combination of traditional statistical and modern machine learning methodologies, our research contributes a unique perspective to the body of knowledge.

   It not only provides a targeted analysis of COVID-19 mortality drivers in Mexico but also underscores the crucial influence of socioeconomic factors on health outcomes. Our study paves the way for informed public health interventions aimed at reducing mortality within socioeconomically diverse populations.

The rest of this work is organized as follows: Section 2 outlines the methodologies employed in our study and presents the dataset we are using. Section 3 discusses some of the related work to our study. Section 4 describes the experimental setup for our study, which is divided into two subsections statistical or traditional analysis 4.2 and machine learning based analysis 4.4.

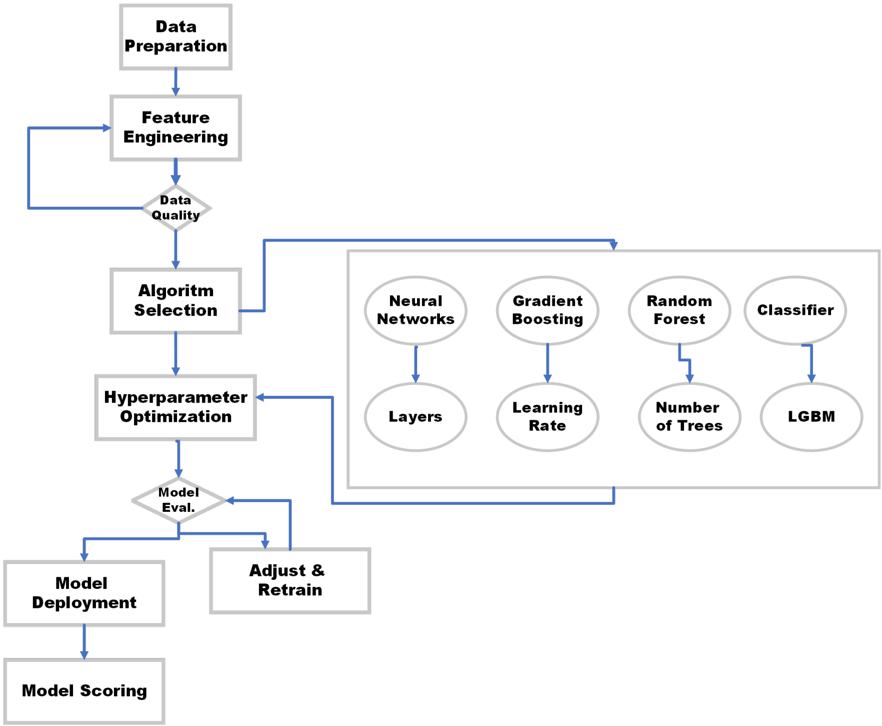


Fig. 1. Experimental setup for machine learning based analysis

Section 5 presents the findings from our experiments, for traditional statistical methods and those after using the machine Learning based analysis. Finally, conclusions and future work are given in Section 6.

2 Methods

This retrospective study is predicated on data from the Mexican Federal Government, endorsed by the Epidemiological Surveillance System for Viral Respiratory Diseases under the Mexican Ministry of Health's purview. Ethical consent for this data's utilization was comprehensively secured from the pertinent health ethics committees.

2.1 Dataset

Employing the COVID-19 Mexican Patients Dataset, our study scrutinizes the demographic profiles, characteristics, and clinical outcomes of the Mexican demographic amid the COVID-19 pandemic. This dataset, curated and disclosed by the Mexican Federal Government and the Ministry of Health, spans January 1, 2023, to August 8, 2023. It aggregates data from 475 Viral Respiratory Disease Monitoring Units, detailing individuals hospitalized following a positive COVID-19 test, totaling 1,021,380 patients with comprehensive mortality data.

2.2 COVID-19 Determination

COVID-19 diagnoses were ascertained through SARS-CoV-2 antigen detection via nasal swabs, conducted across various government-affiliated surveillance and healthcare establishments. The validation of COVID-19 cases employed three methodologies: clinical epidemiological association, a deliberation committee's verdict, or antigen testing. Conversely, a negative result signified the antigen's absence in the sample.

2.2.1 Repeatability Across Databases

Our methodology, characterized by its repeatability across diverse healthcare settings, benefits from the flexible and user-oriented nature of codeless platforms like Uber Ludwig, which significantly enhance and simplify the creation of Linux shell level scripting to allow for automation of ML tasks. This approach incorporates insights from recent studies on machine learning in COVID-19 analysis, spanning techniques from functional and sentiment analysis to causal learning and mental health data examination. Such breadth in machine learning application highlights AutoML's potential to navigate the intricate dynamics of COVID-19 mortality influences adeptly.

Table 1. Patient Demographics and Covariates

	Diabetic	Non-diabetic	Total Individuals
Total Individuals	80,346	940,035	1,021,380
Male	30,533 (38%)	391,404 (45%)	422,344 (41%)
Female	49,813 (62%)	548, 631 (58%)	599,036 (59%)
Native	80,199 (8%)	935,950 (92%)	1,017,140 (99%)
Diabetes	80,346 (100%)	0 (0%)	80,346 (8%)
Hypertension	44,851 (56%)	67,248 (7%)	112,151(11%)
Obesity	15,989 (20%)	61,978 (7%)	78,002 (8%)
Smoking	5,007 (6%)	36,773 (4%)	41,798 (4%)
Pneumonia	6,241 (8%)	22,061 (2%)	28,373 (2%)
ICU	508 (0.7%)	2,181 (0.23%)	2,704 (0.26%)
Intubation	809 (1%)	2,607 (0.28%)	3,436 (0.33%)
Death	2,198 (2.7%)	4,371 (0.46%)	6,581 (0.64%)

3 Related Work

The onset of COVID-19 has precipitated a flux of research employing machine learning to elucidate the virus's outcomes and impacts. Our study aligns with and extends this corpus of work by emphasizing the Mexican demographic and integrating socioeconomic variables into our analysis, distinguishing our research within the burgeoning field of machine learning applications in COVID-19 analysis.

Several recent studies have employed machine learning models to predict COVID-19 outcomes based on patient data. For example, He et al. [3] developed a generalizable and easy-to-use COVID-19 severity stratification model utilizing immune-phenotyping and machine learning, underscoring the importance of a comprehensive approach in determining patient outcomes.

Similarly, Badiola-Zabala et al. [1] conducted a systematic literature review of clinical decision support approaches during the pandemic, highlighting the effectiveness of ML- and AI-based models in predicting mortality among COVID-19 patients.

Moreover, Lages dos Santos et al. [4] provided a comparative analysis of machine learning algorithms for predicting COVID-19 mortality in children and adolescents using a large public dataset in Brazil, indicating the predictive power of various factors, including demographic data and comorbidities.

These studies exemplify the significant potential of machine learning in enhancing COVID-19 prognosis and management through the analysis of patient data. In contrast to these studies, our research expands the scope of analysis by employing advanced techniques such as Snap Random Forest, XGBoost, Extra Trees, and Snap Decision Trees. Our methodological approach involves a more nuanced exploration of the algorithmic search space, allowing for a detailed characterization of the influences on COVID-19 mortality.

This granularity surpasses the typical scope of existing literature by refining feature selection and optimization processes to better capture the complex interplay of factors affecting mortality rates.

Furthermore, while most studies concentrate on medical and biological predictors, our investigation reveals the paramount importance of residential location as a determinant of COVID-19 mortality in the Mexican context.

This finding aligns with research by Mendez-Astudillo [6], which points to socioeconomic factors and economic inequalities as critical determinants of health outcomes during the pandemic, underscoring the influence of social determinants on public health.

Our work contributes a unique perspective by integrating socioeconomic considerations with comorbidity analysis, thereby offering a more holistic understanding of the drivers behind COVID-19 mortality. This approach not only fills a gap in the current literature but also serves as a foundation for future research and policy-making aimed at mitigating the impacts of the pandemic on vulnerable populations.

By examining these recent works in conjunction with our study, it becomes evident that while there is a consensus on the importance of comorbidities and demographic data in predicting COVID-19 outcomes, the role of socioeconomic factors, particularly in the context of Mexico, remains underexplored. Our research aims to bridge this gap, offering insights into the significance of residential location and socioeconomic status in shaping COVID-19 mortality rates.

4 Experimental Setup

The experimental framework is delineated into two principal components. Initially, we delineate the methodology referred to as traditional statistical analysis, as depicted in Sections 4.1 and 4.2. Subsequently, the foundational setup for the analysis predicated on machine learning techniques is presented, as specified in Section 4.4. This investigation constituted a retrospective study that drew upon data sourced from the Mexican Federal Government. The data set used in this study had been publicly disseminated and subjected to validation procedures by the Epidemiological Surveillance System for Viral Respiratory Diseases under the auspices of the Mexican Ministry of Health.

Ethical approval for the use of this data set was obtained in full from the ethics committees associated with the Ministry of Health

4.1 Determination of COVID-19

The diagnosis of COVID-19 was established by detecting the SARS-CoV-2 antigen through nasal swab testing. This diagnostic procedure was conducted at various surveillance and healthcare facilities under the jurisdiction of the Mexican Government, with readily available results.

We utilize three distinct approaches to validate positive COVID-19 cases, which encompass the following: validation via clinical epidemiological association, validation through a deliberation committee, or validation through antigen testing. On the contrary, a negative status indicated the absence of this antigen in the tested samples.

4.2 Statistical Analysis

The demographic and diabetes-related characteristics of individuals testing positive for the SARS-CoV-2 antigen were subjected to analysis employing descriptive statistical methods. Comparative assessments among patients, taking into account relevant covariates, were performed using $T-$tests and X^2 tests. The primary endpoint under investigation was patient survival, defined as the duration from the onset of COVID-19 symptoms to the point of mortality, with censoring applied at the final enrollment date for adult COVID-19 patients admitted to hospitals.

To estimate the survival curve, Kaplan-Meier curves were generated and the statistical significance of the survival durations between hospitalized adult patients with and without diabetes was assessed using the log-rank test. A Cox proportional hazards model was used to calculate the hazard ratio and establish a confidence interval (CI) 95% to gauge the effect of treatment.

All statistical tests were two-sided, and $p-$ value less than 0.05 was considered indicative of statistical significance. In addition to overall survival analysis, the calculation of restricted mean survival time (RMST) was executed for both diabetic and non-diabetic adult COVID-19 patients admitted to hospitals, following propensity matching to mitigate the impact of confounding variables. This approach entailed fitting a parametric survival model to the dataset to estimate the mean survival time for the two distinct groups under investigation.

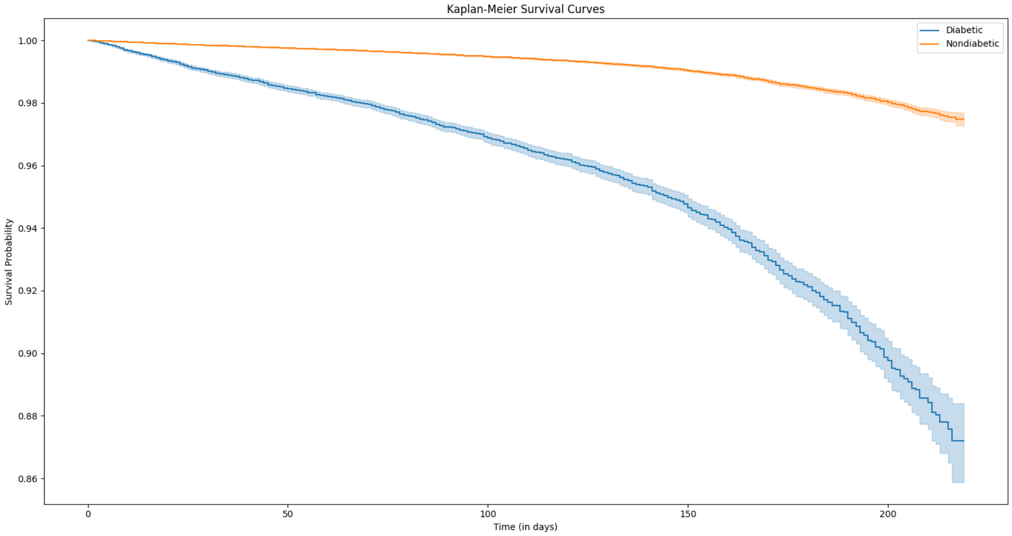


Fig. 2. Multi-variable Kaplan–Meier survival plots comparing diabetic and non-diabetic patients

4.3 Variables

The objective of this study is to investigate the influence of diabetes on the prognosis of COVID-19 in Mexican patients, specifically focusing on the likelihood of in-hospital mortality among those with diabetes. This research constitutes a substantial and nationwide retrospective cohort study, which is poised to offer valuable information on the interplay between diabetes and the outcomes of COVID-19. Such insights have the potential to guide the formulation of effective mitigation strategies and improve the patient triage process.

Key demographic information, including sex, age, country of origin, pre-existing health conditions (such as hypertension, diabetes,obesity and immunosuppression), smoking habits, and pregnancy status, was systematically documented for each individual. Data related to COVID-19 status included records of antigen test results and antigen sample collection. It is worth noting that during their hospitalization, no publicly accessible information was disclosed regarding the patients' clinical progression.

Table 2. Confusion Matrix

Observed	**Predicted**	
	Dead	Alive
Dead	**559**	**93**
Alive	**141**	**936**
% Correct	Overall % Correct: 86.5%	

4.4 Machine Learning Experimental Setup

The experimental setup for our machine learning model development consists of a multi-stage process, as illustrated in the figure 1. The procedure is initiated with Data Preparation, where raw data is collected and pre-processed to ensure it is in a suitable format for analysis. This stage is crucial for the subsequent steps as it directly affects the quality of the insights derived from the data.

Following this, we engage in Feature Engineering, which involves creating new features from the existing data to improve the model's predictive power. This step also includes assessing Data Quality to ensure the integrity and appropriateness of the data for the machine learning algorithms. The next phase is Algorithm Selection, where we choose appropriate machine learning algorithms based on the nature of the data and the problem statement.

This decision impacts the model's ability to learn from the data and make accurate predictions. After selecting the algorithms, Hyperparameter Optimization is conducted to find the optimal settings for each algorithm, enhancing the model's performance. This involves tuning various parameters that govern the learning process of the models.

Once the models are trained with the best hyperparameters, Model Evaluation is performed using appropriate metrics to assess their performance. If the models do not meet the desired performance criteria, they are subject to Adjustment and Retraining to improve their accuracy and reliability. Upon achieving satisfactory evaluation metrics, the model is then moved to Model Deployment, where it is integrated into the production environment to make predictions on new data.

This step is critical for translating the model's capabilities into practical applications. Lastly, Model Scoring is performed on the deployed model, where it is continuously monitored and scored based on its performance in the live environment. This ensures that the model remains accurate and relevant over time.

Within the Algorithm Selection stage, a range of machine learning algorithms is considered. These include Neural Networks, with a focus on the configuration of their Layers; Gradient Boosting methods, with an emphasis on the Learning Rate; Random Forest, where the Number of Trees is a significant parameter; and a generic Classifier, which, in this context, appears to be specified as LGBM (Light Gradient Boosting Machine), a type of gradient boosting framework [1].

5 Results

The results section is systematically organized to reflect the bifurcated approach of the experimental framework. Initially, outcomes stemming from the traditional statistical analysis are elucidated, corresponding to the methodologies outlined in Sections 4.1 and 4.2, refer to 5.1. Subsequently, we present the findings derived from the machine learning-based analysis, adhering to the foundational setup delineated in Section 4.4, refer to 5.2.

This sequential presentation facilitates a comprehensive understanding of the experimental results, allowing for a direct comparison between traditional statistical methodologies and modern machine learning approaches in addressing the research objectives.

5.1 Results for Traditional Statistical Methods

In Table 1, we present a comprehensive summary of demographic variables and relevant covariates for people who tested positive for COVID-19, using data sourced from the Mexican Patient Data Set, which is publicly available through the Mexican Ministry of Health.

This study covers a total of 1,021,380 adult patients who were hospitalized due to COVID-19, all of whom have complete records of their mortality outcomes. Within this patient cohort, 38% were male, while 62% were female, with an average age of 38.41 years (standard deviation = 19.50). The main comorbidities prevalent in this population included hypertension (11%), diabetes (7.9%), and obesity (8%), while 4% were identified as smokers. It is noteworthy that the term ”Nondiabetic” is used to describe individuals without a diagnosis of diabetes, as no cases of diabetes were identified within this group.

These demographic and clinical characteristics provide a foundational understanding of the patient population under investigation in this study. Figure 2 shows the Kaplan-Meier survival graphs, presenting a comparative analysis of survival probabilities among two groups: COVID-19 patients with and without diabetes. The study encompasses a comprehensive cohort of 1,021,380 individuals, among which 6,581 individuals experienced a fatal outcome related to the disease.

Statistical analysis, specifically the log-rank test, revealed a statistically significant disparity in survival rates between these two groups of

[1] Project files can be found at https://github.com/christianemaldonadomti/MLCovidMexico

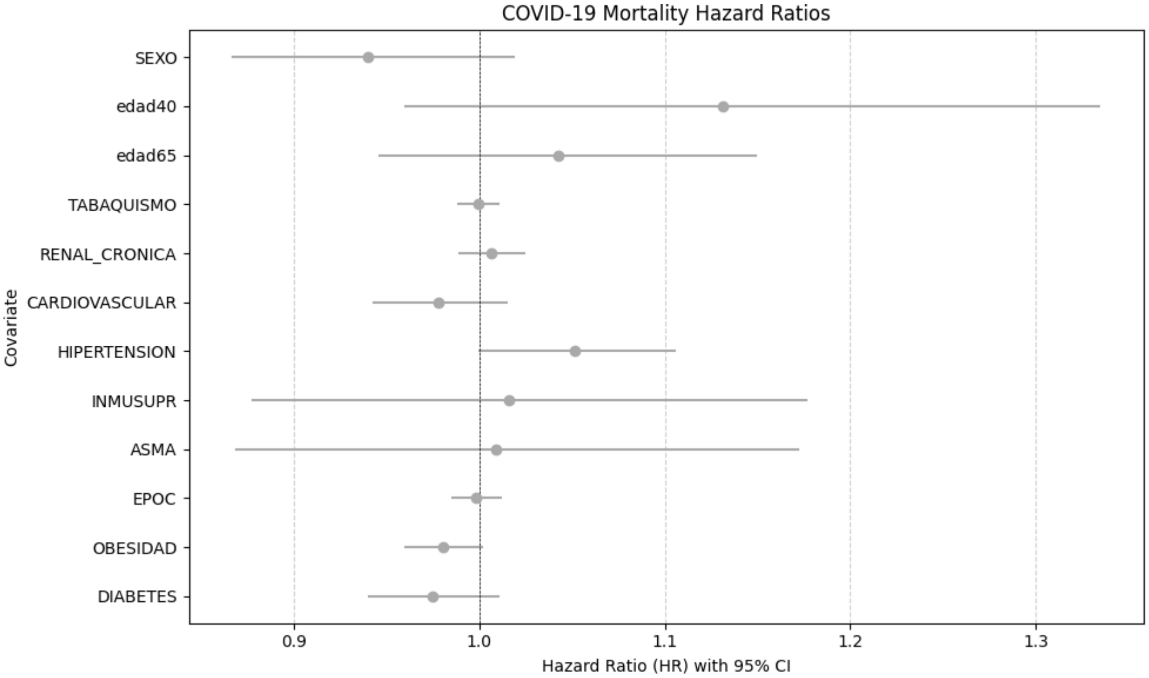


Fig. 3. COVID-19 Cox Mortality Hazard Ratios without location

hospitalized COVID-19 patients (p ¡ 0.01). It is noteworthy that the depicted survival curves adhere to the assumption of proportional hazards, as they do not intersect within the examined time frame.

Although the dataset presents a higher mortality rate among individuals without diabetes, those afflicted with this condition exhibit an acceleration in the progression towards mortality. In other words, they experience a shorter survival time compared to individuals with diabetes.

This phenomenon can be substantiated by referring to Figure 2. Figure 3 depicts the utilization of the Cox proportional hazards model to assess the impact of various covariates on the risk of mortality, with statistical significance determined through the examination of p-values. Our analytical findings yield noteworthy insights into the relationship between comorbidities and the risk of COVID-19 mortality. Specifically, individuals with diabetes displayed a modestly reduced hazard of mortality (Hazard Ratio: 0.975) in comparison to those without diabetes, although this reduction did not attain statistical significance (p-value: 0.167). Similarly, obesity was associated with a slight reduction in mortality hazard (Hazard Ratio: 0.981), although this reduction was only marginally significant (p-value: 0.080).

Table 3. Model Evaluation Measures

Measures	Holdout sc.	X-validation sc.
Accuracy	0.865	0.864
	0.876	0.876
Precision	0.799	0.798
	0.807	0.806
Recall	0.857	0.857
	0.883	0.882
F1	0.827	0.827
	0.843	0.843
Avg. precision	0.892	0.892
	0.904	0.904

Conversely, hypertension was linked to a minor increase in mortality hazard (Hazard Ratio: 1.051), with a p-value that approached statistical significance at 0.051. Other covariates, including

Chronic Obstructive Pulmonary Disease (COPD), asthma, immunosuppression, cardiovascular conditions, renal chronic conditions, smoking habits, various age categories, and gender, did not demonstrate statistically significant effects on the risk of COVID-19 mortality.

5.2 Results for Machine Learning Based Analysis

Our analysis's effectiveness was quantitatively assessed using a confusion matrix, which provides insights into the model's predictive accuracy by comparing actual outcomes against predictions. Table 2 presents the confusion matrix derived from the model evaluation.

The confusion matrix reveals that out of the cases predicted to result in mortality (Dead), 559 were correctly identified (true positives), while 93 were misclassified (false negatives). Conversely, for the cases predicted to result in survival (Alive), 936 were accurately predicted (true negatives), with 141 instances being incorrectly forecasted as mortalities (false positives). This performance yields an overall prediction accuracy of 86.5%, demonstrating the models' robust ability to discern between survival and mortality outcomes based on the assessed comorbidities. Such a high level of accuracy underscores the potential of employing these machine learning techniques for predictive purposes in medical settings, specifically in prognosticating COVID-19 outcomes.

The differential performance across models, as inferred from the confusion matrix, emphasizes the nuanced understanding these algorithms provide regarding the critical factors affecting mortality rates. Notably, the high overall accuracy achieved across different models suggests that machine learning-driven feature selection significantly contributes to identifying the most impactful predictors of COVID-19 mortality.

Moreover, the detailed evaluation of model performance through various metrics highlights the robustness and reliability of the employed machine learning methodologies. The consistent predictive precision across models indicates the efficacy of the selected algorithms in capturing the complexities inherent in the comorbidity factors of COVID-19 patients.

We delve into the results obtained from our comprehensive analysis, which leverages advanced machine learning techniques to illuminate the intricate dynamics of comorbidity factors affecting COVID-19 mortality rates. The algorithms employed—namely, Snap Random Forest, XGBoost, Extra Trees, and Snap Decision Trees—were rigorously evaluated to ensure a robust exploration of the predictive capabilities pertaining to COVID-19 mortality.

The evaluation metrics presented in Table 3 encapsulate the performance of these models across various dimensions, including accuracy, precision, recall, F1 score, and average precision. These metrics were assessed through both holdout and cross-validation methods to validate the models' consistency and reliability. Our analysis showcased the models' commendable performance in characterizing the comorbidity factors influencing COVID-19 mortality, as evidenced by the evaluation scores tabulated in Table 3. The accuracy of the models, as denoted by the holdout score (Holdout sc.) and cross-validation score (X-validation sc.), was observed to be consistently high, with accuracy scores reaching up to 0.876. This high level of accuracy underscores the effectiveness of the machine learning techniques applied in capturing the complexities inherent in COVID-19 mortality risk factors.

Precision, a measure of the models' ability to correctly identify positive instances among the predicted positives, also demonstrated high performance, with scores up to 0.807. This indicates a significant strength in the models' capability to discern true cases of high mortality risk amidst a plethora of potential predictors.

Recall scores, which reflect the models' capacity to identify all relevant instances, were notably high as well, reaching up to 0.883. This suggests that the models are exceptionally adept at capturing the majority of significant cases, thereby reducing the risk of overlooking critical comorbidity factors.

The F1 score, a harmonic mean of precision and recall, further solidifies the models'

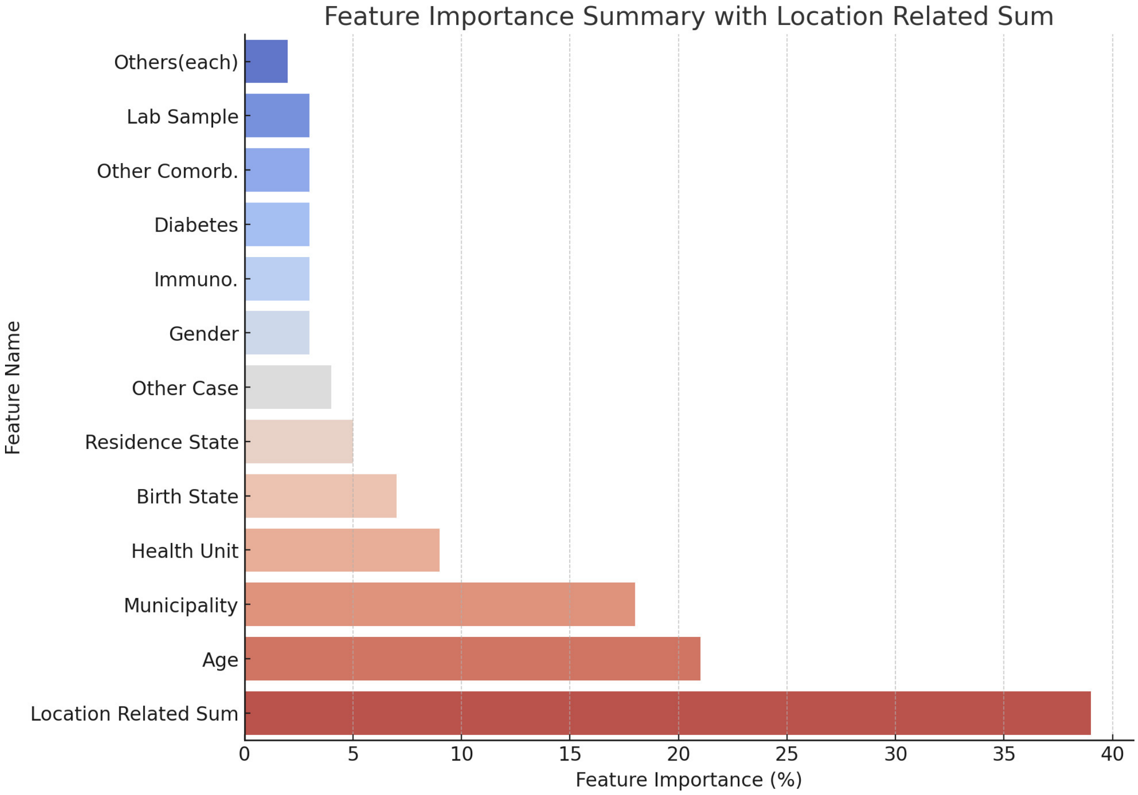


Fig. 4. Leveraging Machine Learning to Unveil the Critical Role of Geographic and Sociodemographic Factors in COVID-19 Mortality Across Mexico

robustness, with scores peaking at 0.843. This balance between precision and recall illustrates the models' overall efficacy in identifying true positives while minimizing false negatives and positives.

Moreover, the average precision score, which provides an aggregate measure of precision across varying thresholds, reached an impressive 0.904. This underscores the models' outstanding precision-recall balance, a critical aspect in the context of medical predictive modeling, where the cost of false negatives can be particularly high.

The performance metrics presented affirm the substantial capability of the employed machine learning models to discern the critical comorbidity factors influencing COVID-19 mortality. The high scores across all evaluated measures attest to the models' precision, recall, and overall accuracy, highlighting their potential utility in aiding public health efforts by providing deepened insights into COVID-19 mortality risk factors.

This detailed exposition not only underscores the predictive prowess of the deployed models but also emphasizes the significance of machine learning-driven feature selection in enhancing our understanding of the key determinants of COVID-19 mortality rates. The findings elucidated herein lay a solid foundation for further research into predictive modeling for infectious diseases, potentially guiding more targeted and effective public health interventions.

5.3 Feature Importance Analysis

A critical component of our study involved the assessment of feature importance, which highlights the relative impact of various factors on the predictive models' outcomes.

Table 4. Feature summary

Feature Name	**Feature Importance**
- Municipality	18.00%
- Health Unit	9.00%
- Birth State.	7.00%
- Residence State.	5.00%
Sum Loc. Related	**39.00%**
Age	21.00%
Other Case	4.00%
Lab Sample	3.00%
Other Comorb.	3.00%
Diabetes	3.00%
Immuno.	3.00%
Gender	3.00%
Others(each)	less than 2.00%

Table 4 showcases the summarized results of this analysis, revealing the percentage contribution of each feature towards the model's predictive capability.

Fig. 4 clearly underscores the paramount importance of geographic and sociodemographic factors in influencing mortality rates in Mexico. As delineated through the comprehensive analysis of various features, it is evident that the collective weight of location-based attributes—encompassing Municipality, Health Unit, Birth State, and Residence State—surpasses even the influence of age, traditionally considered one of the most significant predictors of mortality risk.

This revelation not only highlights the geographical variance within the country but also brings to light the profound impact of sociodemographic elements on health outcomes.

The overarching dominance of geographical factors in determining mortality rates suggests a complex interplay between environmental, economic, and social determinants of health. In Mexico, disparities in healthcare access, differences in environmental exposure, and varying socioeconomic conditions across different regions amplify the mortality risk associated with geographical and sociodemographic characteristics.

The aggregated importance of location-related variables, serves as a compelling testament to the critical need for targeted public health strategies and interventions that are finely tuned to the unique challenges and vulnerabilities of each region. This approach is crucial for mitigating the risks associated with these factors and for paving the way toward more equitable health outcomes across the diverse landscapes of Mexico.

Age emerged as the most significant predictor of COVID-19 mortality, accounting for 21.00% of the model's predictive power. This finding underscores the heightened vulnerability of older populations to severe outcomes following COVID-19 infection.

Following closely, Municipality with an 18.00% importance, indicates the significance of geographical and possibly socio-economic factors in mortality risk. The Health Unit and Birth State features also demonstrate considerable influence, with contributions of 9.00% and 7.00%, respectively, highlighting the role of healthcare access and regional health disparities.

Further down the list, Residence State, Other Case, Lab Sample, and various comorbidities including Diabetes and conditions affecting the immune system (Immuno.), each contribute to the model's ability to predict mortality, albeit to a lesser extent. Notably, each of these features adds valuable insights into the complex interplay between patient characteristics, healthcare system factors, and comorbid conditions in determining COVID-19 outcomes.

Gender, with a 3.00% contribution, and other less impactful features (Others(each)) accounting for less than 2.00% each, further delineate the nuanced landscape of risk factors associated with COVID-19 mortality. This granularity in feature importance not only highlights the predominant role of certain demographics and health-related factors but also underscores the multifaceted nature of COVID-19's impact on various segments of the population.

5.4 Discussion

The derived feature importance from our analysis illuminates the complex and multifactorial nature of COVID-19 mortality risk. Age stands out as a critical determinant, corroborating global observations regarding the disproportionate impact of the virus on older individuals.

The significant role of geographic and healthcare accessibility factors further emphasizes the need for targeted public health interventions and resource allocation to mitigate mortality risks effectively.

The granularity achieved through our machine learning-driven exploration enables a more nuanced understanding of the interactions between various factors and COVID-19 mortality. Such insights are invaluable for guiding clinical decision-making, optimizing healthcare resource distribution, and tailoring public health strategies to address the needs of the most vulnerable populations.

Our study's findings contribute a novel perspective to the ongoing discourse on COVID-19 mortality, providing a comprehensive analysis of comorbidity factors that influence outcomes. This enhanced understanding is pivotal for informing future research, policy-making, and clinical practices aimed at reducing the mortality burden of the pandemic.

6 Conclusion and Future Work

In our investigation, it was discovered that the demographic and socioeconomic areas significantly influence mortality rates due to COVID-19. This finding diverges from common assumptions that comorbidities alone are the primary determinants of mortality outcomes in the Mexican population. Through an exhaustive analysis utilizing advanced machine learning techniques, including Snap Random Forest, XGBoost, Extra Trees, and Snap Decision Trees, our study delved into the myriad factors impacting COVID-19 mortality.

Unlike prior research that often fails to discern the paramount factors affecting mortality rates in localized populations, our approach allowed for a nuanced exploration of the interplay between various determinants.

Our comprehensive examination extended across algorithms within a defined search space, implementing experiments with varying degrees of granularity. This methodology, augmented by machine learning-driven feature enhancement, facilitated a deepened understanding of the elements most critically affecting COVID-19 mortality rates.

The findings underscore the predominance of residential location over comorbidities in determining mortality outcomes, pointing to the substantial role of socioeconomic factors in influencing survival chances. This revelation underscores the necessity for public health strategies and policy-making to prioritize socioeconomic determinants alongside medical considerations in combatting COVID-19 mortality.

The implications of our research are twofold: firstly, it contributes to a more targeted comprehension of the drivers behind COVID-19 mortality in the Mexican context; and secondly, it accentuates the vital impact of socioeconomic conditions on health outcomes. By highlighting these insights, our study provides a foundational basis for the development of informed and effective public health interventions aimed at mitigating COVID-19 mortality within socioeconomically diverse populations.

Acknowledgments

The authors wish to express their profound gratitude to the Consejo Nacional de Ciencias, Humanidades y Tecnología (CONAHCYT), acknowledging the institution's significant support, as two of the authors are esteemed affiliated researchers within this esteemed organization. It is recognized that the synergistic contributions from all involved entities have substantially enhanced the quality and validity of the present study.

References

1. **Badiola-Zabala, G., Lopez-Guede, J. M., Estevez, J., Graña, M. (2024).** Machine learning first response to COVID-19: A systematic literature review of clinical decision assistance approaches during pandemic years from 2020 to 2022. Electronics, Vol. 13, No. 6. DOI: 10.3390/electronics13061005.

2. **Balakrishnan, K. N., Yew, C. W., Chong, E. T. J., Daim, S., Mohamad, N. E., Rodrigues, K., Lee, P. C. (2023).** Timeline of SARS-CoV-2 transmission in Sabah, Malaysia: Tracking the molecular evolution. Pathogens, Vol. 12, No. 8, pp. 1047. DOI: 10.3390/pathogens12081047.

3. **He, X., Cui, X., Zhao, Z., Zhang, H., Ge, Q., Leng, Y. (2024).** A generalizable and easy-to-use COVID-19 stratification model for the next pandemic via immune-phenotyping and machine learning. Frontiers in Immunology, Vol. 15, pp. 1372539. DOI: 10.3389/fimmu.2024.1372539.

4. **Lages-dos-Santos, A., Oliveira, M., Colosimo, E. A., Pinhati, C., Galante, S. C., Martelli-Júnior, H., Simões Silva, A. C., Oliveira, E. (2024).** Comparative analysis of machine learning algorithms for predicting COVID-19 mortality in children and adolescents using a large public dataset in Brazil. Social Science Research Network. DOI: 10.2139/ssrn.4740297.

5. **Latif, S., Usman, M., Manzoor, S., Iqbal, W., Qadir, J., Tyson, G., Castro, I., Razi, A., Boulos, M. N., Weller, A., Crowcroft, J. (2020).** Leveraging data science to combat COVID-19: A comprehensive review. IEEE Transactions on Artificial Intelligence, Vol. 1, No. 1, pp. 85–103. DOI: 10.1109/TAI.2020.3020521.

6. **Mendez-Astudillo, J. (2024).** The impact of comorbidities and economic inequality on COVID-19 mortality in Mexico: A machine learning approach. Frontiers in Big Data, Vol. 7. DOI: 10.3389/fdata.2024.1298029.

7. **Padilla-Rivas, G. R., Delgado-Gallegos, J. L., Garza-Treviño, G., Galan-Huerta, K. A., Buentello, Z. G., Roacho-Pérez, J. A., Santoyo-Suarez, M. G., Franco-Villareal, H., Leyva-Lopez, A., Estrada-Rodriguez, A. E., Moreno-Cuevas, J. E., Ramos-Jimenez, J., Rivas-Estrilla, A. M., Garza-Treviño, E. N., Islas, J. F. (2022).** Association between mortality and cardiovascular diseases in the vulnerable mexican population: a cross-sectional retrospective study of the COVID-19 pandemic. Front Public Health, Vol. 10, pp. 1008565. DOI: 10.3389/fpubh.2022.1008565.

8. **Quenzer, F. C., Coyne, C. J., Ferran, K., Williams, A., Lafree, A. T., Kajitani, S., Mathen, G., Villegas, V., Kajitani, K. M., Tomaszewski, C., Brodine, S. (2023).** ICU admission risk factors for latinx COVID-19 patients at a U.S.–Mexico border hospital. J Racial Ethn Health Disparities, Vol. 6, pp. 3039–3050. DOI: 10.1007/s40615-022-01478-1.

9. **Rahman, M. M., Khan, N. I., Sarker, I. H., Ahmed, M., Islam, M. N. (2023).** Leveraging machine learning to analyze sentiment from COVID-19 tweets: A global perspective. Engineering Reports, Vol. 5, No. 3, pp. e12572. DOI: 10.1002/eng2.12572.

10. **Shi, J., Chen, F., Chen, S., Ling, H. (2023).** COVID-19 over the last 3 years in China, what we've learned. Frontiers in Public Health, Vol. 11, pp. 1209343. DOI: 10.3389/fpubh.2023.1209343.

11. **Syeda, H. B., Syed, M., Sexton, K. W., Syed, S., Begum, S., Syed, F., Prior, F., Yu-Jr, F. (2021).** Role of machine learning techniques to tackle the COVID-19 crisis: Systematic review. JMIR medical informatics, Vol. 9, No. 1, pp. e23811. DOI: 10.2196/23811.

12. **Wolf, J. M., Wolf, L. M., Bello, G. L., Maccari, J. G., Nasi, L. A. (2023).** Molecular evolution of SARS-CoV-2 from december 2019 to august 2022. Journal of Medical Virology, Vol. 95, No. 1, pp. e28366. DOI: 10.1002/jmv.28366.

Article received on 03/10/2023; accepted on 05/01/2024.
** Corresponding author is Mariano Vargas-Santiago.*

ISSN 2007-9737

Building a Data Warehouse for Social Media: Review and Comparison

Maha Ben Kraiem[*,1], Jamel Feki[2]

[1] University of Sfax, MIRACL Laboratory,
Tunisia

[2] University of Jeddah,
Saudi Arabia

maha.benkraiem@gmail.com, jamel.feki@fsegs.rnu.tn

Abstract. The significant advancements in technology over the past few decades have given rise to a relatively straightforward array of Internet applications based on open source software. These applications and services aim to enhance online collaboration for a broad audience, particularly through social networking sites. These platforms have transformed the dynamics of online interaction and information exchange, with millions of users regularly engaging and sharing various digital content. Users express their thoughts and opinions on diverse topics, contributing valuable insights for personal, academic, and commercial purposes. However, the sheer volume and rapid generation of this data present a challenge for decision-makers and the underlying technologies to extract meaningful insights. To leverage the data derived from social networks, researchers have focused on assisting companies in comprehending how to conduct competitive analyses and convert this data into actionable knowledge. This paper offers a comprehensive literature review on data warehouse approaches derived from social networks. We commence by introducing fundamental concepts of data warehousing and social networks, followed by the presentation of three categories of data warehouse approaches, along with an overview of the most notable existing works within each category. Subsequently, we conduct a comparative analysis of these existing works.

Keywords. Data warehouse; social media; opinion analysis; business intelligence; OLAP.

1 Introduction

Over the past two decades, contemporary decision support and information systems have been essential for the efficient operation and expansion of successful global businesses. The cornerstone of decision support in these systems has been the integration of data warehouses and Online Analytical Processing (OLAP).

Widely accepted and employed worldwide, these technologies find application in diverse domains such as manufacturing, telecommunications, e-commerce, healthcare, education, research, and government. Research contributions, coupled with advancements in relevant hardware technology, have matured data warehousing systems, enabling them to manage substantial volumes of data and provide seamless access.

Online Analytical Processing (OLAP) serves as the central element, facilitating multidimensional data analysis, with continuous improvements and extensions made across various domains and datasets. Recent challenges faced by data warehouse technology, including handling multimedia, semi-structured data, text, and streams, have been met with significant and successful efforts.

The initial decade of the 21st century has been characterized by the widespread popularity and utilization of social media in the internet landscape. Billions of users engage with social media platforms for diverse purposes, such as social networking, blogging, information sharing, news discovery, or a combination of these activities.

Since their inception, social media platforms have made users more active in participatory networks, becoming an integral part of daily life by aiding users in connecting with family, keeping up

doi: 10.13053/CyS-28-1-4677

with friends or colleagues, and contributing to online discussions.

Millions of users regularly interact and share a variety of digital content, expressing their sentiments and opinions on a wide range of topics. Social media platforms have also played a strategic role in the corporate world, establishing links that connect customers to companies. Each of these links provides vast amounts of data, offering companies a substantial competitive advantage.

These links and opinions hold significant value for personal, academic, and commercial applications. However, the sheer volume and speed at which they are generated pose a challenge for decision-makers and the underlying technologies to derive meaningful insights from such data.

The massive data volumes generated by social media are characterized by being semi-structured, unstructured, and dynamic, presenting challenges for companies in terms of utilization, analysis, and storage. Managing and storing such large volumes of data without advanced platforms can be daunting for organizations.

Therefore, many companies opt to leverage the efficient technologies of data warehouses, enabling comprehensive analysis of massive data volumes. This analytical capability proves valuable for conducting competitive analyses and transforming the data into knowledge for decision-makers. The wealth of information generated by social media necessitates analysis by systems that can provide reliable and fast access for processing large amounts of data.

Among these systems, the Online Analytical Processing (OLAP) system stands out, offering interactive online data analysis in an environment capable of handling extensive data volumes. OLAP provides a simple and flexible modeling approach for various types of analyses based on a predefined multidimensional model, including pre-calculated data that accelerates the analytical processing.

With the emergence of social media, decision-makers aim to harness the vast volume of information generated by these platforms to enhance their decision-making processes. Many companies utilize data warehouse technologies to collect both their own data and that of their competitors.

Decision-makers often explore these networks to obtain additional information about companies, leading to better decision-making. In recent years, the explosive growth of social media has resulted in the generation of tremendous volumes of user-related data. This data presents a novel way to gather information in real time, giving rise to the field of *Social Media Analysis* [1].

This area has significant importance for the scientific community, addressing goals such as refining marketing strategies, profiling people's tastes, and targeting advertisements [2]. Exploring these data through an OLAP process could be a strategic opportunity, contributing to a wide variety of analytical needs [3]. Recognizing the importance of social media in the decision-making process, several researchers have focused on studying data warehouse approaches derived from social media.

This paper categorizes these approaches into three major classes: those addressing user behavior analyses, those integrating opinion analysis into data warehouse schema, and those dealing with social business intelligence. The primary objective of this paper is to provide a literature review on existing approaches to data warehouse design from social media.

The subsequent sections of this paper are organized as follows: Section 2 introduces the main concepts of data warehousing and social media. Section 3 presents existing approaches dealing with behavior analysis. Section 4 outlines approaches that integrate opinion analysis into data warehouse schema. Section 5 describes approaches related to social business intelligence. Section 6 provides a comparative study of the presented approaches, highlighting their limitations. Finally, Section 7 concludes the paper, outlining the main perspectives in this work.

2 Background

This section provides an introduction to key concepts and terminology in the realms of data warehousing, Online Analytical Processing (OLAP), and social media. It examines different options for the design and implementation architecture of data warehousing, outlining the various structural considerations. Furthermore, the section offers a comprehensive overview of selected popular social media platforms, elucidating their growth trajectories and patterns of usage in the current landscape.

2.1 Data Warehouse Concepts

A Data warehouse (DW) is a centrally managed and integrated database containing data from the operational sources in an organization. DW is an integrated repository of data put into a form that can be easily understood, interpreted, and analyzed by the people who need to use it to make decisions.

The most widely cited definition of a DW is from Inmon [4] who states that “a data warehouse is a *subject-oriented*, *integrated*, *non-volatile*, *and time-variant* collection of data in support of management’s decisions.”

- Subject-oriented: Data is modeled according to the subject area of the respective enterprise, and not according to the application needs of operational systems. A data warehouse does not focus on the ongoing operations; rather it focuses on modeling and analysis of data for decision making.
- Integrated: “A data warehouse is constructed by integrating data from heterogeneous sources such as relational databases, flat files, etc. This integration enhances the effective analysis of data”.
- Non-volatile: “A data warehouse is kept separate from the operational database and therefore frequent changes in operational database are not reflected in the data warehouse”.
- Time-variant: “The data collected in a data warehouse is identified with a particular time period. The data in a data warehouse provides information from the historical point of view.

The data in a data warehouse are organized according to a multidimensional model. This modeling provides a level of abstraction independently of technical aspects and focusing on decision-making needs [5]. The multidimensional modeling consists in defining the subject to be analyzed as a point in a multidimensional space [6].

In fact, data are organized in such a way to bring out the subject of analysis represented by the concept called *fact*, composed of *measures* corresponding to the additive information of the analyzed activity as well as the *dimensions* of this activity representing analysis axes.

A dimension is composed of attributes expressing the characteristics according to which the measures of the fact are analyzed (i.e., activity). The attributes of a dimension can be organized into hierarchies, from the finer to the most general granularity.

Relying on the fact and dimension concepts, it is possible to build different multidimensional models; the most popular one is the star model. A star model is composed of one central fact surrounded by dimensions, whereas the constellation model consists in defining a set of facts that share common dimensions.

2.2 Social Media Concepts

“Over the last few years, the Web has fundamentally shifted towards user-driven technologies such as blogs, social networks and video-sharing platforms. Collectively these social technologies have enabled a revolution in user-generated content, global community and the publishing of consumer opinion, now uniformly tagged as social media.

This movement is dominating the way we use the Internet and the leading social platforms like Facebook, MySpace, YouTube and now Twitter have moved into the mainstream. These sites are the tip of a redefinition of how the Internet works, with every site now incorporating the features that allow users to publish opinions, connect, build community, or produce and share content” [7].

Social media have become one of the most powerful sources of news updating, online collaboration, networking, viral marketing and entertainment. The terms of social media and social network are used every day. We saw to be more or less the same. In fact, social media include social networking, blogs, forums and platforms.

There are several types of social media; each one has features and different purposes. However, many researchers have different classification about types of social media sites ([8, 9, 10]) that emerges as a problem when realizing a study on social media.

In this context, Kaplan et Haenlein [9] classified social media sites into these six types based on social presence, media richness and self-presentation, and self-disclosure. Thus, they have got six forms of social media, which are: (a) collaborative projects (e.g., Wikipedia), (b) blogs (e.g., Wordpress.com, Blogger.com), (c) content communities (e.g., YouTube), (d) social networking sites (e.g., Facebook), (e) virtual game worlds (e.g., World of Warcraft), and (f) virtual communities (e.g., SecondLife)). In 2011, Akar and Topçu [10] add to this classification the microbloggings type such as Twitter.

Shankar and Hollinger [8] classified these new media into three groups: importun (Internet advertising, product placement in video games or advergames et m-commerce), non-importun (Internet advertising, social networking sites, podcasting, buzz or viral marketing) and user-generated (blogs, video site, ratings/recommendations and summary).

Taking into account the different types of social media proposed in the literature, we retain the classification of the most common forms of social media as follows:

- Social Networking sites: “These are sites mainly used for connecting with friends and family. They focus more on person-to-person conversations. Aside from personal conversations, these platforms encourage knowledge sharing. These platforms accommodate the different types of content formats from text to photos, videos, and other creative forms of content. They are considered the center of communication and a jack of all trades. Users are able to create unique interesting content, share their thoughts, and create groups based on similar interests. These sites are user-centered and are built around the social needs of the users and everything that is important to them. Businesses and marketers can fully maximize these platforms because they provide an immense amount of data. Also, they are able to reach the right people through adverts with specific metrics and demographics. They also provide the opportunity to engage with users which helps people connect with your brand on a more personal level. Some of such platforms include Facebook, LinkedIn, and Twitter.”
- Media sharing sites: “they have gained more prominence in recent times. Content like info graphics, illustrations, and images capture the attention of users more. Social media apps like Pinterest, Instagram, and Snapchat are designed to amplify the sharing of images. They say a picture is worth a thousand words, and using this can have lots of positive effects. Video content is one of the most captivating and engaging forms of content. Marketers and businesses have said that they have seen tremendous benefits in using videos. This form of content aids assimilation and understanding, hence why it is largely preferred by users. One major platform that reshaped how people interact with video content is YouTube. With over one billion active users monthly, the platform sometimes serves as a search engine for most users.”
- Discussion forums: “they are very essential because they allow users to ask questions and get answers from different people. These platforms are designed to spark conversations based on shared interests or out of curiosity. Some of such platforms include Quora and Reddit.
- Blogs: they are a great way for businesses and marketers to reach and provide credible information to their target audience. Platforms like Tumblr, Medium, OverBlog, canalblog and blogspot allow users to create a community where people with similar interests can follow them and read all they have to say about certain topics.”

The advent of social media as a novel source of data has significantly introduced new challenges concerning the modeling and handling of data. In the subsequent sections, we will conduct an extensive examination of strategies employed in designing a data warehouse derived from social media.

Our classification encompasses three main categories: (1) data warehousing for behavior analysis, (2) the incorporation of opinion analysis into the data warehouse, and (3) data warehousing tailored for social business intelligence. Section 3 will outline approaches centered around behavior analysis, followed by Section 4, which will elaborate on approaches proposing the integration of opinion analysis into the data warehouse. Lastly, section 5 will scrutinize approaches addressing social business intelligence.

3 Building Data Warehouse for Behavior Analysis

Social media is considered as an environment for human beings so they can express themselves through their interactions. Numerous approaches focused on user's activities on social media in order to help the decision makers to discover new knowledge and to analyze the behavior of people using social media.

The emergence of social media has sparked numerous research initiatives focused on behavior analysis and the extraction of knowledge from the data pertaining to users and their messaging activities. These researches include but are not limited to these works [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]. We examine these works which are the most representative of the state of the art that treat data warehouse.

3.1 Approach Proposed by (Bringay et al. 2011)

[11] devised a multidimensional star model tailored for the analysis of a substantial volume of tweets. Their approach incorporated information retrieval methods, introducing a modified metric named "TF-IDF $_{\text{adaptive}}$." This metric aimed to identify the most significant words based on hierarchical levels within the cube, specifically focusing on the location dimension. The initial step involved instantiating the multidimensional model of tweets to accommodate new dimensions that vary across different contexts.

Subsequently, the authors introduced the "TF-IDF $_{\text{adaptive}}$" measure in the second step, identifying the most relevant words in alignment with the hierarchy level of the cube by incorporating a new dimension termed MotMesh (MeshWords) into their multidimensional model. The final step aimed to determine the context of a tweet.

However, it's important to note that their case study was confined to a specific domain—tracking the progression of diseases, utilizing the MeSH (Medical Subject Headings) thesaurus. The proposed model was specifically crafted for this particular thematic trend.}

Additionally, the absence of schema definition rules for the data warehouse and the lack of adaptation of the proposed data warehouse model to the vast amount of social data present notable limitations in their approach.

3.2 Approach Proposed by (Liu et al. 2012)

A similar approach is put forth in the work of [12], where the authors introduce a text cube for the analysis and modeling of human, social, and cultural behavior (HSCB) from the Twitter stream within a textual database. They developed the Social Cube framework, comprising four core steps.

In the initial step, a framework for the data collection component is proposed, enabling the automatic extraction of relevant data from diverse sources like Twitter. Utilizing Twitter's Application Programming Interface (API), the authors create code for the automated extraction of real-time tweets on a given topic, transforming and loading them into the textual database for subsequent HSCB analysis.

Moving to the second step, the HSCB feature analysis component extracts linguistic features from the text using text analytics tools. These linguistic features serve as fundamental elements for HSCB dimensions, including affect, deception, and a sense of fatalism, as well as any prospective HSCB dimension.

The analysis framework considers the selection of linguistic features with reference to theories and psychological expectations.

The third step involves the design of a star schema to store the linguistic features extracted from various HSCB dimensions. Finally, in the last step, data mining techniques are employed to identify crucial linguistic features and construct predictive models for each HSCB dimension based on the selected features.

The primary objective for decision-makers in this work is to define the architecture of a text cube geared towards organizing social media data in multiple dimensions and hierarchies. However, it's noteworthy that the proposed data warehouse model is not tailored to effectively handle the substantial volume of social data.

3.3 Approach Proposed by (Rehman et al. 2012)

In the same context, [13] introduced a system designed for warehousing streams from Twitter, structured around a five-layer architecture. The layers include: i) The data source layer, utilizing available Twitter APIs, ii) The ETL layer (Extract, Transform, and Load) responsible for extracting data from tweets and processing it into a suitable format for the target database, iii) The Data warehouse layer dedicated to storing data derived from tweets, iv) The Analysis layer specifically designed for OLAP analyses of the tweets, and v) The Presentation layer presenting the results of the analyses.

The objective of this paper is to construct a comprehensive cube for OLAP analysis of tweets. However, it's important to note that the schema definition rules for the data warehouse are overlooked, and the proposed model is not optimized for handling the vast volume of social data. The authors focused on a specific type of social media, and their approach seems tailored to analyze a particular event.

3.4 Approach Proposed by (Cuzzocrea et al. 2015, 2016)

The presented work concentrates on the integration of knowledge mining approaches, specifically FFCA, and OLAP technology for analyzing unstructured data exchanged in social media, facilitating advanced analytics services. With Twitter as the focal point, the authors underscore the significance of implicit information within tweets that goes beyond the explicitly available metadata.

In a subsequent extension of their work, [15] delves into the definition of a multidimensional data model for storing tweet data to facilitate OLAP analysis. The authors commence by outlining the structure of the data cube.

Within their tweet cube model, dimensions are categorized into two types: (i) Semantic Dimension, extracted from the Wikipedia knowledge base, leveraging titles of Wikipedia articles and the Wikipedia category graph, and (ii) Metadata Dimension, encompassing information about the tweet derived from its metadata, such as timestamp, user, hashtag, location, etc.

Furthermore, the authors introduce a measure that exploits a wikification service, representing a sentence with a set of Wikipedia concepts. Subsequently, a summarization algorithm is proposed to select the most representative tweets for each cube based on the OLAP dimensional fact model.

A case study is presented, addressing microblog summarization through the utilization of timed fuzzy lattice generated from the execution of time-aware FFCA on the unstructured content of tweets.

3.5 Approach Proposed by (Yangui et al. 2017)

The researchers in [16, 17] present a four-stage methodology for defining a data warehouse schema from social networks. Initially, they design the initial data warehouse schema using a classical approach based on existing methods and tailored to structured and heterogeneous sources. In the second stage, they articulate a set of transformation rules facilitating the conversion of data warehouse schema concepts to specific concepts within a NoSQL database.

This stage comprises three key steps: defining features, generating clusters, and determining multidimensional concepts. Subsequently, social network profiling data is clustered based on user requirements, enabling the dynamic discovery of multidimensional concepts. To achieve this, the

SHICARO (Semi-supervised Hierarchical Clustering based on ranking features using Ontology) approach is employed.

Finally, the discovered multidimensional concepts enrich the NoSQL data warehouse schema to ensure schema evolution. However, it's important to note that in this work, the schema definition rules for the data warehouse are overlooked, and the proposed model is not adaptable to the substantial volume of social data.

3.6 Approach Proposed by (Moulai and Drias 2018)

In their work, Moulai and Drias [18] introduced a specific type of data warehouse named "Information Warehouse," primarily designed with information fact tables. The authors put forth a generic information warehouse architecture intended for the storage and analysis of various information sources, including scientific papers, press articles, and social media.

The outlined infrastructure is then applied to the case of Twitter, where a multidimensional information model is defined. The collected information flow is subjected to analysis using the A-priori algorithm, a data mining technique, to uncover association rules indicative of the topics discussed in the Twitter collection. The results obtained are promising, confirming the potential of the proposed paradigm.

However, despite the robust foundation of their approach for identifying a multidimensional structure suitable for social media data, the authors did not emphasize the semantics of the analyzed text in favor of a specific application domain. Additionally, they did not address the challenges related to the volume and velocity of social media data.

3.7 Approach Proposed by (Jenhani et al. 2019)

In their work, the authors in [19] introduce a large-scale system designed for structured information extraction from streaming and voluminous social media text, with the aim of easily integrating this information into a data warehouse.

They implement a novel approach within a large-scale architecture comprising Storm and Hadoop for extracting events from streaming social media text. Leveraging Storm's real-time processing capabilities; they collect tweets from the Twitter Streaming API and employ clustering techniques for data filtering.

To facilitate this process, the authors propose a snowflake schema for modeling event data. This schema enables both independent analysis of social media events and their integration with the existing enterprise data warehouse.

Additionally, the authors utilize the power of Hadoop for batch processing of large volumes of data, focusing on structured information extraction, specifically entities and events.

This entire process is considered a data preparation stage preceding the well-known ETL (Extract, Transform, Load) process. Once events are extracted, they can be loaded into the Social Media Data Warehouse (SMDW) using any ETL tool, and standard OLAP, data mining, and BI tools can be employed for further analysis.

For the data warehouse design, the authors propose a customized conceptual model specifically tailored for event data type modeling. This multidimensional design allows for the separate analysis of social media events and their integration with the existing enterprise data warehouse (EDW) data, enabling more accurate analysis. The connection between the SMDW and the EDW is facilitated through the addition of an intermediate bridge table.

3.8 Approach Proposed by (Ben kraiem et al. 2020)

In their work, [20] applied data warehousing technology to facilitate a comprehensive analysis of massive data volumes generated by the Twitter social network. They introduced a multidimensional model dedicated to online analytical processing (OLAP) of data exchanged through tweets.

The model comprises a set of facts and dimensions constructed from the structure of tweets, designed to be generic and not limited to predefined analytical requirements, thus offering broad analytical potential and capacity to address ad-hoc needs. Special considerations were given to the specifics of tweet data, including links between tweets and tweet responses. To accommodate this, the authors extended the concept of a fact by proposing a new type named “reflexive fact”, allowing connections between instances of the fact table and one or several instances of the same table.

Various options for enriching the multidimensional model were suggested, such as adding new elements like measures and hierarchies. To validate their proposals, the authors developed a software prototype called TweetOLAP, demonstrating through extensive experimentation how the resulting data warehouse can be used for various analytical tasks.

Additionally, a solution based on five OLAP operators was proposed to support analyses considering the specificities of the proposed multidimensional model called “Tweet Constellation”.

In a related work [21], facing large volumes of data with a significant amount of missing data, the researchers proposed extended versions for conventional OLAP operators, namely Null-Drilldown, Null-Rollup, and Null-Select. These extended operators process OLAP queries on datasets with missing data, providing options for handling missing data in analysis results.

The researchers introduced the options of All, $\text{All}_{\text{NullLast}}$, or Flexible, with $\text{All}_{\text{NullLast}}$ reorganizing the multidimensional table by moving non-significant rows to the bottom and Flexible displaying percentages of non-null data. Furthermore, to exploit the reflexive relationship on fact instances, two specific OLAP operators, FDrilldown and FRollup, were proposed.

These operators facilitate intuitive navigation between different levels within the fact, catering to decision-making applications and enabling diverse analyses by showcasing how information propagates through each tweet.

4 Building Data Warehouse for Opinion Analysis

Sentiment analysis, also known as opinion mining, involves the computational study of opinions, sentiments, and emotions expressed in text. The integration of opinion data has become a prominent topic in various research communities, notably within Data Warehousing and Decisional Support.

The aim of this section is to thoroughly examine the existing approaches for integrating sentiment analysis into the schema of data warehouses. Specifically, we scrutinize the research conducted by the most notable authors [22, 23, 24, 25, 26, 27, 28, 29, 30] which are the most representative works.

4.1 Approach Proposed by (Moya et al. 2011)

[22] introduced a multidimensional data model designed to integrate sentiment data extracted from Web 2.0 customer opinion forums into the corporate data warehouse.

This model comprises two primary components. The first part focuses on corporate information, sourcing data from internal documents and company databases through traditional Extract, Transform, and Load (ETL) processes. Corporate facts in this section typically encompass standard Business Intelligence (BI) measures such as sales and profits.

The second part is dedicated to sentiments within the data warehouse, containing information derived from user reviews on products obtained from opinion forums. This part operates at two levels of granularity: the overall sentiment regarding the product and specific sentiments about the product's features, as mentioned by users in their opinion posts.

The initial step in the proposed approach involves gathering customer opinions from the web to identify products that have been subject to customer feedback. Subsequently, the authors identified potential features influenced by opinion words. They compiled a list of opinion words through the intersection of adjectives from two lists, manually verifying and supplementing it with adverbs and verbs of context-independent polarity.

The second step involves classifying potential features based on their importance, determined by both functional relevance and feature frequency [31]. Finally, the authors suggested calculating characteristics of synonym groups using the Jaccard distance function, which considers both the lexicon and overlapping word synonyms.

4.2 Approach Proposed by (Costa et al. 2012)

[23] introduced an architecture that emphasizes the integration of social networks and sentiment analysis with user decision-making processes. The primary focus of this work is on extracting data from Twitter and applying sentiment analysis to generate a data warehouse.

The proposed software architecture, named Online Social Networks Business Intelligence (OSNBIA), is structured as follows: (1) Social Networks Crawling: Utilizing application programming interfaces (API) provided by Social Network Sites, the authors retrieve tweets containing the text "lenovo ThinkPad". (2) Data Cleansing: In this phase, inconsistencies in the data are corrected before moving on to the next step.

Aspects such as completeness, consistency, validity, conformity, accuracy, and integrity are addressed. For missing data, the constant 'NOT AVAILABLE' is used to fill attributes when certain data attributes are inaccessible or lack content. (3) Analysis using Mining Algorithms: With the cleaned data, [23] employed link mining and opinion mining algorithms to identify the sentiments expressed in 58,906 tweets.

New attributes resulting from this analysis are added to the tables, creating an analyzed data repository that is inserted into the data warehouse. (4) Data Warehouse Insertion: The researchers then inserted these files into a data warehouse to analyze the sentiments expressed in tweets relative to sales performance.

Following the generation of the data warehouse, QlikView was utilized to develop a Business Intelligence (BI) analysis application, providing greater flexibility for data analysis. However, it is important to note that a drawback of this approach is the absence of a data warehouse schema.

4.3 Approach Proposed by (Rehman et al. 2013)

[24] extends their previous work [13], aiming to enhance Online Analytical Processing (OLAP) for multidimensional analysis of data from social networks. The extension involves integrating text mining methods, opinions, and knowledge discovery techniques with a data warehousing system. The researchers initiate the process by identifying the facts, measures, hierarchies, and dimensions of the Twitter data warehouse.

This proposed data warehouse adopts the aggregation-centric multidimensional data model, facilitating drill-down and roll-up operations. Subsequently, the researchers enrich and extend the social media dataset to provide new analytical aspects for business analysts. Text and opinion mining algorithms, along with sentiment analysis, are applied to support both exploratory and predictive analysis of social media data.

Two APIs, namely AlchemyAPI for sentiment analysis and OpenCalais for topic extraction and concept tagging, are utilized to ensure uniformity of results. Using decision tree classifications, the authors mine the dataset for features classifying tweets into multiple popularity classes, considering hashtags, sentiment, and user popularity as input features for the model.

In the final stages, the researchers modify concepts related to slowly changing dimensions as presented in [6]. They update the name and screen name attributes by replacing existing data with new ones, focusing on changes in dimensions and hierarchies within the data warehouse. The ETL process (extraction, transformation, and loading) is repeated each time new data are uploaded into the data warehouse.

The resulting data warehouse was used in an attempt to perform analyses during the 2012 European Football Championship final played between Spain and Italy on July 1, 2012.

4.4 Approach Proposed by (Walha et al. 2016)

[25] introduces the integration of social opinion data in multidimensional design, combining sentiment analysis techniques and Extract, Transform, Load (ETL) design to present a novel approach for social ETL design. The researchers define a lexicon opinion analysis approach that extracts sentiment polarity from informal text expressed on the Twitter social network.

They propose a new algorithm, POLSentiment, based on lexical resources to extract opinion words and emoticons from tweets and then determine their positive or negative polarity. The process involves the following steps: (1) *Creating an*

Opinion Dictionary: The researchers construct an opinion dictionary based on the AFINN word list, specifically designed for microblogs and considered a standard for opinion analysis.

This dictionary includes 2477 English words and phrases. They further enrich the dictionary with a list of positive and negative opinion words, sentiment words for English, and emoticons. (2) *Automatic Lexicon-Based Method*: The researchers propose an automatic lexicon-based method to determine tweet polarity based on the opinion lexicon used in the tweet, which includes emoticons and opinion words. (3) *Tweet Preprocessing*: This step involves cleaning the tweet by removing diacritics, useless characters, URLs, repetitive characters, etc. (4) *Tweet Tokenization*: The text is segmented into words, phrases, and symbols called tokens. (5) *Detecting Tweet Polarity*: This process determines whether a piece of writing is positive, negative, or neutral.

The authors extract the opinion lexicon used in the tweet, including opinion words, their modifiers, and emoticons. The lexicon is determined from previously defined opinion and emoticon dictionaries. The POLSentiment algorithm is then used to calculate tweet polarity and perform opinion analysis. (6) *Loading Step*: In the final step, opinion analysis subject and axes are defined in a Data Warehouse Bus (DWB) star schema, which includes dimensions, measures, facts, attributes, and parameters.

4.5 Approach Proposed by (Ahsene Djaballah et al. 2019)

In 2019, Ahsene Djaballah et al. [26] introduced an approach for analyzing terrorism-related activities in social networks through the utilization of data warehousing and OLAP analysis.

The architecture of their proposed approach comprises five layers: (1) the data source layer, which is represented by available APIs for searching social network data and their metadata, including external sources such as location data; (2) the ETL layer, responsible for extracting data from heterogeneous sources, performing necessary treatments and cleanings using text mining techniques, and loading the processed data into the data warehouse; (3) the Data Warehouse layer, characterized by a star multidimensional model designed for analyzing business processes; (4) the analysis layer, incorporating an OLAP server that translates users' queries into requests on the data warehouse and provides results to decision support tools; and (5) the presentation layer, consisting of reporting tools for different visualizations of the analysis layer.

The researchers implemented their approach on the Twitter social network, employing the FEEL dictionary (a French Expanded Emotion Lexicon) for sentiment analysis to determine positive scores, specifically tweets inciting terrorism. However, a drawback of this approach is its reliance on a single type of social media, and the proposed model may not be adaptable to the vast amount of social data.

4.6 Approach Proposed by (Valêncio et al. 2020)

[27] introduced a normalized data warehouse schema designed for modeling social media data originating from two distinct platforms, Facebook and Twitter. This normalized data warehouse model aims to eliminate redundant data storage by focusing on quantitative attributes from publications on social media, thereby enhancing the efficiency of data mining algorithms and reducing execution time.

The authors presented the Configurable Load and Acquisition Social Media Environment (CLASME), a tool facilitating data preparation from Facebook and Twitter to uncover valuable knowledge and support analysts in decision-making. The ETL (Extract, Transform, Load) phase involves obtaining data from various sources, followed by data cleaning and standardization during the transformation phase, depending on the application's objectives.

The load phase maps and stores the transformed data into the appropriate section of the data warehouse, known as Data Mart. Subsequently, qualitative data are categorized as positive, negative, or neutral, and opinions about the posts are determined before loading quantitative data.

Once the ETL and data warehousing processes are complete, data mart algorithms are applied during the data analysis phase to validate the classification model. Finally, the results obtained are interpreted to facilitate the decision-making process.

4.7 Approach Proposed by (Gutiérrez-Batista et al. 2021)

[28] aimed to enhance decision-making based on a substantial volume of text extracted from social media, spanning various topics and timeframes, by introducing a fuzzy sentiment analysis dimension. The authors employed OLAP for text storage and extraction within their multidimensional model.

The fuzzy dimension's hierarchy levels encompassed five tiers, ranging from the most general to the most specific. Texts underwent clustering, sentiment scores were assigned, and Fuzzy Logic was applied for processing. Tools like Text Blob and VADER, known for their efficacy with social texts, were utilized as unsupervised tools to ensure a more generic applicability.

A comparative study conducted on real tweets and movie reviews, employing six machine learning algorithms, demonstrated the high performance and accuracy of this method. The proposed approach involves four primary processes. Initially, a Fuzzy Sentiment Dimension is created from texts extracted from social networks to facilitate multidimensional sentiment analysis.

Subsequently, an automatic process of document clustering is established, considering the sentiments expressed in the texts. Sentiment evaluations are automatically assigned to each document using *linguistic labels*, and a hierarchical structure is constructed to enable sentiment analysis at different granularity levels.

An adaptive process is then developed for the automatic selection of linguistic labels and the definition of membership functions for these labels. Finally, storage and query extensions are defined to support the Fuzzy Sentiment Dimension.

The first extension allows the definition of structures such as cubes, dimensions, hierarchies, and levels to facilitate fuzzy multidimensional analysis of users' opinions.

The second extension enables querying the Fuzzy Sentiment Dimension by defining operations in the multidimensional model, such as roll-up and drill-down. The notable advantage of this method lies in its fully automated and unsupervised nature.

4.8 Approach Proposed by (Moalla et al. 2017, 2022)

In 2017, Moalla et al. [29] propose a new method of opinion analysis based on machine learning that determines the polarity of users ‘comments shared on different social media. The latter will be integrated in the ETL (Extract, Transform and Load) process to analyze the users' opinions.

The proposed method is based on the n-grams technique to construct a semi-automatic dictionary for positive and negative keywords that is used in the learning phase to establish the prediction model. In addition, they propose a new features vector specific for social media for classifying the comments as positive, negative or neutral.

The evaluation results performed on the both publicly data sets Stanford Twitter Sentiment (STS) and Sanders dataset showed a high accuracy level. In 2022, [30] presented an extension of their previous work. They propose a new approach for building a data warehouse from social media for opinion analysis.

The proposal consists of four phases: Data extraction and cleansing, Transformation, loading, and analysis. They have presented the different stages of data extraction and cleansing. These steps are intended to model data marts for each social media. In the transformation phase, the authors have detailed the mapping and merging step to obtain a generic data warehouse schema.

In addition, they have summarized the opinion analysis step. After that they presented the implementation of a data warehouse under a database NoSQL- oriented documents. Finally, in the analysis and reporting steps, they performed some queries on our data warehouse.

5 Building Data Warehouse for Social Business Intelligence

Business intelligence (BI) is the set of devices, the querying tools and the data analysis used to drive a company and help it in the decision making. Today, BI decision-making processes are informed by social media pattern. Social networks are an essential aspect of the information infrastructure. These social media sites have attained an unparalleled degree of penetration for users, customers, and enterprises to provide the professional environment with a valuable information source.

In this context, a novel area known under the name of Social Business Intelligence (SBI) appeared which refers to the discipline that aims at combining corporate data with user generated content (UGC) to let decision makers analyze and improve their business based on the trends and moods perceived from the environment [33].

The purpose of this section is to study in depth the proposed existing approaches dealing with behavior analysis. More precisely, we examine these works of [32, 34, 35, 36, 37, 38, 40].

5.1 Approach Proposed by (Gallinucci et al. 2013, 2015)

In 2013, Gallinucci et al. [32] focused in their study on the social business intelligence, which enables to combine corporate data with the user-generated content (UGC) to help the decision makers to improve their company and aggregate subjects at different levels.

Therefore, they proposed to model topic hierarchies in ROLAP systems called meta-stars. This approach is based on the combination of the traditional dimension tables and the navigation tables to deal with the dynamics of the subject area. Gallinucci et al. [32] introduced the architecture for SBI (social business intelligence) that integrates both the corporate data and the sentiment data of the Web users (UGC).

In the implementation of this architecture, the researchers manually defined the topics and roll-up relationships. After that, they presented a cube to analyze the sentiments expressed by the Web users. Furthermore, they defined a set of relationships between the topics in the hierarchy roll-up. In fact, they proposed to model topic hierarchies on ROLAP platforms combined with classical dimension tables and with recursive navigation tables.

Then, they extended the obtained result by using meta-modeling called meta-stars. Therefore, the authors tested some OLAP queries to evaluate the performance of meta-stars against star schema. In fact, they noted that meta-stars are better in terms of space efficiency and query expressiveness and lower in terms of time.

In [33] improved their work by extending their meta-stars model. Firstly, they took non-covering and non-strict hierarchies. Secondly, they dealt with the techniques of slowly changing topics and levels. Thirdly, they supported the semantics queries on topic hierarchies.

Finally, they evaluated the meta-star approach proposed on wider set of tests. Nevertheless, these works are limited to the use of one type of social media. Additionally, the proposed model is not adaptable to the huge amount of social data.

5.2 Approach Proposed by (Francia et al. 2014)

The authors of [34] propose an iterative methodology for designing and maintaining SBI applications that reorganizes the activities and tasks normally carried out by practitioners. They proposed architecture for the SBI process where the information resulting from clip analysis is stored into a data mart in the form of multidimensional cubes to be accessed through OLAP techniques.

This architecture is composed of six features: (1) An ODS (Operational Data Store) that stores all the relevant data about clips, their topics, their authors, and their source channels; to this end, a relational database is coupled with a document-oriented database that can efficiently store and search the text of the clips and with a triple store to represent the topic ontology.

(2) A data mart that stores clip and topic information in the form of a set of multidimensional cubes to be used for decision making. (3) A crawling component that runs a set of keyword based queries to retrieve the clips (and the related meta-data) that lies within the subject area.

(4) An ETL (Extraction, Transformation, and Loading) component that turns the semi-structured output of the crawler into a structured form and loads it into the ODS, and then periodically extracts data about clips and topics from the ODS to load them into the data mart. (5) A semantic enrichment component that works on the ODS to extract the semantic information hidden in the clips, such as the topic(s) related to the clip, the syntactic and semantic relationships between words, and the sentiment related to a whole sentence or to each single topic it contains.

(6) An OLAP front-end to enable interactive and flexible analysis sessions of the multidimensional cubes. The disadvantage of this approach is the lack of data warehouse schema.

5.3 Approach Proposed by (Kurnia et al. 2018)

In the same context, Kurnia et al. [35] developed a business intelligence dashboard to observe the performance of each Topic or channel of news posted on Facebook and Twitter. For this reason, a data warehouse model and software for the business intelligence system are designed and implemented. The architecture of the proposed system is composed of four stages.

Firstly, data collection is extracted from Facebook and Twitter through the API available on each platform. Secondly, content analysis used text classification techniques like Naive Bayes, Decision Tree and SVM to attribute a category or class to the data retrieved based on the characteristics of the document.

But before going into the classification algorithm processing, the data will go through the preprocessing stage such as case folding, tokenizing, filtering, and stemming to eliminate data noises. Thirdly, the Data warehousing design method used is [6] method which there are 4 stages that must be passed in the design of data warehouse that is select the business process, declare the grain, identify the dimensions, and identify the facts.

Therefore, a star schema model is proposed to show the number of comments, tweets, likes, etc., for each topic. Fourthly, the design of business intelligence with the Carlo Vercellis method, where in this method there are four main phases, namely analysis, planning, planning, implementation and control. Nevertheless, the proposed model is not adaptable to the huge amount of social data.

5.4 Approach Proposed by (Girsang et al. 2020)

Similarly, [36] presented a Business Intelligence application dashboard using a data warehouse to provide a solution for powerful, effective, and limitless news sources to the journalistic community. The proposed approach is divided into four main phases. The first phase is the data collection which consists of collecting data from selected Social Media Platforms.

Data retrieval will be carried out periodically by crawlers who have been created with the Social Media API Token input. Thus, the results of data retrieval will be saved on the database as raw data. The second phase is content analysis s including retrieving data for content analysis from each data on each Social Media platform. Then, the data is processed for text classification using SVM into 10 news categories.

The results of text processing will be stored in the database as analysis data. The third phase is the data warehouse process. This is done by defining the transformation and loading procedures of the ETL process.

Finally, the last phase is the client side. A business intelligence dashboard is created when the data is stored which can help journalism in the analysis of news information. The disadvantage of this approach is the use of one type of social media. Additionally, the proposed model is not adaptable to the huge amount of social data.

5.5 Approach Proposed by (Mouyassir et al. 2021)

The authors of [37] presented an analysis of the applicability of social media in BI that helps businesses to get a global and well-defined perception of consumer's sentiments and emotions. This process aims to analyze data according to several important components. The first step in this process is data collection.

The researchers use Apache Flume as data collection tool. In this second step, they deal with the data cleaning, including several errors that require filtering and sorting, discarding irrelevant data, meaningless data, eliminating redundant data. The third stage includes evaluating the quality of the social media data after it has been filtered, and using text classification.

Mouyassir et al. used a set of text classification algorithms like SVM, Decision Tree. The fourth step is store data. At this phase, the data will be processed in a data warehouse in a distributed and non-volatile method. Then, the authors use the NoSQL language to collect all the data that has been deposited in a distributed manner.

Finally, reports are created as part of this pre-process to help the end-users understand the results. These end-users would be able to get a better understanding of consumer behavior, allowing them to interpret the data and making it understandable.

Reporting is about transforming data into information, while analysis is the process of transforming information into knowledge. Nevertheless, in this work the schema definition rules of data warehouse are ignored. Besides, the authors have not specified a type of social media. Therefore, the proposed approach can be used only for ensuring a special treatment.

5.6 Approach Proposed by (Aramburu et al. 2021)

[38] define the special requirements of the analysis cubes of a Social Business Intelligence (SoBI) project. They present a new data processing method for SoBI projects whose main contribution is a phase of data exploration and profiling that serves to build a quality data collection with respect to the analysis objectives of the project. The authors propose a new data processing methodology that consists of three main phases: Collection Construction, Data Preparation and Data Exploitation (see Figure 1).

The first phase is the construction of a collection of tweets through an exploratory process executed by the user and directed by the quality of the recovered data. When a quality data collection is ready, in the data preparation phase, the facts of the analytical cubes are extracted from the posts and then exploited in the last phase of the process.

During the Collection Construction phase, the user executes some data exploratory and profiling tasks to assess and improve data coverage and data quality until obtaining a quality collection that meets the project's analysis objectives. More specifically, this phase consists of two complementary and iterative tasks: Evaluating the subject coverage of the collection with respect to its topics and users and analyzing and improving the quality of the collection by filtering the posts of low quality or out of the scope.

In the Collections Construction and Data Preparation phases, the processing of tweets to extract the measures and values that serve both to clean the collection and to feed the analysis cubes, can be made in different ways. Some values are directly available in the tweets metadata, such as post-date and number of followers of the user. Other values can be calculated with a simple processing like counting tweets over a period. Evaluating the grammatical richness of a post is executed by a process that calculates some textual measures [39].

Finally, in the Data Exploitation phase, the analysis cubes constructed by processing the tweets collection can be stored into the corporate Data Warehouse for future uses. OLAP applications, or any other Business Intelligence or Data Mining tools, can be applied to analyze and extract new insights from these cubes. However, in this work the schema definition rules of data warehouse are ignored. Additionally, the proposed model is not adaptable to the huge amount of social data.

5.7 Approach Proposed by (Lanza-Cruz et al. 2023)

[40] propose a methodology for author profiling (AP) in Twitter based on social business intelligence roles. The method allows the unsupervised construction of a labeled dataset that serves as input to different text classification tasks. They automatically build a training dataset from unlabeled user descriptions by making use of the multidimensional user profile knowledge model provided by the analysts.

The proposed methodology relies on semantic knowledge encapsulated in ontologies provided by analysts at the commencement of a Social Business Intelligence (SBI) project. From these ontologies, basic linguistic information is extracted to identify potential unlabeled user profiles. Consequently, the generated training data are directly associated with the concepts represented in the knowledge multidimensional model, such as users' roles.

This integration allows [40] to verify the consistency and identify conflicts within the training data. This approach contributes to the existing body of knowledge by offering an integrated perspective on ontologies and predictive models for Audience Profiling (AP) in social networks.

Table 1. A comparative study of data warehouse design approaches for behavior analysis

Approach	Modeling level		ETL process		Social media	Analysis process	Objectives	Drawbacks
	Conceptual	Logical	Tools used	Language				
Bringay et al. 2011	Star schema	Not mentioned	Postgre SQL and Pentaho Mondrian	Not mentioned	Twitter	Classic OLAP operators	Leveraging a multidimensional star model for the examination of tweets and proposing relevant metrics conducive to knowledge exploration.	Utilizing a partial modeling approach with a predefined data warehouse schema.
Liu et al. 2012	Star schema	Not mentioned	Not mentioned	Not mentioned	Twitter	Classic OLAP operators	Text cube architecture is presented for analyzing human social and cultural behavior with the capacity to develop prediction models and perform analyses.	Insufficient theoretical framework for opinion analysis.
Rehman et al. 2012	X-DFM	Not mentioned	BaseX and Microsoft SQL Server	Not mentioned	Twitter	Classic OLAP operators	The authors propose an exhaustive cube for OLAP analysis of tweets. The suggested model may be utilized to complete any tasks based on the mining or aggregation of data.	Absence of defined rules for schema.
Cuuzzocrea et al. 2015, 2016	DFM	Not mentioned	Not mentioned	Not mentioned	Twitter	Classic OLAP operators	OLAP technology used with knowledge mining methods (i.e., FFCA) to analyze multidimensional tweet streams of unstructured social media data.	The vast volume of social data cannot be accommodated by the suggested approach.
Yangui et al.. 2017	X-DFM	NOSQL Data base	Not mentioned	Not mentioned	Twitter		The suggested methodology makes use of the established design methods' maturity, the NOSQL Data Base's scalability, and the capacity to dynamically identify multidimensional concepts using clustering algorithms.	The schema definition rules of data warehouse are ignored.
Moulai and Drias. 2018	Star schema	Not mentioned	Not mentioned	Not mentioned	Twitter		A specific DW called “Information Warehouse” which focused on semantic extraction and modeling. It is the structure which stores data having meaning and significance such as text, image, video, etc.	Absence of the semantic of analyzed text to the profit of a specific application domain. Absence of volume and velocity problems of social media data.
Jenhani et al. 2019	Snowflacke Schema	Not mentioned	Hadoop	Not mentioned	Twitter		An approach in a large-scale architecture based on distributed storage and parallel processing for event extraction from streaming social media data.	the design process is not present in detail by presenting the rules that lead to the data warehouse schema.
Ben kraiem et al. 2020	Extended conceptual constellation schema	ROLAP	JAVA and ORACLE 10	Not mentioned	Twitter	Extended OLAP operators	The conceptual model takes into account the specificity of tweet and tweet response and missing data. Proposal of new OLAP operators to deal with the specificities of the proposed model.	The proposed model is not adaptable to the huge amount of social data.

Furthermore, the method is adaptable to dynamic scenarios where semantic knowledge requires updating to accommodate new roles or dismiss others. In such instances, the method constructs a new training dataset and develops new predictive models based on the updated knowledge.

Another noteworthy aspect of this approach is that the utilization of user profiles, rather than their posts or metrics, is deemed sufficient for characterizing their business roles. Previous methods, relying on posts and metrics, yielded poor results due to the redundant, often shared, and heterogeneous nature of social network content.

The multidimensional AP approach addresses the demand for analysis based on dynamic dimensions inherent in social media. It aids information systems in characterizing the audience of popular topics and news.

However, a drawback of this approach is the absence of a data warehouse schema. Additionally, the authors concentrated on a specific type of social media.

Table 2. A comparative study of data warehouse design approaches for opinion analysis

Approach	Modeling level		ETL process		Social media	Analysis process	Objectives	Drawbacks
	Conceptual	Logical	Tools used	Language				
Moya et al. 2011	Constellation schema	-	SQL server Business Intelligence Studio	Not mentioned	web	Classic OLAP operators	The display of a sentiment-integrated multidimensional data model. Sentiment data extraction yields a semantically rich data collection that supports sophisticated queries.	Lack of DW schema.
Costa et al. (2012)		-	Data Manager and ORACLE	Not mentioned	Twitter	Classic OLAP operators	The establishment of a business intelligence tool that combines social network and sentiment analysis with the decision-making processes of the user.	Lack of schema definition rules.
Rehman et al. 2013et al. 2012	X-DFM	-	BaseX and Microsoft SQL Server	Not mentioned	Twitter	Classic OLAP operators	The integration of opinion mining methods and knowledge discovery techniques into the data warehousing system, in order to perform multidimensional social media analysis.	Insufficient theoretical framework for opinion analysis.
Walha et al. 2016	Star schem	-	Not mentioned	Not mentioned	Facebook	Not mentioned	An ETL design technique that incorporates user opinions as given on the well-known social network Facebook. The list of each element of the ETL process is defined.	The massive volume of social data cannot be accommodated by the suggested.
Ahsene Djaballah et al. 2019	Star schema	Not mentioned	Postgre RDBMS	-	Twitter	Classic OLAP operators	Proposition of a data warehouse using data mining technique to analyze a related to terrorism in the social network twitter.	Lack of schema definition rules.
Valêncio et al. 2020	Constellation schema	Not mentioned	PostgreSQL9.5 JAVA	-	Facebook Twitter		The development of a social media data integration model based on a data warehouse to reduce the computational costs related to data analysis, as well as supports the application of techniques to discover useful knowledge.	Lack of a theoretical approach for opinion analysis.
Gutiérrez-Batista	Star schema	-		-	Twitter Movie reviews	Extended OLAP operators	Creating a Fuzzy Sentiment Dimension from texts extracted from social networks that facilitates the multidimensional sentiment analysis in social networks. Establishing an automatic process of documents clustering, taking into account the sentiments expressed in the.	The proposed model is not adaptable to the huge amount of social data.
Moalla et al. 2017, 2022	X-DFM	Presented	Microsoft SQL Server and XML (2017) MongoDB	-	Facebook Twitter Youtube		Proposition of a technique for social media opinion analysis that uses machine learning to determine the degree of polarity in user comments.	Insufficient theoretical framework for opinion analysis.

6 A Comparative Study of the Existing Approaches

Social media, as a burgeoning data source, has introduced novel challenges in data analysis and manipulation. The research landscape has witnessed a surge in studies focusing on the analysis of data extracted from social media, leading to the emergence of new analytical domains.

However, a limited number of studies have delved into the realm of multidimensional data modeling of data warehouses derived from social media.

Table 1 provides a comparison of behavior analysis approaches, while Table 2 delineates a comparative overview of methods integrating sentiment analysis into data warehouse structures.

Table 3 presents a comparative analysis of approaches integrating social business

Table 3. A comparative study of data warehouse design approaches for social business intelligence

Approach	Modeling level		ETL process		Social media	Analysis process	Objectives	Drawbacks
	Conceptual	Logical	Tools used	Language				
Gallinucci et al. 2013, 2015	Meta-star schema	-	Talend	Not mentioned	Web	-	Model suggestion for ROLAP platforms that considers topic hierarchies The ability to enable OLAP queries with increasing expressiveness and complexity, starting with queries that solely use static levels and progressing to queries that take semantics into account.	Topic and roll-up relationships are defined manually.
Francia et al. 2014	Meta-star schema	-	MongoDB	Not mentioned	Twitter	-	The proposal of an interactive methodology for designing and maintaining Social Business Intelligence (SBI) applications.	Lack of schema definition rules.
Kurnia et al. 2018	Star schema	-	CodeIgniter framework	-	Facebook + Twitter	-	Development of a business intelligence dashboard to evaluate the performance of each topic posted on Facebook and Twitter.	Partial approach at the modeling level (fixed DW schema).
Girsang et al. 2020	Star schema	-	Pentaho	Python script	Twitter	-	The presentation of a Business Intelligence application dashboard employing a data warehouse to assist and provide the journalistic community with a solution for powerful, effective, and limitless news sources.	The proposed model is not adaptable to the huge amount of social data.
Mouyassir et al. 2021	Meta-star schema	-	Apache Flume	NoSQL	Not mentioned		An analysis of the applicability of social media in BI that helps businesses to get a global and well-defined perception of consumer's sentiments and emotions.	Lack of schema definition rules.
Aramburu et al. 2021	Star schema	-	-	-	Twitter		Building a quality data collection in the data preparation phase of Social Business Intelligence projects. A methodology that considers collection construction as an iterative exploration process in which the user analyses the current collection from the point of view of the analysis objectives and discovers clues about how to improve it.	Lack of DW Schema.
Lanza-Cruz et al. 2023	Constellation schema	-	-	-	Twitter		The application of author profiling (AP) in order to characterize both the contents generators and the audience that is interacting with these contents.	The proposed model is not adaptable to the huge amount 0.

intelligence into data warehouse schemas. Several criteria are employed for comparison:

– Modeling level: Indicates the level of modeling, encompassing conceptual (star model, constellation model, snowflake model, DFM, x-DFM, etc.) and logical models (ROLAP, MOLAP, HOLAP).

– ETL process (extract–transform–load) : Encompasses the tools and modeling languages employed in the ETL process.

– Social media: Identifies the specific social media platforms used as data sources.

– Analysis process: Reveals whether the approach utilizes classic OLAP operators (drill down, rollup) or defines specific operators.

– Objective: Provides a succinct overview of the general idea behind each approach.

– Drawbacks: Describes the limitations and shortcomings of each approach.

Upon examining the modeling level, it is apparent that most approaches explicitly address conceptual modeling, with [20] being an exception as it presents both logical and conceptual modeling.

Regarding the social media criterion, the majority of approaches focus on a single social media platform, such as Twitter, Facebook, or the web. Only a few studies tackle the challenge of modeling data warehouse schemas from multiple social media platforms [29, 33, 35, 27, 28].

In terms of the ETL process criterion, which is crucial in data warehouse construction, only [25, 36, 37]) explicitly address this stage. Notably, other researchers do not provide a detailed definition of the various functions of the ETL process, and diverse tools are employed in this process.

Concerning the analysis process criterion, most approaches leverage classical OLAP operators, with a notable lack of specific operator definitions, except for the approach proposed by [20], which introduces new OLAP operators enhancing existing solutions and dealing with missing values and reflexive relationships on fact instances.

Each of the discussed approaches has its strengths but also exhibits certain weaknesses. Notably, existing design methods predominantly focus on Twitter as a data source, neglecting other social media platforms. The design process lacks detailed presentation, particularly in defining rules guiding data warehouse schema creation.

Despite the use of commercial tools for opinion analysis in the reviewed approaches, there is a noticeable absence of a theoretical framework for opinion analysis. Moreover, only a few works have concentrated on creating data warehouses under Hadoop, MapReduce, and NoSQL databases to handle the voluminous and massive data generated from social media.

To address these limitations, we propose a novel approach leveraging data warehousing technology to comprehensively analyze massive data volumes from various social media platforms. This approach aims to define a method for opinion analysis within a decisional system and utilize NoSQL databases to efficiently handle large amounts of data and enhance the analysis process.

7 Conclusion

In this work, we reviewed the research on social media data warehouse architecture strategies. To be more explicit, we discussed the fundamental ideas behind data warehouses and social media, and we classified social data warehouse design methods into three groups, namely behavior analysis, integration of sentiment analysis and social business intelligence in data warehouse schema.

Subsequently, based on the criteria we had established, we offered a comparison of the existing approaches. These criteria include modeling level, ETL process, the used social media, the objective and the drawbacks of each approach. Although the contributions that have been given are strong, they have significant flaws.

They may be summed up as the absence of schema defining standards, the usage of just one particular social networking platform, and the exclusive reliance on relational databases for storage. As future work, we intend to apply the data warehousing technology to enable comprehensive analysis of massive data volumes generated by the most popular social media. We aim to propose a multidimensional model dedicated to the on-line analytical processing (OLAP) of the data exchanged through social media.

We will ensure that this model is generic, that is, not limited to a set of pre-determined analytical requirements, which gives it a broad analytical potential and capacity to respond to ad-hoc needs. Besides, we will also take into account the specificities of such data. It would be interesting to define an approach enabling OLAP to keep up with volatile data using the concepts of slowly changing dimensions to enable analysis of both the recent state of data and any of its previous states. Also, it would be interesting to define new OLAP operators that take into consideration the specificities of data extracted from social media.

These operators will allow facilitating the interpretation of the results of the multidimensional analyses on the tweets and their metadata. We also expect to exploit the "Text Mining" techniques in order to extract knowledge from data and strengthen more semantics.

8 Declaration

- **Declaration of interests:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
- The authors declare the following financial interests/personal relationships which may be considered as potential competing interests
- **Financial interests:** The authors have no relevant financial or non-financial interests to disclose.
- The authors have no conflicts of interest to declare that are relevant to the content of this article.
- All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.
- The authors have no financial or proprietary interests in any material discussed in this article.

References

1. **Kaplan, A. M., Haenlein, M. (2010).** Users of the world, unite! The challenges and opportunities of Social Media. Business Horizons, Vol. 53, No. 1, pp. 59–68. DOI: 10.1016/j.bushor.2009.09.003.
2. **Cuzzocrea, A., De Maio, C., Fenza, G., Loia, V., Parente, M. (2015).** Towards OLAP analysis of multidimensional tweet streams. DOLAP '15: Proceedings of the ACM Eighteenth International Workshop on Data Warehousing and OLAP, pp. 69–73. DOI: 10.1145/2811222.2811233.
3. **Chaudhuri, S., Dayal, U. (1997).** Data warehousing and OLAP for decision support. Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, pp. 507–508. DOI: 10.1145/253260. 253373.
4. **Rizzi, S. (2007).** Conceptual modeling solutions for the data warehouse. Data Darehouses and OLAP. Concepts, Architectures and Solutions, IGI Global, pp. 1– 26.
5. **Kimball, R. (1996).** The data warehouse toolkit: practical techniques for building dimensional data warehouses. John Wiley & Sons, Inc.
6. **Smith, T. (2009),** Conference notes - The social media revolution. International Journal of Market Research, Vol. 51, No. 4, pp. 559–561. DOI: 10.2501/S1470785309200773.
7. **Shankar, V., Hollinger, M. (2007).** Online and mobile advertising: current scenario, emerging trends, and future directions. Marketing Science Institute, Vol. 31, No. 3, pp. 206–207.
8. **Akar, E., Topçu, B. (2011).** An examination of the factors influencing consumers' attitudes toward social media marketing. Journal of Internet Commerce, Vol. 10, No. 1, pp. 35–67. DOI: 15332861.2011.558456.
9. **Bringay, S., Béchet, N., Bouillot, F., Poncelet, P., Roche, M., Teisseire, M. (2011).** Towards an on-line analysis of tweets processing. Database and Expert Systems Applications: 22nd International Conference, DEXA´11, Springer Berlin Heidelberg, pp. 154–161. DOI: 10.1007/978-3-642-23091-2_15.
10. **Liu, X., Tang, K., Hancock, J., Han, J., Song, M., Xu, R., Manikonda, V., Pokorny, B. (2012).** SocialCube: A text cube framework for analyzing social media data. 2012 International Conference on Social Informatics, pp. 252–259. DOI: 10.1109/ SocialInformatics.2012.87.

11. **Rehman, N. U., Mansmann, S., Weiler, A., Scholl, M. H. (2012).** Building a data warehouse for twitter stream exploration. 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, IEEE, pp. 1341–1348.

12. **Cuzzocrea, A., De Maio, C., Fenza, G., Loia, V., Parente, M. (2016).** OLAP analysis of multidimensional tweet streams for supporting advanced analytics. Proceedings of the 31st Annual ACM Symposium on Applied Computing, pp. 992–999. DOI: 10.1145/285 1613.2851662.

13. **Yangui, R., Nabli, A., Gargouri, F. (2016).** Automatic transformation of data warehouse schema to NoSQL data base: comparative study. Procedia Computer Science, Vol. 96, pp. 255–264. DOI: 10.1016/j.procs.2016. 08.138.

14. **Yangui, R., Nabli, A., Gargouri, F. (2017).** DW4SN: A Tool for dynamic data warehouse building from social network. Research in Computing Science, Vol. 134, pp. 191–205.

15. **Moulai, H., Drias, H. (2018).** From data warehouse to information warehouse: application to social media. Proceedings of the international conference on learning and optimization algorithms: Theory and applications. pp. 1–6. DOI: 10.1145/3230905. 3230914.

16. **Ferdaous, J., Gouider, M. S. (2022).** Large-scale system for social media data warehousing: the case of twitter-related drug abuse events integration. International Journal of Data Warehousing and Mining (IJDWM), Vol. 18, No. 1, pp. 1–18. DOI: 10.4018/IJDWM. 290890.

17. **Kraiem, M. B., Feki, J., Khrouf, K., Ravat, F., Teste, O. (2015).** Modeling and OLAPing social media: the case of twitter. Social Network Analysis and Mining, Vol. 5, No. 47, pp. 1–15. DOI: 10.1007/s13278-015-0286-9.

18. **Kraiem, M. B., Alqarni, M., Feki, J., Ravat, F. (2020).** OLAP operators for social network analysis. Cluster Computing, Vol. 23, pp. 2347–2374. DOI: 10.1007/s10586-019-03006- z.

19. **Moya, L. G., Kudama, S., Cabo, M. J. A., Llavori, R. B. (2011).** Integrating web feed opinions into a corporate data warehouse. Proceedings of the 2nd International Workshop on Business intelligencE and the WEB, pp. 20–27. DOI: 10.1145/1966883. 1966891.

20. **Costa, P. R., Souza, F. F., Times, V. C., Benevenuto, F. (2012).** Towards integrating online social networks and business intelligence. Proceedings of the international conferences web based communities and social media, pp. 21–32.

21. **Rehman, N. U., Weiler, A., Scholl, M. H. (2013).** OLAPing social media: The case of Twitter. Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining, pp. 1139–1146. DOI: 10.1145/2492517.2500273.

22. **Walha, A., Ghozzi, F., Gargouri, F. (2015).** ETL design toward social network opinion analysis. Computer and information science 2015, Springer International Publishing, pp 235–249. DOI: 10.1007/978-3-319-23467-0_16.

23. **Djaballah, K. H., Boukhalfa, K., Bouassid, O. (2019).** Datawarehouse-based approach for the analysis of terrorism-related activities in social networks. Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS).

24. **Valêncio, C. R., Silva, L. M. M., Tenório, W., Zafalon, G. F. D., Colombini, A. C., Fortes, M. Z. (2020).** Data warehouse design to support social media analysis in a big data environment. Journal of Computer Science, pp. 126–136. DOI: 10.3844/jcssp.2020. 126.136.

25. **Gutiérrez-Batista, K., Vila, M. A., Martin-Bautista, M. J. (2021).** Building a fuzzy sentiment dimension for multidimensional analysis in social networks. Applied Soft Computing, Vol. 108. DOI: 10.1016/j.asoc. 2021.107390.

26. **Moalla, I., Nabli, A., Hammami, M. (2017).** Integration of a multidimensional schema from different social media to analyze customers' opinions. 2017 11th International Conference on Research Challenges in Information

Science (RCIS), IEEE, pp. 391–400. DOI: 10. 1109/RCIS.2017.7956564.

27. **Moalla, I., Nabli, A., Hammami, M. (2022).** Data warehouse building to support opinion analysis in social media. Social Network Analysis and Mining, Vol. 12, No. 1, pp. 123. DOI: 10.1007/s13278-022-00960-2.
28. **Zhang, L., Liu, B., Lim, S. H., O'Brien-Strain, E. (2010).** Extracting and ranking product features in opinion documents. Coling 2010: posters, pp. 1462–1470.
29. **Gallinucci, E., Golfarelli, M., Rizzi, S. (2013).** Meta-stars: multidimensional modeling for social business intelligence. Proceedings of the Sixteenth International Workshop on Data Warehousing and OLAP. ACM, pp 11–18. DOI: 10.1145/2513190.2513195.
30. **Gallinucci, E., Golfarelli, M., Rizzi, S. (2015).** Advanced topic modeling for social business intelligence. Information Systems, Vol. 53, pp. 87–106. DOI: 10.1016/j.is.2015.04.005.
31. **Francia, M., Golfarelli, M., Rizzi, S. (2014).** A methodology for social BI. Proceedings of the 18th International Database Engineering & Applications Symposium, pp. 207–216. DOI: 10.1145/2628194.2628250.
32. **Parama-Fadli, K. S. (2018).** Business intelligence model to analyze social media information. 3rd International Conference on Computer Science and Computational Intelligence, Vol. 135, pp. 5–14. DOI: 10.1016/ j.procs.2018.08.144.
33. **Girsang, A. S., Isa, S. M., Ginzel, M. E. C. (2020).** Implementation of a journalist business intelligence in social media monitoring system. Advances in Science, Technology and Engineering Systems Journal, Vol. 5, pp. 1517–1528. DOI: 10.25046/aj0506182.
34. **Mouyassir, K., Hanine, M., Ouahmane, H. (2021).** Business intelligence model to analyze social media through big data analytics. SHS Web of Conferences Vol. 119, pp. 07006 DOI: 10.1051/shsconf/202111907006
35. **Aramburu, M. J., Llavori, R. B., Lanza-Cruz, I. (2021).** Quality management in social business intelligence projects. ICEIS, Vol. 1, pp. 320–327. DOI: 10.5220/0010495703 200327.
36. **Gupta, A., Kumaraguru, P., Castillo, C., Meier, P. (2014).** Tweetcred: real-time credibility assessment of content on twitter. Proceedings of the 6th International Conference on Social Informatics, pp. 228–243. DOI: 10.1007/978-3-319-13734-6_16.
37. **Lanza-Cruz, I., Berlanga, R., Aramburu, M. J. (2023).** Multidimensional author profiling for social business intelligence. Information Systems Frontiers, pp. 1–21. DOI: 10.1007/ s10796-023-10370-0.

Article received on 04/08/2023; accepted on 20/04/2023.
[*] *Corresponding author is Maha Ben Kraiem.*

ISSN 2007-9737

Real-Time Helmet Detection and Number Plate Extraction Using Computer Vision

Jyoti Prakash-Borah, Prakash Devnani, Sumon Kumar-Das,
Advaitha Vetagiri, Partha Pakray*

National Institute of Technology, Silchar,
India

{jyoti20_ug, prakash20_ug, sumon20_ug,
advaitha21_rs, partha}@cse.nits.ac.in

Abstract. In the contemporary landscape, two-wheelers have emerged as the predominant mode of transportation, despite their inherent risk due to limited protection. Disturbing data from 2020 reveals a daily toll of 304 lives lost in India in road accidents involving two-wheeler riders without helmets, emphasizing the urgent need for safety measures. Recognizing the crucial role of helmets in mitigating risks, governments have made riding without one a punishable offense, employing manual strategies for enforcement with limitations in speed and weather conditions. In today's world of advancing technology, we can leverage the power of computer vision and deep learning to tackle this problem. This can eliminate the need for constant human surveillance to be kept on riders and can automate this process, thus enforcing law and order as well as making this process efficient. Our proposed solution utilizes video surveillance and the YOLOv8 deep learning model for automatic helmet detection. The system employs pure machine learning to identify helmet types with minimal computation cost by utilizing various image processing algorithms. Once the helmet-less person is detected, the number plate corresponding to the rider's motorcycle is also detected and extracted using computer vision techniques. This number plate is then stored in a database thus allowing further intervention to be done in this matter by the authorities to ensure penalties and enforce safety rules properly. The model developed achieves an overall accuracy score of 93.6% on the testing data, thus showcasing good results on diverse datasets.

Keywords. Image dataset, YOLOv8, deep learning model, object detection, image processing algorithms.

1 Introduction

The field of Artificial Intelligence (AI) encompasses a diverse range of technologies and applications, with its roots in creating intelligent systems that can perform tasks that typically require human intelligence.

One prominent subfield, Computer Vision, focuses on endowing machines with the ability to interpret and understand visual information from the world, opening up possibilities for applications in image analysis, video processing, and augmented reality.

Within the realm of Computer Vision, the You Only Look Once (YOLO) algorithm stands out as a groundbreaking approach to object detection. YOLO's innovation lies in its unified, real-time processing capabilities, achieved through a single neural network that can simultaneously predict bounding boxes and class probabilities for objects within an image.

The influential paper introducing YOLO, authored by Joseph Redmon and Santosh Divvala in 2016 [16], has since garnered widespread attention and has become a foundational reference in the field of computer vision, influencing subsequent developments and applications of object detection technologies. The increasing population of India in the last 30 years is leading to the use of more vehicles.

doi: 10.13053/CyS-28-1-4906

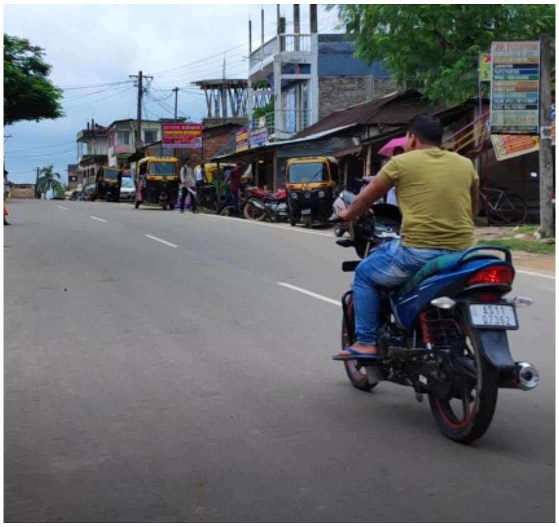


Fig. 1. A biker without helmet

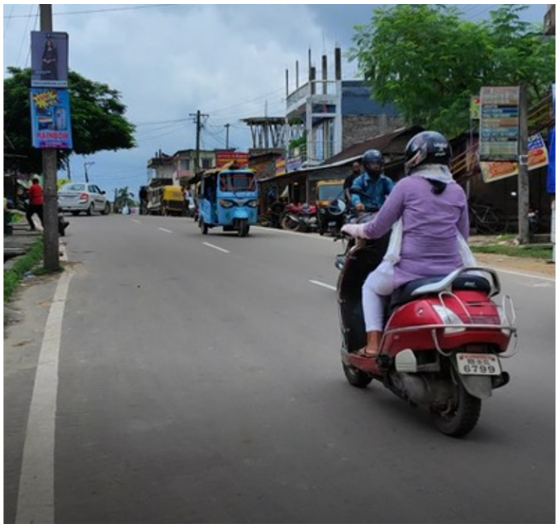


Fig. 2. A biker with helmet

According to Statista, as of 2023, the current population of India is 1.429 billion[1]. A study by Financial Express found that the majority of India's population is middle-class, which is about 31% of the population in 2020–2021 and is expected to rise to 61% by 2046–47[2], a two-wheeler is the most sought-after vehicle in India.

Two-wheeler domestic sales rose from 13.57 million in the financial year 2022 to 15.86 million in the financial year 2023[3], as suggested by data from Statista. The increasing use of two-wheelers without helmets and reckless driving is leading to the deaths of riders. A news article by the Times of India shows that, in the year 2021, 47,000 Indians died in two-wheeler accidents due to not wearing helmets[4].

Head injuries sustained by riders who do not wear helmets are a major cause of these deaths. Addressing this issue requires a comprehensive approach that combines technology and law enforcement. A study shows that using surveillance cameras in traffic has led to decreased road accidents.

Times of India reports that, in the state of Kerala in India, the use of surveillance cameras led to a decrease in accidents from 1,669 deaths in road accidents from June 5, 2022, to October 31, 2022. However, it dropped to 1,081 during the same period in 2023 after the installation of AI cameras[5].

A study found out that wearing helmets lowers the death rate chances by 37% and the head injury rate chances by 69% [10]. So there is a need to automate the process of helmet detection for proper law enforcement and to reduce deaths by two-wheelers.

The implementation of an automated system for monitoring helmet usage and identifying license plate numbers of non-compliant two-wheelers is a crucial step toward enhancing road safety. AI and computer vision algorithms can analyze real-time CCTV camera footage, enabling the detection of riders without helmets and the retrieval of their license plate numbers.

Our approach will be using state of the art YOLOv8 [6] model to extract the number plates of the without-helmet bike riders and store them in a database. This information can then be used to enforce helmet usage regulations and educate riders about the importance of helmet safety.

[1] Statista: www.statista.com/statistics/263766/total-population-of-india/

[2] Financial Express: tinyurl.com/56thtnfh

[3] Two-wheeler: www.statista.com/statistics/318023/two-wheeler-sales-in-india/

[4] Times of India: tinyurl.com/mw9advuw

[5] Times of India: tinyurl.com/2twhyy25

[6] YOLOv8: github.com/ultralytics/ultralytics

2 Related Work

Numerous domains, including pose detection, decision-making, self-driving vehicles, computer vision, and digital image processing techniques. The use of deep learning models has demonstrated success in a variety of fields, including healthcare [18], social sciences [11], earth sciences [2], etc.

R. Meenu et al. [12], carried out research where they were performing helmet detection and number plate extraction using Faster Region-Convolutional Neural Network (Faster R-CNN). They used CCTV footage and then split it into frames for analysis. Their methodology was split into four stages: motorcycle detection, head detection, helmet detection, and then number plate detection.

They utilized image processing algorithms like the Gabor wavelet filter to get accurate head positions. They achieved an accuracy of around 92%, depending on the quality of the CCTV cameras. However, cases of false detection are not addressed in the solution. Kunal Dahiya et al. [3] applied algorithms like background subtraction to detect only moving motorcycles and deal with false detection rates.

They also used Gaussian models to deal with various environmental detection challenges. Further, after extracting the foreground layer, many image processing algorithms were applied, like a noise filter and a Gaussian filter, and a binary image was obtained. Furthermore, objects were detected only based on a threshold area range that can be likely classified as a motorcycle.

They used techniques like Histogram of Oriented Gradients (HOG) and scale-invariant feature transformation for feature extraction. For classification, they used a Support Vector Machine(SVM). To remove false detection, they also consolidated the results using the information from the past frames.

They achieved a frame processing time of 11.58 ms and a frame generation time of around 33 ms, implying high efficiency. However, there is a lack of comprehensive evaluation on a diverse range of datasets, thus limiting the generalizability of the results.

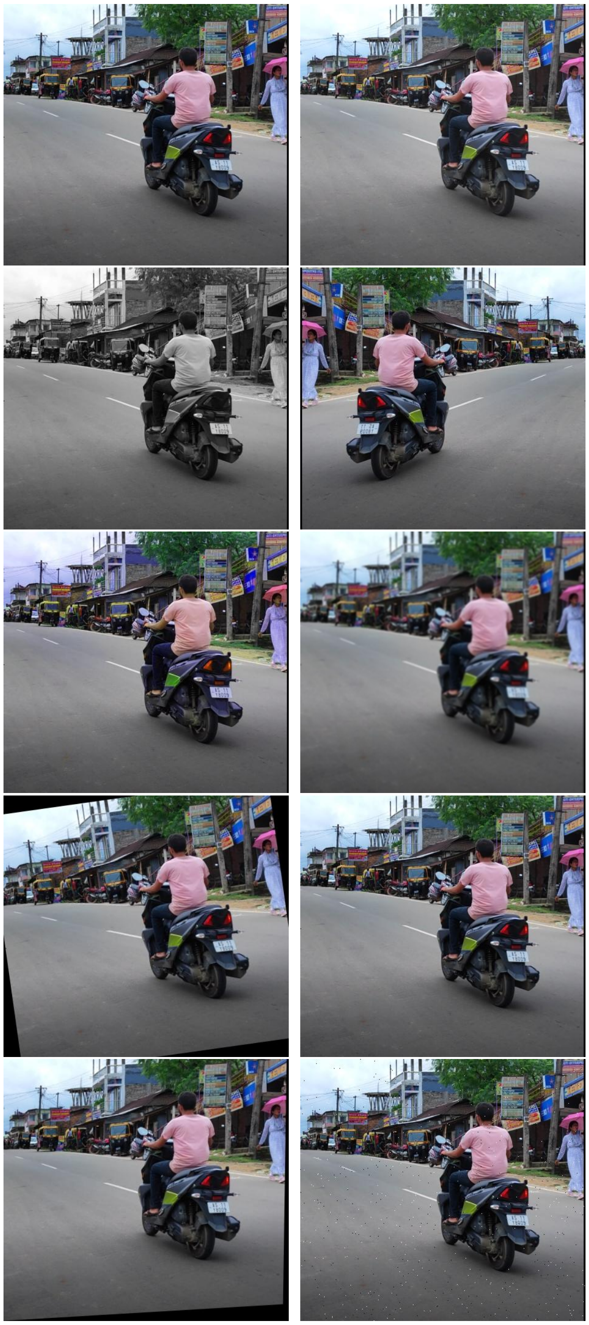


Fig. 3. Example of the original image and various image augmentations applied

Pushkar Sathe et al. [17] used yolov5 for helmet detection with an accuracy of 0.995 mean Average Precision(mAP) score [15]. They are using two methods to check if the rider is wearing a helmet.

Table 1. Total number of images from different sources

Sources	Total Images
Outside the campus	12 videos collected
Online sources including Google and news articles	3600
Data from the private repository of Roboflow	3155

Firstly, they check with the help of overlapping boxes of the helmet, numberplate, and the person and verify through a set of conditions if the person is wearing a helmet or not. The second method uses a range of motorcycle coordinates to check for helmets. Finally, they are using EasyOCR for character recognition of number plates.

However, this suffers from the lack of inclusion of a diverse dataset to make the model more generalizable. J Mistry et al. [13] used YOLOv2 for first detecting persons in a frame, citing the better performances in detecting a person rather than a motorcycle of the model. It then proceeds to detect the helmet, and if it is not found, then it goes for the number plate.

For no number plate detected, the model infers that the person detected is a pedestrian. The model achieved a 0.9470 value accuracy for helmet detection. However, this model also suffers from generalizability as not all cases of numberplates, riders, and helmet positions are discussed. M.M. Shidore and S.P. Narote [19] worked on techniques for efficient and accurate extraction of number plates from vehicles.

They used image processing techniques like histogram equalization and grey-scale conversions to deal with low-resolution images. Candidate number plate areas were extracted, and then true number plate areas were extracted. Character regions were enhanced, and background pixels were weakened.

Further character segmentation is done to get information about each number plate character. Then, finally, SVM was used to classify each character properly. The final results showcased an accuracy of around 85%. However, there is no mention of the dataset used for training and testing the system, which could be a limitation in evaluating the performance of the proposed approach. Waranusast et al. (2013) [21], in their work suggested a four step process to automatically identify motorcycles and determine whether they are wearing helmets or not.

Utilizing machine vision methodologies, the system employs algorithms to extract dynamic entities from the scene, distinguishing between motorcycles and other objects. Following this differentiation, it proceeds to enumerate and segment the heads of riders.

Subsequently, a comprehensive analysis is conducted to determine helmet usage, facilitated by a K-Nearest Neighbor (KNN) classifier. This classifier utilizes distinct features extracted from the segmented head regions to discern whether a helmet is present or not.

Through this iterative process, the system effectively identifies motorcyclists, segments their heads, and evaluates helmet compliance. However, the paper does not discuss the model performance under different lighting conditions or presence of occlussion.

Rupesh Chandrakant et al. (2022) [7] used a pre-trained model that uses the YOLO algorithm to detect whether the rider is wearing a helmet or not. Weights were tweaked as per the requirements. The authors created the dataset to ensure relevant data availability.

An accuracy of 96% and a frame detection time of around 1.35 sec were achieved. However, there is a lack of diversity in the dataset, including variations in lighting conditions, camera angles, and different types of helmets, which may limit the generalizability of the model.

V, Sri Uthra, et al. (2020) [20] presented significant findings where the paper proposed a motorcycle detection and classification method, helmet detection and helmet detection, and license plate recognition. Vehicle Classification was performed using an SVM classifier.

Helmet detection was done by applying Convolutional Neural Network (CNN) algorithms to extract image attributes, followed by classification using the SVM classifier. License plate recognition was done using Optical Character Recognition (OCR).

Algorithm 1: Extract Number Plate From the Frame

```
Data: YOLO predictions for image or frame: predictions, YOLO class names: classNames, YOLO class
      indexes: classIndexes, Image dimensions: imageWidth, imageHeight
Result: Cropped license plate regions and extracted text stored in a database
Initialize empty arrays: motorcycleBboxes, licensePlateBboxes, withoutHelmetBboxes;
foreach object in predictions do
    Extract class index, $\mathrm{x\_center}$, $\mathrm{y\_center}$, $\mathrm{bbox\_width}$, $\mathrm{bbox\_height}$ of the object;
    Convert normalized YOLO coordinates to image coordinates using imageWidth and imageHeight;
    Calculate bounding box lower left and upper right corner points ($\mathrm{x\_min}$, $\mathrm{y\_min}$, $\mathrm{x\_max}$, $\mathrm{y\_max}$);
    if classNames[class index] is "motorcycle" then
        Append ($\mathrm{x\_min}$, $\mathrm{y\_min}$, $\mathrm{x\_max}$, $\mathrm{y\_max}$) to motorcycleBboxes;
    else
        if classNames[class index] is "licensePlate" then
            Append ($\mathrm{x\_min}$, $\mathrm{y\_min}$, $\mathrm{x\_max}$, $\mathrm{y\_max}$) to licensePlateBboxes;
        else
            if classNames[class index] is "withoutHelmet" then
                Append ($\mathrm{x\_min}$, $\mathrm{y\_min}$, $\mathrm{x\_max}$, $\mathrm{y\_max}$) to withoutHelmetBboxes;
            else
foreach motorcycleBbox in motorcycleBboxes do
    foreach licensePlateBbox in licensePlateBboxes do
        foreach withoutHelmetBbox in withoutHelmetBboxes do
            if CheckInsideBoundingBox(licensePlateBbox, motorcycleBbox) and
              CheckInsideBoundingBox(withoutHelmetBbox, motorcycleBbox) then
                Crop the license plate region from the image;
                Send the cropped license plate to an OCR for text extraction;
                Store the extracted text in a database;
```

The system utilized background subtraction and feature extraction using Wavelet Transform. The accuracy for motorcycle classification is 93%, for helmet classification is 85%, and license plate recognition is about 81%. The paper, however, didn't mention computational requirements.

Adil Afzal et al. (2021) [1] introduce a deep learning-based methodology for the automatic detection of helmet wear by motorcyclists in surveillance videos. Leveraging the Faster R-CNN model, the approach involves two phases: helmet detection using the Region Proposal Network (RPN) and subsequent recognition of the detected helmets.

Trained on a self-generated dataset from three distinct locations in Lahore, Pakistan, the methodology achieves a notable 97.26% accuracy in real-time surveillance video analysis. Its strengths lie in the effective utilization of deep learning techniques, the accuracy afforded by the Faster R-CNN model, and the realism added by the use of a self-generated dataset from actual surveillance footage.

However, limitations include the lack of detailed information on addressing challenges like low resolution and varying weather conditions, limited generalizability to other locations or datasets, and a lack of discussion on the computational requirements and scalability of the proposed methodology.

Further, Mamidi Kiran Kumar et al. (2023) [9] use the YOLO Darknet deep learning framework to automate the detection of motorcycle riders wearing helmets from images, simultaneously triggering alerts for non-compliance. Through bounding boxes and confidence scores, the model identifies regions of interest like riders, helmets, and number plates.

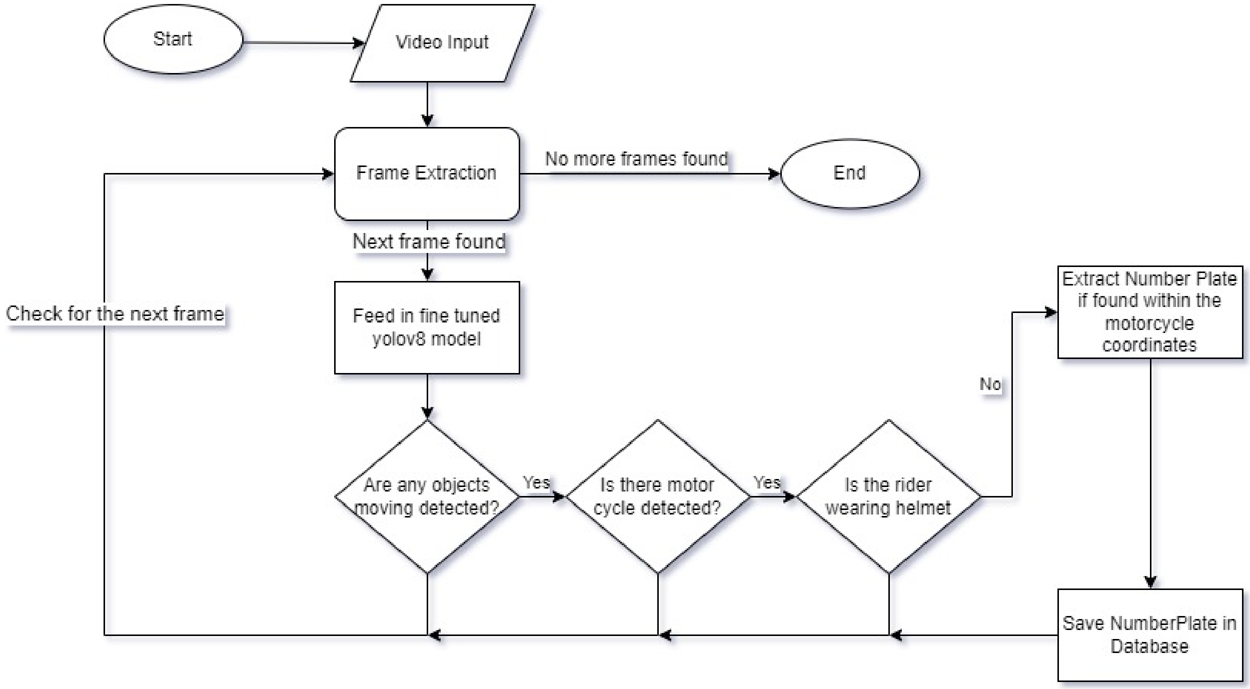

Start
Video Input
Frame Extraction
No more frames found
End
Next frame found
Check for the next frame
Feed in fine tuned yolov8 model
Extract Number Plate if found within the motorcycle coordinates
No
Are any objects moving detected?
Yes
Is there motor cycle detected?
Yes
Is the rider wearing helmet
Save NumberPlate in Database

Fig. 4. Flowchart of the proposed solution

The dataset used for training encompasses a diverse collection of images with 80 object categories, capturing a broad spectrum of real-world scenarios. The strengths of the model lie in its automated and efficient solution for helmet detection, eliminating the need for manual checks, and its utilization of the YOLO Darknet framework, enabling real-time detection and alert generation.

However, the limitations include the absence of detailed information on performance metrics or evaluation results, making it challenging to assess the model's accuracy, and a lack of specificity about the training dataset, raising concerns about its representativeness and potential biases.

3 Dataset

For our work we collected various images by ourselves and annotated them, we used state-of-the-art YOLOv8 for its ability to detect images in a single pass, its speed and efficiency in object detection tasks, we fined-tuned it.

In the subsequent sections, we provide detailed insights into our fine-tuning methodology, including the selection of hyperparameters, the augmentation strategies employed, and the evaluation metrics used to assess the model's performance. Our objective was to harness the power of YOLOv8 to deliver precise and efficient object detection for our application.

3.1 Dataset Statistics

Our dataset[7] was compiled based on both online sources and self-collected data. Since there was no such public repository for bike riders' images, we scrapped various news articles to get images of interest. Figure 2 shows samples of images from our dataset. First, a total of 3155 images were sourced from online, enriching our dataset with diverse visual data for comprehensive model training. These images were already annotated to serve our purpose.

[7] Dataset Link: github.com/Jyoti764/Helmet-Violation-Detection-Dataset

Table 2. Evaluation Metrics

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95	Correct Instances
all	726	2600	0.932	0.907	0.936	0.751	2402
licensePlate	726	762	0.946	0.966	0.964	0.755	737
motorcycle	726	819	0.924	0.939	0.952	0.845	778
withHelmet	726	686	0.902	0.834	0.887	0.672	586
withoutHelmet	726	333	0.955	0.888	0.939	0.733	301

Further, we collected 12 videos from outside the National Institute of Technology, Silchar, campus. The images were then annotated using the Roboflow [8] online annotation tool. Also, various image augmentation techniques were applied so that we could further diversify our dataset and ensure the model remained robust and had good generalizability.

Techniques like flips, rotation, blur, and adjusting the values of RGB channels were employed to achieve a total of 3600 images among the self-collected data. In total, we amassed a total of 6755 images, among which 3600 were self-collected and self-annotated, and 3155 were outsourced from Roboflow as in 1.

3.2 Augmentations Applied on Images

We used various augmentation techniques to improve the training and diversify the dataset.

We applied horizontal flip, coloured images were augmented to grayscale images to simulate nighttime CCTV video feeds or images, rotation was applied with a magnitude between -15° to +15° shear was done randomly with a magnitude between -16° to +16° in the horizontal direction and -23° to +23° in the vertical direction, hue and saturation of the images were changed between -25° to +25° gaussian blur was applied to the extent of 0.75 pixels, brightness of images were changed between -25% to +25% and lastly noise was added to 5% of the pixels.

Figure 3 shows various augmentations applied on a sample image from the dataset.

3.3 Dataset Annotation and Validation

We utilized the online Roboflow annotation tools to label the images nicely in YOLO format for the image annotation. This annotation format is useful for object detection tasks, as it divides the image in a grid and assigns bounding boxes to objects in those grid cells.

The Roboflow annotation tools provided us with an interactive interface to accurately mark and label objects of interest in the images. Also, we used Roboflow's generation tools to apply augmentations. For dataset validation, we inspected and verified the annotated dataset using the built-in validation features of the Roboflow tool.

The tool provides a visual graphic of the annotations, allowing us to quickly verify the completeness of the labelled objects. This manual validation step was important for ensuring the dataset's quality and removing potential errors in annotations.

4 Methodology

Our first step involves segmenting the video into consecutive frames and then applying some image processing techniques for better inference. We are using the Open Source Computer Vision Library (OpenCV)[9] library to first read the video as consecutive frames.

Then, for each frame, we are first resizing the frame to the YOLO input standard size (480) and then applying the following transformations:

[8] Roboflow: roboflow.com/

[9] OpenCV: github.com/opencv/opencv

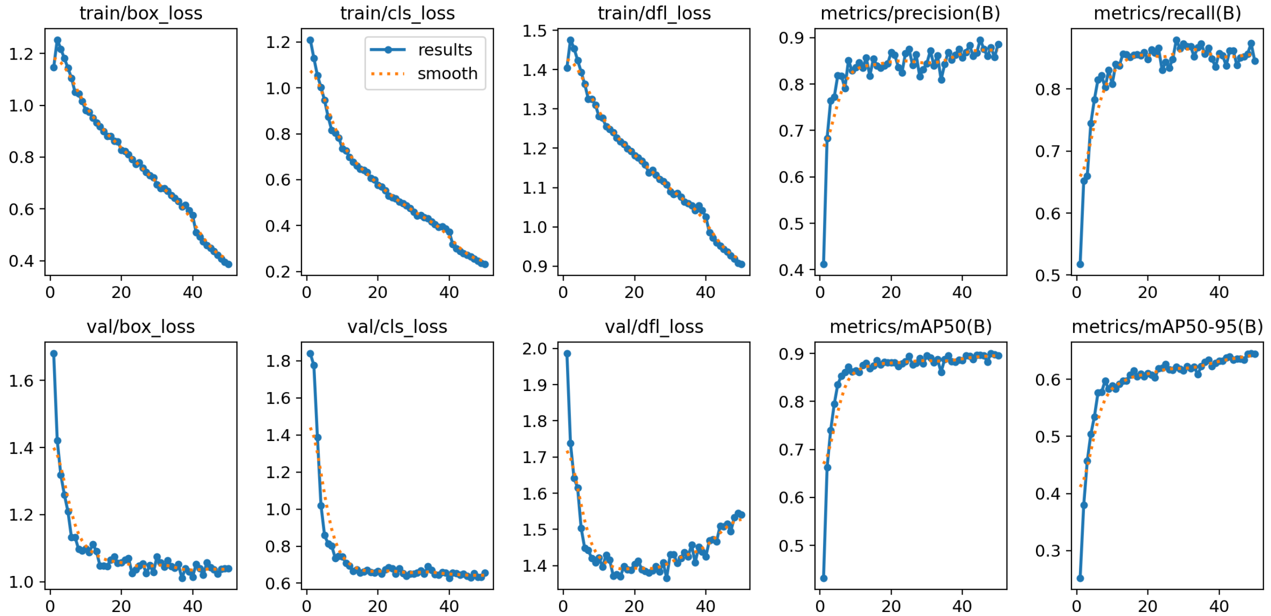


Fig. 5. Training and Validation Metrics

- Grayscale conversion: Grayscale conversion uses the values of the RGB channel and then calculates the pixel value using the following formula:

$$\text{Grayscale} = 0.299\ \text{R} + 0.587\ \text{G} + 0.114\ \text{B}. \qquad (1)$$

- Histogram Equalization [4]: This method is performed after grayscale conversion. It is a method that improves an image's contrast to stretch out the intensity range.

Equalization implies mapping one distribution (the given histogram) to another distribution (a wider and more uniform distribution of intensity values) so the intensity values are spread over the whole range. The remapping should be the cumulative distribution function to accomplish the equalisation effect. To use this as a remapping function, we have to normalize such that the maximum value is 255.

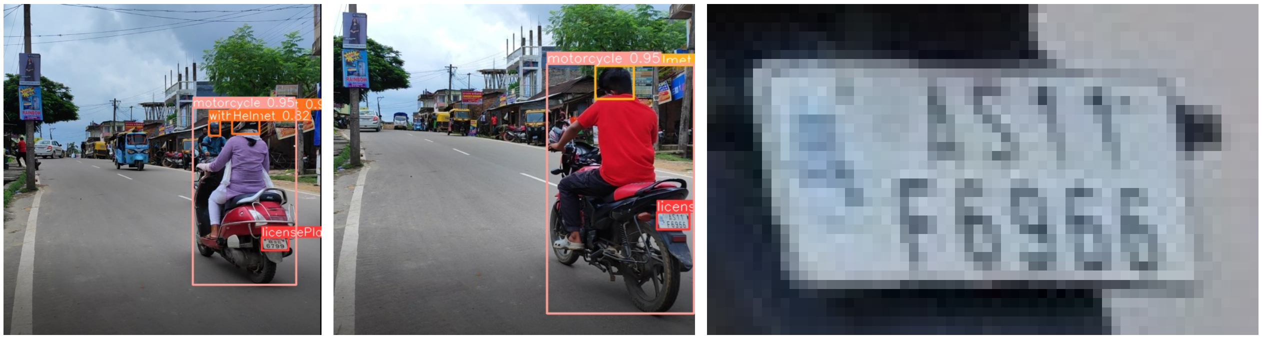
- Gaussian blur [5]: This blur focuses on taking a weighted mean, where neighbourhood pixels that are closer to the central pixel contribute more "weight" to the average. This generally helps in removing noise from our image.

Then, we are using background substraction [6] to separate the forward mask from the image. Then, to enhance the quality of the mask, we apply morphological transformations and then extract the contours beyond a threshold. Thus, the first step is complete, as we have the bounding boxes of all the moving objects in the frame.

This ensures that, in any case, non-moving objects shouldn't get selected in a frame. Our second step involves passing the frame through our fine-tuned YOLOv8 model. To prevent repeated boxes from being sent, we are first taking the union of intersecting boxes. Then, all the bounding boxes of moving objects corresponding to that frame are sent to the model. The model detects the image in four classes of objects, namely: "Motorcycle", "WithHelmet", "WithoutHelmet", and "NumberPlate".

With reference to the algorithm 1 and figure 4, our model first stores the information of the bounding boxes of the number plates, motorcycles, and WithoutHelmet classes. Then, for every motorcycle's bounding box, we are only considering the top 40% section of the box, as this serves as the most likely region where we are going to find a rider's head.

ISSN 2007-9737



(a) (b) (c)

Fig. 6. Inferences from the model. (a) A biker with Helmet. (b) A Biker without Helmet. (c) Extracted Numberplate. Here (c) is the numberplate extracted from (b), i.e. without helmet biker

Then, for each without Helmet and Numberplate class, we check if both exist within the motorcycle's bounding boxes. If this is the case, then we are sure that one of the riders on the bike is not wearing a helmet, and the numberplate detected also belongs to that motorcycle itself. So, the Numberplate coordinates can be extracted and saved for further inference.

4.1 Model Parameters

We used Adam Optimizer [8] during the training process. The learning rate, momentum, and weight decay are set to 0.00125, 0.8, and 0.0005 for 104 weights and 0.0 for 97 weights, respectively. The number of epochs was 55, and the batch size was set to 16.

We evaluate the mean Average Precision of the object detection to measure the performance of our model. The Intersection Over Union (IOU) [14] threshold range for measuring the accuracy of predicted bounding boxes relative to ground truth has been set to 0.50 to 0.95, with an interval of 0.05.

5 Results

In object detection, precision, recall, and mAP are commonly used metrics to evaluate the performance of a model such as YOLO. Precision, recall, and mAP can be defined as follows:

Precision is a measure of the accuracy of positive predictions made by an object detection model. It is defined as the ratio of true positives to the total predicted positives. The precision formula for object detection is given by:

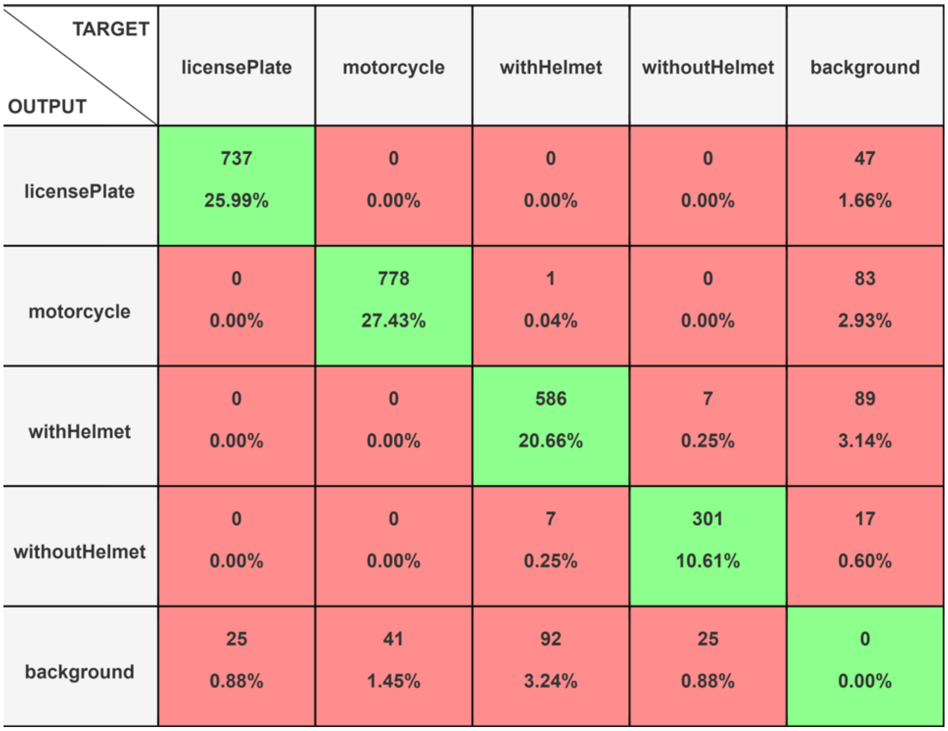
$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}. \qquad (2)$$

True Positives are correctly predicted positive instances, and false positives are those predicted as positive but actually negative. In the context of object detection, a "positive" prediction typically means the model correctly identified and localized an object of interest.

Recall, also known as sensitivity or true positive rate, is a measure of the ability of an object detection model to capture all relevant instances. It is defined as the ratio of true positives to the total actual positives. The recall formula for object detection is given by:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}, \qquad (3)$$

where true positives are the correctly predicted positive instances and false negatives are the instances that are actually positive but were predicted as negative. Recall helps assess how well the model captures all instances of the objects in the dataset. The mAP at IoU 0.5 is calculated by averaging the precision values at a specific IoU threshold (commonly set to 0.5) for each class.



TARGET OUTPUT	licensePlate	motorcycle	withHelmet	withoutHelmet	background
licensePlate	737 25.99%	0 0.00%	0 0.00%	0 0.00%	47 1.66%
motorcycle	0 0.00%	778 27.43%	1 0.04%	0 0.00%	83 2.93%
withHelmet	0 0.00%	0 0.00%	586 20.66%	7 0.25%	89 3.14%
withoutHelmet	0 0.00%	0 0.00%	7 0.25%	301 10.61%	17 0.60%
background	25 0.88%	41 1.45%	92 3.24%	25 0.88%	0 0.00%

Fig. 7. Confusion Matrix

The precision at IoU is calculated using the precision-recall curve. The formula is given by:

$$\text{mAP@50} = \frac{1}{C}\sum_{i=1}^{C} \text{AP}_i^{50}, \qquad (4)$$

where C is the total number of classes and AP_i^{50} is the Average Precision at IoU 0.5 for class i. The mAP from IoU 0.5 to 0.95 with a step of 0.05 is calculated by averaging the precision values over a range of IoU thresholds for each class. The precision at each IoU threshold is calculated using the precision-recall curve. The formula is given by:

$$\text{mAP@50:95} = \frac{1}{C}\sum_{i=1}^{C} \frac{1}{10}\sum_{t=50}^{95} \text{AP}_i^{t}, \qquad (5)$$

where C is the total number of classes, t represents the IoU threshold (from 50 to 95 with a step of 5), and AP_i^{t} is the Average Precision at IoU t for class i.

5.1 Testing Results

The mAP serves as a performance metric, with higher values generally indicating better overall object detection accuracy. Further analysis and adjustments may be considered to optimize and enhance model performance.

5.2 Training Results

The model training dataset comprises a total of 6755 images. The dataset is divided into three subsets: the testing, validation, and training sets. The testing set consists of 726 images, serving as a separate portion for assessing the model's performance. The validation set, consisting of 755 images, is employed for fine-tuning and parameter optimization during the training process.

The majority of the dataset, totalling 5274 images, forms the training set, providing the foundation for training the model to recognize and generalize patterns from the input images. Figure 5 shows metrics for training and validations.

Our model underwent evaluation on a diversity dataset containing a total of 726 images with a total of 2600 instances across all classes, achieving promising results across all classes. Figure 6 shows some of the inferences from our model. The overall performance, as indicated by the “all” class, demonstrated high mAP50 of 93.6% and mAP50-95 of 75.1%, contributing significantly to the robustness of the model.

The motorcycle class also exhibited strong performance, achieving a mAP50 of 95.2%. Additionally, the model performed well in identifying instances of withHelmet and withoutHelmet, showcasing its versatility in handling diverse scenarios in object detection tasks. Further, the overall performance metrics are shown in the Table 2 and figure 7. However, our model showed variations in performance across different classes.

Though licensePlate and motorcycle classes achieved outstanding results, the withHelmet and withoutHelmet classes showed lower precision and recall values, indicating potential room for optimization. The model speed, with preprocessing, takes 0.8 milliseconds, inference takes 29.2 milliseconds and postprocessing consumes 3.5 milliseconds per image, showing its efficiency in real-time applications.

In summary, our model with YOLOv8 architecture demonstrated high accuracy in detecting and localizing objects across multiple classes. The detailed class-wise metrics provide insights into the model’s strengths and areas for refinement, informing potential adjustments or fine-tuning strategies to enhance its overall performance.

6 Conclusion

This paper presented the development and evaluation of our fine-tuned YOLOv8 model for detecting without helmets bike riders and extracting their number plates. We employed various augmentation techniques to improve the accuracy and robustness of our model. The result shows a high mAP50 score of 0.936 on the testing data, correctly labelling the majority of the classes regardless of lighting and weather conditions of the images or videos showcasing the working of the model under diverse scenarios. Our model can also be efficiently deployed in real-time applications to monitor traffic in cities and highways. This model will help law enforcement agencies enforce laws on helmets properly and reduce the incidence of fatalities resulting from failure to wear helmets, undeniably contributing to saving lives.

Further improvements can be made by increasing the size of the dataset. We anticipate that our efforts will serve as a catalyst for additional investigations in this field, fostering the creation of models that are more precise and more efficient in enhancing safety for individuals on motorcycles, including riders, passengers, and fellow commuters on the road.

Acknowledgments

We extend our thanks to the Computer Science & Engineering Department at the National Institute of Technology Silchar for enabling us to pursue our research and experiments. Additionally, we acknowledge the support and resources offered by the Center for Natural Language Processing (CNLP) and Artificial Intelligence (AI) laboratories and the conducive research environment.

Declarations

Data Availability The authors declare that their data will be made available on request.

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. **Afzal, A., Draz, H. U., Khan, M. Z., Khan, M. U. G. (2021).** Automatic helmet violation detection of motorcyclists from surveillance videos using deep learning approaches of computer vision. Proceedings of the International Conference on Artificial Intelligence, pp. 252–257. DOI: 10.1109/ICAI52203.2021.9445206.

2. **Camps-Valls, G., Reichstein, M., Zhu, X., Tuia, D. (2020).** Advancing deep learning for earth sciences: From hybrid modeling to interpretability. IEEE International Geoscience and Remote Sensing Symposium, pp. 3979–3982. DOI: 10.1109/IGARSS39084.2020.9323558.

3. **Dahiya, K., Singh, D., Mohan, C. K. (2016).** Automatic detection of bike-riders without helmet using surveillance videos in real-time. Proceedings of the International Joint Conference on Neural Networks, pp. 3046–3051. DOI: 10.1109/IJCNN.2016.7727586.

4. **Garg, P., Jain, T. (2017).** A comparative study on histogram equalization and cumulative histogram equalization. International Journal of New Technology and Research, Vol. 3, No. 9.

5. **Gedraite, E. S., Hadad, M. (2011).** Investigation on the effect of a gaussian blur in image filtering and segmentation. Proceedings of the International Symposium on Electronics in Marine, pp. 393–396.

6. **Goyal, K., Singhai, J. (2017).** Review of background subtraction methods using gaussian mixture model for video surveillance systems. Artificial Intelligence Review, Vol. 50, No. 2, pp. 241–259. DOI: 10.1007/s10462-017-9542-x.

7. **Jaiswal, R., Srushti, C., Deo, V. (2023).** Helmet detection using machine learning. International Journal of Emerging Technologies and Innovative Research, Vol. 9, pp. d10–d17.

8. **Kingma, D., Ba, J. (2015).** Adam: A method for stochastic optimization. pp. 1–15. DOI: 10.48550/arXiv.1412.6980.

9. **Kiran-Kumar, M., Sanjana, C., Shireen, F., Harichandana, D., Sharma, M., Manasa, M. (2023).** Automatic number plate detection for motorcyclists riding without helmet. E3S Web of Conferences, Vol. 430, pp. 01038. DOI: 10.1051/e3sconf/202343001038.

10. **Liu, B. C., Ivers, R., Norton, R., Boufous, S., Blows, S., Lo, S. K. (2008).** Helmets for preventing injury in motorcycle riders. Wiley. DOI: 10.1002/14651858.cd004333.pub3.

11. **Lundberg, I., Brand, J. E., Jeon, N. (2022).** Researcher reasoning meets computational capacity: Machine learning for social science. Social Science Research, Vol. 108, pp. 102807. DOI: 10.1016/j.ssresearch.2022.102807.

12. **Meenu, R., Sinta, R., Smrithi, P. P., Swathy, S., Alphonsa, J. (2020).** Detection of helmetless riders using faster R-CNN. International Journal of Innovative Science and Research Technology, Vol. 5, No. 5, pp. 1616–1620.

13. **Mistry, J., Misraa, A. K., Agarwal, M., Vyas, A., Chudasama, V. M., Upla, K. P. (2017).** An automatic detection of helmeted and non-helmeted motorcyclist with license plate extraction using convolutional neural network. Proceedings of the 7th International Conference on Image Processing Theory, Tools and Applications, pp. 1–6. DOI: 10.1109/ipta.2017.8310092.

14. **Nowozin, S. (2014).** Optimal decisions from probabilistic models: The intersection-over-union case. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 548–555. DOI: 10.1109/cvpr.2014.77.

15. **Padilla, R., Netto, S. L., da-Silva, E. A. B. (2020).** A survey on performance metrics for object-detection algorithms. Proceedings of the International Conference on Systems, Signals and Image Processing, pp. 237–242. DOI: 10.1109/iwssip48289.2020.9145130.

16. **Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016).** You only look once: Unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788. DOI: 10.1109/CVPR.2016.91.

17. **Sathe, P., Rao, A., Singh, A., Nair, R., Poojary, A. (2022).** Helmet detection and

number plate recognition using deep learning. IEEE Region 10 Symposium, pp. 1–6. DOI: 10 .1109/tensymp54529.2022.9864462.

18. **Shailaja, K., Seetharamulu, B., Jabbar, M. A. (2018).** Machine learning in healthcare: A review. Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, pp. 910–914. DOI: 10.1109/ICECA.2018.8474918.

19. **Shidore, M. M., Narote, S. P. (2011).** Number plate recognition for indian vehicles. International Journal of Computer Science and Network Security, Vol. 11, No. 2, pp. 143–146.

20. **Sri, U. V., Sariga, D. V., Vaishali, K. S., Padma, P. S. (2020).** Helmet violation detection using deep learning. International Research Journal of Engineering and Technology, Vol. 7, pp. 3091–3095.

21. **Waranusast, R., Bundon, N., Timtong, V., Tangnoi, C., Pattanathaburt, P. (2013).** Machine vision techniques for motorcycle safety helmet detection. Proceedings of the 28th International Conference on Image and Vision Computing New Zealand, pp. 35–40. DOI: 10.1109/IVCNZ.2013.6726989.

Article received on 16/11/2023; accepted on 29/01/2024.
** Corresponding author is Partha Pakray.*

ISSN 2007-9737

From Words to Paragraphs: Modeling Sentiment Dynamics in Notes from Underground with GPT-4 by Differential Equations Via Quantile Regression Analysis

Volkan Duran[1], Iskander Akhmetov[2,3],
Elman Hazar[4], Alexander Gelbukh[*,5], Ezgi Kaya[4]

[1] Iğdır University, Department of Psychology,
Türkiye

[2] Institute of Information and Computational Technologies, Almaty,
Republic of Kazakhstan

[3] Kazakh-British Technical University, Almaty,
Republic of Kazakhstan

[4] Iğdır University, Department of Mathematics,
Türkiye

[5] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

volkan.duran8@gmail.com, i.akhmetov@kbtu.kz, gelbukh@cic.ipn.mx,
{elman.hazar, ezgi.kaya}@igdir.edu.tr

Abstract. This study examines how the sentiment values in the first part of the book entitled as “Underground” of Fyodor Dostoevsky’s ”Notes from Underground” change from words to sentences to paragraphs. Using the GPT-4 language model, we conducted a descriptive analysis of standardized sentiment values and calculated cumulative binned values of the sentiment trajectories over the text. We then created differential equation models to model the sentiment tones using quantile regression analysis. We show that binned values can reveal a more dynamic and potentially chaotic structure when applied to the cumulative sum of sentiments for word, sentence, and paragraph levels. We model differential equations derived for word, sentence, and paragraph levels via quantile regression. They demonstrate how the rate and acceleration of sentiment change are influenced by their current state and rate of change. In conclusion, this study’s findings are important for enhancing the capabilities of AI-driven chatbots in sentiment analysis, particularly in dissecting and understanding the layered emotional landscapes of literary works.

Keywords. Sentiment analysis, differential equations, GPT-4, curve fitting, quantile regression analysis.

1 Introduction

Opinion mining or sentiment analysis (SA) examines opinions in text using a blend of mathematics and linguistics [22]. It offers valuable insights for enhancing educational practices [11]. SA operates mainly at four levels: Document, Sentence, Phrase, and Aspect [20, 26].

doi: 10.13053/CyS-28-1-4905

Document level classifies the overall sentiment of a text, while Sentence level focuses on individual sentences. Phrase level mines opinion words and Aspect level analyzes the emotional components of phrases, assigning polarity to each.

Sentiment analysis is a multifaceted field involving various NLP tasks like aspect extraction and sarcasm detection [27]. It employs diverse techniques, including machine learning, lexicon-based, rule-based, and statistical models [10, 17, 23, 28]. Specialized methods like aspect-based analysis and deep learning have also been developed [12, 2, 21, 25, 29].

Moreover, multi-modal algorithms are emerging that analyze not just text but also visual data [5]. Sentiment analysis is already applied in diverse sectors like marketing, politics, and healthcare [6, 7, 9, 15]. By fusing AI-driven sentiment analysis with mathematical models, this research sets the stage for deeper exploration into sentiment dynamics, enriching its application across various fields.

Recent research suggests that keyword-based techniques may be inadequate for nuanced texts [19]. In the literature, some researchers focus on the ratings and reviews for the sentiment analysis, and it is the most fundamental part of this area. For instance, by using lemmatization, stemming techniques, and eliminating the stop words so that the data from the dataset are classified using logistic regression approach [18].

Additionally, subdividing the training corpus by topic (local news, sports, hi-tech, and others) and training separate sentiment classifiers for each sub-corpus improves classification F1 scores can also be used as topic-aware sentiment analysis of news articles [1]. The article is different from most of the previous literature by utilizing the GPT-4 language model for a descriptive analysis.

And of standardized sentiment values and calculating cumulative binned values of sentiment trajectories. It uses differential equation models and quantile regression analysis to model sentiment tones, a method that's more complex and potentially better suited for capturing the nuanced changes in sentiments in literary texts.

Although differential equations have previously been used in social sciences [16], the contribution of this research lies in building sentiment models through quantile regression analysis although There are some studies creating sentiment models through linear regression and curve fitting options [13, 14].

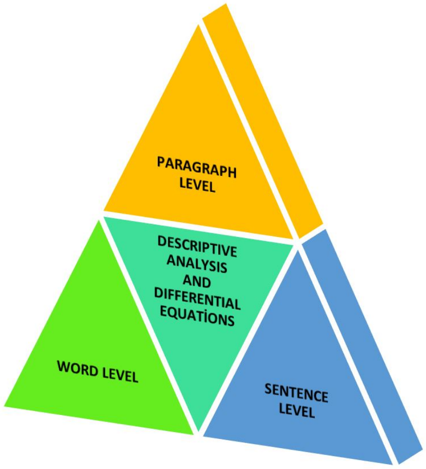


Fig. 1. The main units of the analysis

This approach not only enhances credibility but also allows for the study of complex sentiment relationships across various textual levels. It opens up avenues for predicting sentiment behavior in different contexts.

Given the intricacies of text sentiment representation and the intersection of AI-driven sentiment analysis with mathematical models, it is evident that understanding sentiment behavior in various contexts is not only crucial but intricate.

Drawing on the principles of mathematical modeling and physics, this research takes innovative steps in employing techniques from stratified symbolic regression, genetic programming, and the finite difference method.

Such techniques have proven instrumental in extracting differential equations from data, as showcased by many researchers [3, 4, 8, 24]. By bridging the gap between AI sentiment analysis and mathematical modeling, this research promises to provide a more credible, predictive, and enriched understanding of sentiment behavior across textual forms. Therefore, research on the development of sentiment representation using

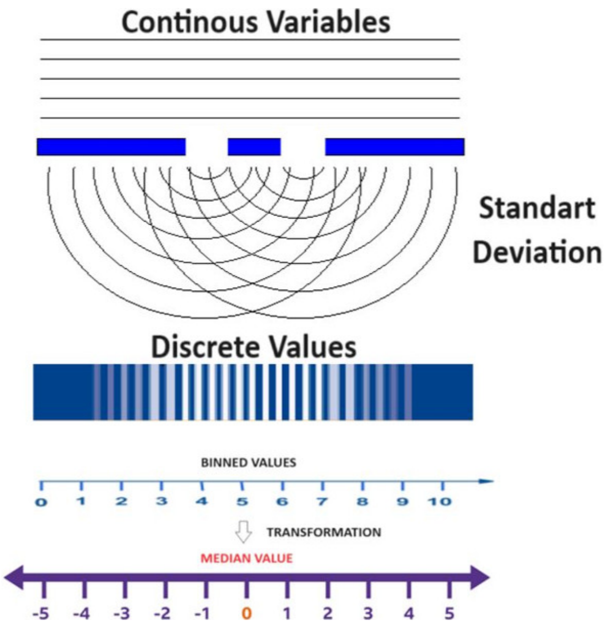


Fig. 2. Transformation of binned valued into sentimental range via median value

Table 1. Classification results using PumaMedNet-CXR and ResNet-18

	Category	Word	Sentence	Paragraph
Negative	Very low= -2	19.4	19.0	21.4
	Low= -1	28.4	35.3	23.2
	Total	47.9	54.3	44.6
Positive	High=1	33.5	24.0	51.8
	Very High=2	18.7	21.5	3.6
	Total	52.1	45.5	55.4

AI-driven analysis combined with mathematical modeling is undeniably relevant.

2 Methodology

This study is based on a quantitative research design. We analyzed the sentiments in terms of Word level, sentence level, and paragraph level (Figure 1) in the first part of the book entitled “Underground”.

In the first part of the study, we descriptively investigated the general characteristics of the sentiments in standardized forms. Finally, we used quantile regression models to get differential equations regarding the sentimental tones by using SPSS 25.

We get the three given equations representing the sentiment points at different levels of text (word, sentence, and paragraph) as a function of x. The x variable could be interpreted as the position within the text. We used GPT-4, which is a multimodal large language model created by OpenAI and the fourth in its GPT series, to label sentiment values at the word, sentence, and paragraph levels.

In this analysis, we have three main units of the research as words, sentences, and paragraphs (Figure 1) where GPT4 assigned sentiment scores between -1 and 0 (negative sentiments) and 0 and 1 (positive sentiments) to each word/phrase or to each sentence in a passage out to an entire passage of text.

2.1 Analysis

This study consists of two main parts. The first part involves an analysis of binned sentiment values based on the first standard deviation. Specifically, the cumulative sentiment time series is divided into bins separated by one standard deviation. This binning allows for examining the dynamics and potential chaos in the cumulative sentiment data. In the first part, the procedure can be given below:

1. **Descriptive Analysis of Sentiments:** The study starts by analyzing sentiments within the text, likely using GPT-4 or a similar tool to assess the sentiment of words, sentences, and paragraphs.
2. **Visual Binning Based on Standard Deviation:** They categorize sentiments into three levels based on the first standard deviation of the sentiment distribution. This approach effectively groups sentiments into categories like ’low’, ’medium’, and ’high’.
3. **Encoding Binned Variables as Integers:** Each sentiment level is then encoded as an integer (e.g., -2 for very low, -1 for low, 1 for high, 2 for very high). This quantifies the sentiment levels, making them easier to analyze numerically.

doi: 10.13053/CyS-28-1-4905

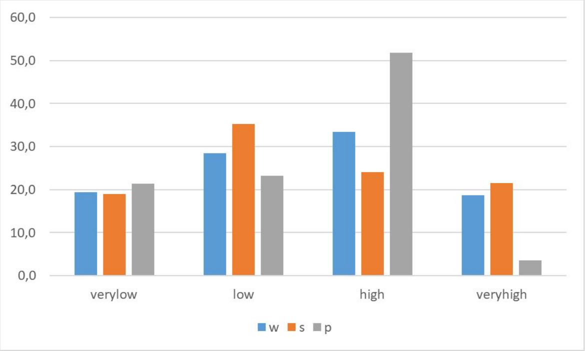


Fig. 3. The distribution of the percentages of the sentiments in different categories

Table 2. Classification results using PumaMedNet-CXR and ResNet-18

			Statistic	Std. Error
Wordlevel	Mean		-0.0868	.01238
	95% Confidence Interval for Mean	Lower Bound	-0.1111	
		Upper Bound	-0.0625	
	5% Trimmed Mean		-0.0892	
	Median		0.0000	
	Variance		0.195	
	Standard Deviation		0.44116	
	Minimum		-0.90	
	Maximum		0.80	
	Range		1.70	
	Interquartile Range		0.80	
	Skewness		0.024	0.069
	Kurtosis		-1.061	0.137

In this process, we focus on the median values of the binned categories, if there are an odd number of categories (n).

The median category, which is at position (n+1)/2, is considered the ’neutral’ or ’base’ sentiment and is assigned a value of 0. For categories below the median, negative integers are assigned, starting from -1 and decreasing for each category, moving away from the median. For categories above the median, positive integers are assigned, starting from 1 and increasing for each category moving away from the median.

If there is an even number of categories (n), the median is determined by averaging the n/2 and (n/2)+1th values. This average value represents the ’neutral’ sentiment and is assigned a value of 0 since the negative values.

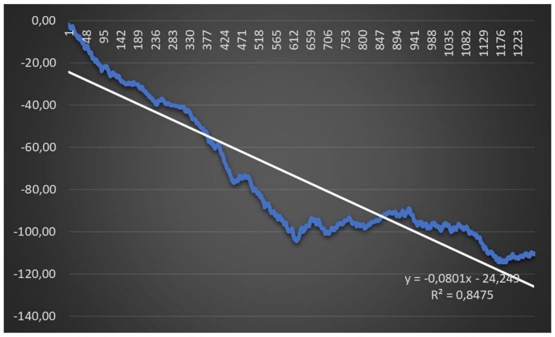


Fig. 4. The values of the water flow chart (cumulative sum) of the raw values of the sentiments of the words

But also correspond to low values of the binned variables hence in order to label them negative we did such a procedure in the raw data. Similar to the odd-numbered case, categories below this median are assigned negative integers, and those above are assigned positive integers.

This method of encoding allows for a more nuanced analysis of sentiment data as it preserves the ordinal nature of the sentiment levels while converting them into a format that can be easily used in various statistical and machine learning models.

It’s especially useful when dealing with sentiment analysis where the intensity or degree of sentiment is important (Figure 2). Binning is a process of transforming continuous data into categories or bins.

If the aim is to categorize data based on whether they fall within one standard deviation of the mean, we are essentially creating a non-linear partition of the data. We can make an analogy with wave-particle duality in the context of a double slit experiment with binding. In quantum mechanics, many physical properties, such as energy, angular momentum, and charge, are quantized.

This means they can only take on discrete values, much like how binning categorizes continuous data into discrete bins. The act of measuring a quantum system can ’bin’ the system into one of the possible states. Before measurement, quantum systems are described by a probability distribution (wave function), which encompasses many potential outcomes.

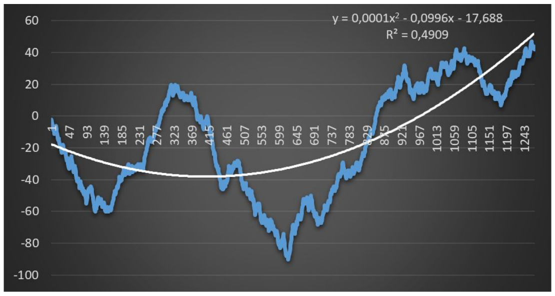


Fig. 5. The values of the water flow chart (cumulative sum) chart of the binned values sentiments of the words and the relevant equation

Measurement ’collapses’ this wave function, forcing the system into one of the distinct states, akin to assigning a data point to a specific bin. While these analogies can be helpful in visualizing some aspects of quantum mechanics, it’s crucial to remember that they are metaphorical (Figure 2). One problem with such methodology in the interpretation of the sentiments of text might be whether it really depicts the actual trajectories of the sentiments, but this can be remedied by comparative studies.

Comparative studies serve as a valuable tool to validate and refine this approach, ensuring a more accurate and nuanced understanding of sentiments in textual analysis. This method preserves the ordinal nature of sentiment data. The relative ordering (from very negative to very positive) is maintained, which is important for many statistical analyses and machine learning models that can leverage this order information.

By encoding sentiments this way, you can conduct more detailed and meaningful analyses of sentiment data, capturing not just whether sentiments are positive or negative but the degree of positivity or negativity.

This is particularly useful in areas like customer feedback analysis, social media sentiment tracking, and market research, where understanding the intensity of sentiments can be as important as knowing their direction.

4. **Cumulative Sum of Encoded Values:** The cumulative sum of these encoded sentiment values is computed. This means that for each point in the text (word, sentence, or paragraph), they add its sentiment score to the total of all previous scores. The result is a running total of sentiment values. The computation of the cumulative sum of these sentiment values represents the aggregation of sentiment over the text. In a literary context, this could reflect the buildup or fluctuation of emotional tone throughout the narrative. This step transforms the sentiment trajectory into a path.

5. **Interpreting Cumulative Sum:** In this phase, the cumulative sum of sentiment values is interpreted as a narrative sentiment progression. This approach considers the cumulative total as a reflection of the evolving emotional tone within the text. Each sentiment score at various textual levels - whether a word, sentence, or paragraph - contributes to an ongoing narrative sentiment trajectory.

   This trajectory, depicted as a cumulative sum, showcases how emotional tones build, shift, and fluctuate over the course of the narrative. Unlike the independent steps of a random walk, this sentiment progression is influenced by the contextual and sequential nature of the literary work, highlighting the interconnectedness and dependency of emotional expressions as the story unfolds. This interpretation provides insights into the nuanced and structured dynamics of sentiment in literature, illustrating how emotions evolve and interact throughout the narrative journey.

6. **Graphing and Curve Fitting:** The cumulative sentiment scores are then graphed, likely showing how sentiment evolves throughout the text. Curve fitting is applied to this graph to analyze the sentiment dynamics further.

In summary, by encoding sentiments as numerical values and accumulating these values over the course of the text, the authors transform the sentiment data into a format that can be analyzed, revealing insights about the sentiment dynamics in the text. The second part focuses on developing differential equation models using the raw, unbinned sentiment values. These differential equations relate the change in sentiment (first

Table 3. The descriptive values of the raw values of the sentiments of the sentences

			Statistic	Std. Error
Sentencelevel	Mean		-0.1400	.01593
	95% Confidence Interval for Mean	Lower Bound	-0.1713	
		Upper Bound	-0.1088	
	5% Trimmed Mean		-0.1442	
	Median		-0.2000	
	Variance		0.172	
	Standard Deviation		0.41489	
	Minimum		-0.90	
	Maximum		0.80	
	Range		1.70	
	Interquartile Range		0.70	
	Skewness		0.174	0.069
	Kurtosis		-0.959	0.187

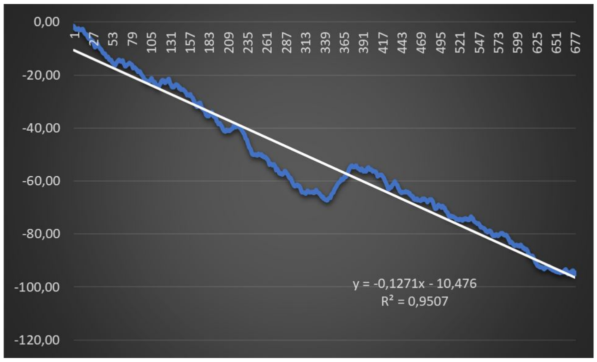


Fig. 6. The values of the water flow chart (cumulative sum) of the raw values of the sentiments of the sentences

derivative) and acceleration of sentiment (second derivative) to the current sentiment values.

By modeling the derivatives, the equations aim to capture the continuous evolution of sentiment through textual data. We used a quantile regression model to get differential equations regarding the sentimental tones. Creating a differential equation model using the difference method of variables, curve fitting, and linear regression involves several steps by using IBM SPSS 27 and the Excel program. Here's a general outline of the process:

1. **Data Collection:** We collected sentiment data generated by GPT-4, where the variable x can be interpreted as the position within the text.

2. **Calculation of Differences:** We computed the differences between consecutive data points to approximate derivatives such as the first and second derivatives. We used the finite forward difference method to calculate these numerical derivatives, denoted as metrics.

3. **Curve Fitting:** Curve fitting was performed on both the original sentiment data and the calculated differences.

4. **Quantile Regression Analysis:** Quantile regression is a statistical modeling technique that examines the association between a group of explanatory variables and particular percentiles, referred to as quantiles, of a response variable. The response variable is typically the median. There are two primary advantages associated with this method in comparison to Ordinary Least Squares regression. Quantile regression is a statistical method that does not rely on any assumptions about the underlying distribution of the dependent variable. Quantile regression exhibits a robustness against the impact of extreme observations. Quantile regression is extensively employed in various fields, including ecology, healthcare, and financial economics, for the purpose of research.

5. **Formulation of the Differential Equation:** Based on the results of curve fitting and quantile regression, a differential equation model was formulated.

   Coefficients from the regression were used to define the relationship between the dependent variable(s) and their derivatives in the differential equation.

6. **Final Equations:** We derived three equations representing sentiment at different textual levels (word, sentence, and paragraph) as functions for the position within the text.

2.2 Limitations

– The main limitation of this study is that we chose the English translation of the book rather than the original one (Notes from Underground

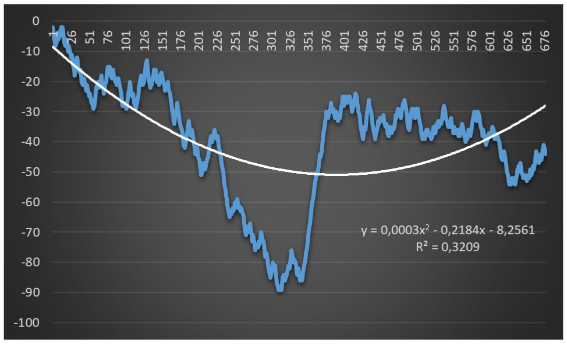


Fig. 7. The values of the water flow chart (cumulative sum) chart of the binned values of the sentiments of the sentences and the relevant equation

Table 4. The descriptive values of the raw values of the sentiments of the paragraphs

			Statistic	Std. Error
Paragraphlevel	Mean		-0.2330	0.03214
	95% Confidence Interval for Mean	Lower Bound	-0.2974	
		Upper Bound	-0.1686	
	5% Trimmed Mean		-0.2238	
	Median		-0.2000	
	Variance		0.058	
	Standard Deviation		0.24051	
	Minimum		-0.75	
	Maximum		0.15	
	Range		0.90	
	Interquartile Range		0.40	
	Skewness		-0.524	0.319
	Kurtosis		-0.921	0.628

(Vintage Classics) by Fyodor Dostoevsky (Author), Richard Pevear (Translator), Larissa Volokhonsky (Translator). Although GPT-4 works well with Russian, it is supposed that it can analyze the results best in English since the main aim is to analyze sentiments.

- The second limitation is that we use the GPT-4 model since there are a lot of different libraries and algorithms for this, so our results are restricted within the capabilities of the GPT-4 chatbot.

- Sentiment analysis and NLP face a number of obstacles, including idiosyncrasies in writing style, sarcasm, irony, and linguistic peculiarities. Many terms in many languages have nuanced or shifting meanings based on the specific setting or field in which they are used.

- Performing regression analysis on a variable and its numerical derivative based on a different method might not be ideal for several reasons like loss of information, amplification of noise, data requirements, assumption violations, non-stationarity, causality, and ınterpretation issues.

  However, there are cases where using derivatives in a regression analysis could be beneficial.

  For example, if someone is interested in the rate of change or if the relationship between variables is best modeled by considering rates of change, then the derivative might be appropriate.

- The encoding of sentiments into integers (-2 to 2) may lose some granularity of sentiment data. Literature often contains more complex emotions that this range might not fully capture.

- While cumulative sums can reveal overall trends, they might obscure local sentiment fluctuations. It’s important to balance the overall trajectory with local sentiment variations.

- The choice of curve fitting techniques and their interpretation can significantly influence the conclusions. It’s vital to ensure that the chosen method accurately reflects the sentiment dynamics.

3 Findings

3.1 General Descriptive Findings of the Cumulative Binned Values of the Sentiments

When we look at the sentiments at different levels, we observed the following results:

- **Negative Sentiments:** More prevalent at the sentence level (54.3%) compared to word (47.8%) and paragraph levels (44.6%). This suggests that negative sentiments are more distinctly identified or articulated in sentences.

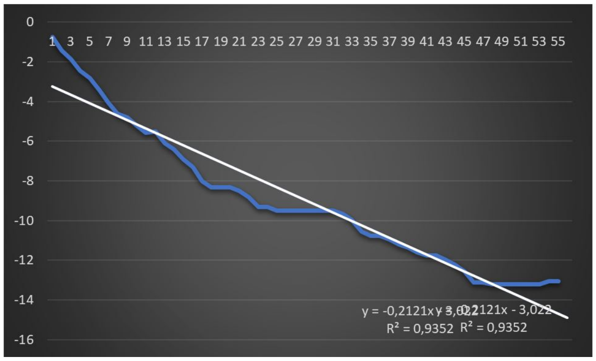


Fig. 8. The graph represents the standardized values of the water flow chart (cumulative sum) of the raw values of the sentiments of the paragraphs

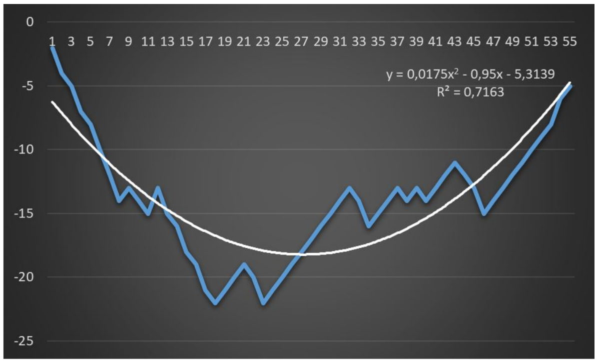


Fig. 9. The graph represents the standardized values of the water flow chart (cumulative sum) of the raw values of the sentiments of the paragraphs

– **Positive Sentiments:** Dominant at the paragraph level (55.4%), followed by the word level (52.2%) and sentence level (45.7%). This indicates that positive sentiments are more pronounced or become clearer in larger text contexts, such as paragraphs.

– **Word Level Analysis:** Shows a slightly higher occurrence of positive sentiments compared to negative sentiments.

– **Sentence Level Analysis:** Negative sentiments are more prominent than positive sentiments, indicating that sentences might convey negativity more distinctly.

– **Paragraph Level Analysis:** A significant tilt towards positive sentiments, suggesting that overall positivity is more likely to be perceived in longer text blocks.

Table 5. Observed Model Quality (q=0.5)[a,b,c]

Pseudo R Squared	0.510
Mean Absolute Error (MAE)	0.3562

a: Dependet Variable: d2word.
b: Model: (Intercept), word, dword.
c: Method: Symplex algorithm.

Table 6. Null Model Quality (q=0.5)[a,b,c]

Pseudo R Squared	0.000
Mean Absolute Error (MAE)	0.7273

a: Dependet Variable: d2word.
b: Model: (Intercept).
c: Method: Symplex algorithm.

We concluded that the context (word, sentence or paragraph) significantly influences sentiment perception. Negative sentiments are more pronounced in sentences, while positive sentiments are more likely to be identified in paragraphs and this data can imply that the nuance and complexity of sentiments become more apparent in larger textual contexts. Binning is a method used in data analysis to group a range of values into bins, or intervals, which can help in identifying trends in a dataset that may not be apparent when analyzing the raw data.

In summary, while raw cumulative sums provide a direct sequential aggregation of sentiment values, binned values can uncover a more nuanced, dynamic, and sometimes chaotic structure in sentiment data, showcasing trends and patterns that may not be immediately evident in the raw cumulative sum.

3.1.1 The Descriptive Interpretation of the Sentiment Values at Word Level

The descriptive values of the sentiment values at the word level show a generally negative sentiment at the word level (Table 2). The data suggests a slight overall negative tendency in the sentiment of words analyzed, but with a balanced median and a wide range of sentiment values.

The distribution of sentiment scores is fairly symmetrical and moderately varied, indicating a diverse set of sentiments in the words analyzed. This might reflect a dataset with a broad spectrum of emotional expressions, leaning slightly towards negative sentiment. The cumulative sentiment of the words analyzed decreases significantly over the series of data points. The high $\mathbb{R}^2$ value indicates that the trend is strongly consistent. This could imply that the data set or time period being analyzed is characterized by an increasing prevalence of negative sentiment.

If this were a time-based analysis, one could conclude that the overall sentiment is becoming more negative over time. If this represents a sequence of events or another type of series, it would suggest a downward trend in sentiment associated with the progression of that series (Figure 4). The binned analysis with a quadratic model shows that the cumulative sentiment of words has a more complex dynamic than a simple linear decrease. It highlights periods where the sentiment becomes more positive before turning more negative again.

The presence of a curve in the trend line and the variable distribution of the data points suggest that there are underlying factors or patterns causing these shifts in sentiment over the series. This could reflect the nature of the data source, such as a text or series of texts where the sentiment fluctuates with context or events rather than showing a steady trend in one direction.

In sum, the binned and quadratic analysis provides a nuanced view of sentiment progression, emphasizing the non-linear and cyclical nature of sentiment changes within the dataset (Figure 5).

3.1.2 The Descriptive Interpretation of the Sentiment Values at Sentence Level

The descriptive statistics indicate that, at the sentence level, the sentiment is generally negative, with a mean and median both in the negative range. However, there is moderate variability in sentiment across sentences, with a wide range of values and a relatively flat distribution that is not heavily skewed in any direction.

Table 7. Parameter estimates (q=0.5)[a,b]

						95% Confidence Interval	
Parameter	**Coefficient**	**Std Error**	**t**	**Df**	**Sig.**	**Lower Bound**	**Upper Bound**
(Intercept)	-0.579	0.0181	-11.245	1265	0	-0.036	0.036
Word	0.514	0.0407	12.65	0	-0.679	-0.478	-0.478
Dword	-1.603	0.0407	-39.399	1265	0	-1.683	-1.523

This implies a diverse sentiment across the sentences, with a slight tendency toward more negative expressions (Table 3). The graph depicts the cumulative sum of sentiment scores at the sentence level (Figure 4). A predominantly downward trend, as shown by the blue line, indicates that the cumulative sentiment becomes more negative over the sequence.

The strong negative slope and the high R2 value suggest that there is a consistent and significant negative trend. The sentiment of sentences over this data sequence. This could imply that as the sequence progresses, the sentences express increasingly negative sentiments.

If this is a time series, for example, it could suggest a worsening of sentiment over time. If it's a sequence of sentences from a text or series of texts, it could indicate a narrative arc that becomes more negative (Figure 6). The values of the water flow chart (cumulative sum) chart of the sentiments of the sentences show that at the beginning of the graph, there is a decline in the sentiment values (Figure 5).

The graph reflects the cumulative sentiment of sentences when binned, showing a complex sentiment pattern with fluctuations, including a significant downturn and subsequent recovery.

The lower 2R2 value indicates that while the quadratic trend line captures the overall shape of the data, there is still a considerable amount of variation that it does not explain.

This suggests that the sentiments of the sentences exhibit non-linear behavior with significant variance, which may be influenced by various factors not captured by a simple quadratic trend. The binned approach smooths out some of the variations and helps identify broader patterns in the sentiment data (Figure 7).

Table 8. Covariances of parameter Estimates (q=0.5)[a,b]

	(Intercept)	Word	dword
(Intercept)	0.00033	0.00023	0.00011
Word	0.00023	0.00265	0.00132
Dword	0.00011	0.00132	0.00166

a: Dependet Variable: d2word.

b: Model (Intercept), word, dword.

Table 9. Correlations of parameter Estimates (q=0.5)[a,b]

	(Intercept)	Word	Dword
(Intercept)	1	0.247	0.156
Word	0.247	1	0.633
Dword	0.156	0.633	1

a: Dependet Variable: d2word.

b: Model (Intercept), word, dword.

Table 10. Observed Model Quality (q=0.5)[a,b,c]

Pseudo R Squared	0.478
Mean Absolute Error (MAE)	0.3297

a: Dependet Variable: d2sdt2.

b: Model (Intercept), s,dsdt.

c: Method: simplex algorithm.

3.1.3 The Descriptive Interpretation of the Sentiment Values at Paragraph Level

The descriptive statistics for sentiment analysis at the paragraph level indicate a negative bias in sentiment with a mean of -0.2330 and a standard error of 0.03214. The confidence interval suggests this mean is statistically significant and is not due to random chance. The median of -0.2000 is in line with the mean, further indicating negative sentiment.

The variance and standard deviation are relatively low, suggesting sentiments across different paragraphs are not widely dispersed but are fairly consistent around the mean. The minimum and maximum values show that sentiments range from moderately negative to slightly positive. Overall, these statistics suggest that paragraphs tend to express negative sentiments more frequently than positive ones, with a relatively consistent sentiment distribution that is moderately concentrated around the mean and median values (Table 4). The graph depicts the cumulative sum of sentiment scores at the paragraph level (Figure 6). A predominantly downward trend, as shown by the blue line, indicates that the cumulative sentiment becomes more negative over the sequence.

The strong negative slope and the high R2 value suggest that there is a consistent and significant negative trend in the sentiment of sentences over this data sequence. This could imply that as the sequence progresses, the paragraphs express increasingly negative sentiments. If this is a time series, for example, it could suggest a worsening of sentiment over time.

If it's a sequence of sentences from a text or series of texts, it could indicate a narrative arc that becomes more negative (Figure 8). The cumulative sentiment values of the paragraphs show a non-linear pattern, initially declining and then rising, which suggests variability in sentiment throughout the paragraphs.

The substantial R2 value indicates a good fit for the quadratic model but also implies that there are other factors affecting sentiment that are not explained by this model alone. This could mean that the paragraphs may follow a narrative arc, with shifts in sentiment that could correspond to different stages or events in the text (Figure 9).

3.2 Modelling Differential Equations for Raw Values of the Sentiments Via Quantile Regression

3.2.1 The Differential Equations Modelling for the Words as the Main Unit of the Research

The table presents the outcomes of a quantile regression analysis aimed at understanding the factors that influence the median value of the dependent variable 'd2word', using 'word' and 'dword' as predictors.

The Mean Absolute Error (MAE) is a measure of the average magnitude of the errors in a set of predictions without considering their direction. An MAE of .3562 indicates that, on average, the predictions of the median value of the dependent variable deviate from the observed median values by .3562 units.

The model explains a significant portion of the variability at the median level and provides insights with a relatively low average prediction error. In the realm of statistical analysis, the evaluation and comparison of models using metrics such as the Mean Absolute Error (MAE) and Pseudo R Squared is pivotal for understanding model performance.

A lower MAE indicates better predictive accuracy. In the model's MAE is significantly lower than that of the null model, suggesting that including the predictors ('word' and 'dword') improves the model's ability to accurately predict the median of 'd2word'.

The Pseudo R Squared value for the model indicates that about 51% of the variability in the median of 'd2word' is accounted for by the model. In contrast, the null model, with a Pseudo R Squared of 0.000, explains none of the variability. This further suggests that your model provides a substantial improvement over the null model. Based on the information from Table 7.

In this table which provides parameter estimates for a statistical model, we can construct the equation for the dependent variable d2word. The table lists the coefficients for an intercept, Word, and Dword, along with other statistical details:

$$\frac{d^2}{dx^2}\,\mathrm{Word} = -0.579 \times \mathrm{Word} - 1.603 \times \frac{d}{dx}\,\mathrm{Word}\,. \quad (1)$$

Multicollinearity refers to a situation where predictor variables in a regression model are highly correlated. The covariance values between the different parameters (intercept, word, dword) are relatively small. This suggests that the predictors are not highly correlated with each other. Based on the correlation coefficients, there is a potential issue of multicollinearity in your model, particularly between the variables 'word' and 'dword'. While this level of correlation is a concern, it does not automatically invalidate the model.

Table 11. Null Model Quality (q=0.5)[a,b,c]

Pseudo R Squared	.000
Mean Absolute Error (MAE)	.6312

a: Dependet Variable: d2sdt2.

b: Model (Intercept).

c: Method: simplex algorithm.

3.2.2 The Differential Equations Modelling for the Sentences as the Main Unit of The Research

The regression analysis is focused on the median (0.5th quantile) of the dependent variable.

Quantile regression at the median is particularly useful for understanding the central tendency of the dependent variable, especially in cases where the data might be skewed or have outliers. Pseudo R Squared, 0.478 value suggests that approximately 47.8% of the variability in the median of the dependent variable ('d2sdt2') is explained by the model.

In quantile regression, the Pseudo R-squared provides a measure of the model's explanatory power, though it does not have a direct analog to the R-squared in OLS regression. A value of 0.478 indicates a moderate level of explanatory power.

Mean Absolute Error (MAE), 0.3297: The MAE value of 0.3297 means that the average magnitude of the errors in the model's predictions is 0.3297 units. This metric helps to understand the average error in predictions without considering the direction of the errors. A lower MAE is generally preferable, indicating more accurate predictions.

The comparison clearly shows that the full model with the predictors 's' and 'dsdt' performs substantially better than the null model. This is evident both in terms of the model's explanatory power (Pseudo R Squared) and its predictive accuracy (MAE):

– **Pseudo R Squared:** The increase from 0.000 in the null model to 0.478 in the full model indicates a substantial improvement in the explanatory power of the model. A Pseudo R Squared of 0.478 suggests that approximately 47.8% of the variability in the median of 'd2sdt2' is explained by the full model, whereas the null model explains none.

Table 12. Parameter estimates (q=0.5)[a,b]

						95% Confidence Interval	
Parameter	**Coefficient**	**Std. Error**	**T**	**Df**	**Sig.**	**Lower Bound**	**Upper Bound**
(Intercept)	-,087	,0264	-3,306	674	,001	-,139	-,035
S	-,590	,0718	-8,216	674	,000	-,730	-,449
Dsdt	-1,627	0629	-25,848	674	000	-1,750	-1,503

Table 13. Covariances of parameter Estimates (q=0.5)[a,b]

	(Intercept)	**S**	**dsdt**
(Intercept)	0.00070	0.00072	0.00036
S	0.00072	0.00515	0.00258
Dsdt	0.00036	0.00258	0.00396

a: Dependet Variable.

b: Model (Intercept), s,dsdt.

- **Mean Absolute Error (MAE):** The decrease in MAE from 0.6312 to 0.3297 is significant. This indicates that the full model, with its predictors, is much more accurate in predicting the median of 'd2sdt2' compared to the null model, which merely uses the median of the dependent variable for prediction.

Based on the information provided in Table 12, which includes parameter estimates for a statistical model, we can write down the equation for the dependent variable d2sdt2. This table presents the coefficients for an intercept, S and Dsdt, along with other statistical details:

$$\frac{d^2\,\mathrm{Se}}{dx^2} = -0.087 - 0.590 \times \mathrm{Se} - 1.627 \times \frac{\mathrm{dSe}}{dx}, \quad (2)$$

where Se = Sentence.

The values in the table represent the covariances between the estimates of the model parameters. Covariance is a measure of how much two random variables vary together.

In the context of regression coefficients, it provides insight into the relationship between the precision of the estimates of different parameters. (Intercept) Row and Column: The covariance of the intercept with itself (0.00070) is its variance. The covariances between the intercept and each of the predictors ('s' and 'dsdt') are 0.00072 and 0.00036, respectively.

These values indicate how the estimate of the intercept co-varies with the estimates of the other parameters. Row and Column: The variance of the 's' coefficient is 0.00515. Its covariance with 'dsdt' is 0.00258. These values tell us how the estimate of 's' changes in relation to both the intercept and 'dsdt'.

dsdt Row and Column: The variance of the 'dsdt' coefficient is 0.00396, and its covariance with the other parameters is indicated in the respective cells. The correlation of 0.571 between 's' and 'dsdt' suggests there might be some level of multicollinearity.

However, this level of correlation is not extremely high, so it may not be severe enough to significantly distort your regression coefficients or their standard errors. It's important to note that while moderate correlations can indicate potential multicollinearity, they don't always warrant significant concern unless they're very high (closer to 1 or -1) (Table 14).

3.2.3 The Differential Equations Modelling for the Paragraphs as the Main Unit of the Research

In the context of quantile regression, the provided data indicates a model assessing the median (50th percentile) of the dependent variable 'dp2dt2', utilizing two predictors, 'p' and 'dpdt'. The model's fit is moderately good, as indicated by a Pseudo R Squared value of 0.464, meaning approximately 46.4% of the variation in the dependent variable is explained by the model.

The Mean Absolute Error (MAE) of 0.1498 suggests the predictions are reasonably accurate. The model employs the Simplex algorithm, a method commonly used for solving linear programming problems in optimization scenarios.

This approach provides a more nuanced understanding of the data compared to traditional regression methods, especially in terms of distribution tails (Table 15). Comparing the two models in the context of quantile regression, both aimed at predicting the median of 'dp2dt2', reveals significant differences in their performance.

Table 14. Covariances of parameter Estimates (q=0.5)[a,b]

	(Intercept)	S	dsdt
(Intercept)	1	0.378	0.215
S	0.378	1	0.571
Dsdt	0.215	0.571	0.1

a: Dependet Variable: d2sdt2.

b: Model (Intercept), s,dsdt.

Table 15. Observed model quality (q=0.5)[a,b]

Pseudo R Squared	0.464
Mean Absolute Error (MAE)	0.1498

a: Dependet Variable: d2sdt2.

b: Model (Intercept), p, dpdt.

c: Method: simplex algorithm.

The null model (Table 16), which only includes an intercept, shows no explanatory power (Pseudo R Squared of 0.000) and a higher Mean Absolute Error (MAE) of 0.2796, indicating less accurate predictions. In contrast, the last model, which includes two predictors, 'p' and 'dpdt', along with an intercept, shows considerable improvement.

Its Pseudo R Squared value of 0.464 indicates it explains about 46.4% of the variation in the dependent variable, and its lower MAE of 0.1498 suggests more accurate predictions. Based on the provided table of parameter estimates for a statistical model, we can write down the equation for the dependent variable dp2dt2. The table lists the coefficients for an intercept, P, and Dpdt.

Along with their standard errors, t-values, degrees of freedom (df), significance levels (Sig.), and confidence intervals (Table 17). This table presents the coefficients for an intercept, p, and Dpdt, along with other statistical details:

$$\frac{d^2\,\mathrm{Par}}{dx^2} = -0.303 \times Par - 1.485 \times \frac{\mathrm{dPar}}{dx}, \qquad (3)$$

where par = Paragraph.

The covariance matrix for the quantile regression model at the 0.5 quantiles, predicting 'dp2dt2' with predictors 'p' and 'dpdt', provides insights into the relationships and variability of the parameter estimates. The diagonal elements show the variances of each parameter's estimate, with values of 0.00198 for the Intercept, 0.02043 for 'p', and 0.02290 for 'dpdt', indicating the spread of each estimate.

The off-diagonal elements represent covariances between pairs of parameters, such as 0.00478 between the Intercept and 'p', and 0.01100 between 'p' and 'dpdt'. These covariances reveal how changes in one parameter estimate are associated with changes in another, with positive values indicating a tendency for the estimates to increase together.

This matrix is crucial for understanding the precision of estimates and identifying potential multicollinearity in the model. While correlation does not imply causation, high correlation coefficients (like 0.751 between the Intercept and 'p') might hint at potential collinearity issues. Collinearity can make it difficult to discern the individual impact of predictors on the dependent variable, potentially leading to unreliable coefficient estimates.

The presence of significant correlations between parameters necessitates careful interpretation of the model coefficients (Table 19). There is an inherent relationship between a variable and its derivatives. The first derivative represents the rate of change of the variable, and the second derivative represents the rate of change of the first derivative. This natural linkage can lead to a high correlation among these predictors.

In quantile regression, like in other regression types, multicollinearity can affect the precision of the coefficient estimates. If the model's primary goal is prediction and it shows good predictive performance (i.e., it accurately predicts the dependent variable 'dp2dt2'), then it may still be considered valid for that purpose, even with multicollinearity. We don't present the solutions of the differential equations there since the primary interest lies in understanding the relationships and dynamics represented by the differential equation rather than in the specific solutions. The equation itself can reveal how different factors are related and how they influence the rate of change of a variable.

This is particularly relevant in sentiment analysis, where the rate of change of sentiment is more informative than the absolute sentiment

Table 16. Null model quality (q=0.5)[a,b]

Pseudo R Squared	0.000
Mean Absolute Error (MAE)	0.2796

a: Dependet Variable: d2sdt2.

b: Model (Intercept).

c: Method: simplex algorithm.

value at a specific point. Moreover, differential equations provide a generalized model of a system's behavior. The solutions, however, are often specific to initial conditions or particular parameters.

By presenting the equations, researchers can convey the general dynamics that apply across various scenarios rather than being tied to specific instances.

4 Discussion

The 1864 novella ”Notes from Underground” by Fyodor Dostoevsky introduces the Underground Man, a cynical recluse living in St. Petersburg. In the philosophical first half, he contends that human nature is irrational, making ideal societies impossible.

Overall, the sentiments in the ”Underground” section are dark, complex, and fraught with tension. They reflect a deep sense of disillusionment with both society and the self, as well as a profound existential despair. The second half follows a more conventional format.

The opening ”Underground” section establishes a gloomy, contemplative mood through the protagonist's cynical monologues on society, reason, and the meaning of life. He grapples with complex ideas that lead to dark, nihilistic conclusions about human nature and the pursuit of happiness.

The tone reflects his mental agony and sense of estrangement. Both the beginning and the end of the ”Underground” section are negative, but the nature of this negativity shifts. The beginning is more confrontational and critical, actively challenging societal norms and intellectual trends.

The end, in contrast, is more resigned and reflective, focusing on the inescapable suffering and irrationality of the human condition. We showed that when we bin the cumulative sum of sentiments, we might uncover more complex structures and dynamics in the data that are not evident when examining the raw, ungrouped cumulative totals.

This can be particularly useful for detecting chaotic patterns and understanding the true sentiment dynamics within a dataset. Binned values can reveal a more dynamic and potentially chaotic structure due to several reasons:

1. **Smoothing Effect:** Binning can smooth out short-term fluctuations in the data, making it easier to observe longer-term trends and patterns. This smoothing can sometimes reveal underlying structures that are obscured by noise in the raw data.

2. **Highlighting Extremes:** By grouping data into bins, extreme values can have a more pronounced effect on the visual representation of the data. This can make the highs and lows of sentiment more evident, showing a more volatile or chaotic structure.

3. **Revealing Non-Linearity:** When sentiment values are binned, non-linear trends may become more apparent. The raw cumulative sum might show a general trend up or down, but binned values could show cycles or patterns of sentiment that change direction or have variable intensity.

4. **Aggregating Variability:** Binning combines the variability of individual values within each bin, which can highlight the range of sentiments within sections of the data. This variability can indicate a more chaotic sentiment structure, with rapid shifts from positive to negative or vice versa.

5. **Focus on Distribution:** The binned cumulative sum shifts the focus from individual data points to the distribution of data within each bin. This can reveal a more complex sentiment structure that includes the frequency and intensity of sentiment scores.

Table 17. Parameter estimates (q=0.5)[a,b]

						95% Confidence Interval	
Parameter	**Coefficient**	**Std Error**	**t**	**Df**	**Sig.**	**Lower Bound**	**Upper Bound**
(Intercept)	0.000	0.0445	0.000	51	1.000	-0.089	0.089
P	-0.303	0.1429	2.120	51	0.039	-0.590	-0.016
Dpdt	-1.485	0.1513	-9.812	51	0.000	-1.789	-1.181

Table 18. Covariances of parameter estimates (q=0.5)[a,b]

	(Intercept)	P	dpdt
(Intercept)	0.00198	0.00478	0.00234
P	0.00478	0.02043	0.01100
Dpdt	0.00234	0.01100	0.02290

a: Dependet Variable: d2sdt2.

b: Model (Intercept), p, dpdt

Aggregating sentiment into cumulative sums makes sense to see overall trends and patterns over time rather than just individual data points. Analyzing binned cumulative sentiment can reveal hidden patterns, trends, and dynamics compared to looking at raw sentiment data.

The hypothesis that binning will uncover more complex dynamics and chaos that are hidden in the raw data is reasonable, as binning can help detect signals and patterns from noise. It summarizes the sentiment while still highlighting the complex, chaotic nature of how sentiment evolves. Overall, this technique of binning cumulative sentiment time series appears to uncover more structure and chaos in the data than may be apparent from only considering individual sentiment values.

The analysis provides insight into the dynamic nature of cumulative sentiment. In the second part of the study, we get three equations that is a linear second-order differential equations.

They describe how the respective functions (Word, Sentence, and Paragraph) change with respect to some variable x. The coefficients (-0.579, -1.603, etc.) modify the effect of the function and its derivatives in the equation. The presence of the first and second derivatives indicates the rate of change and the acceleration of change, respectively, for each level (word, sentence, and paragraph) with respect to x.

Table 19. Correlation of parameter estimates (q=0.5)[a,b]

	(Intercept)	P	dpdt
(Intercept)	1	0.751	0.347
P	0.751	1	0.509
Dpdt	0.347	0.509	1

a: Dependet Variable: d2sdt2.

b: Model (Intercept), p, dpdt.

The equation for Word sentiments suggests that the acceleration of change in Word (represented by the second derivative) is influenced by both the current state of Word and its rate of change (first derivative).

The coefficient -0.579 affects the direct influence of Word, while -1.603 modifies the influence of its rate of change. The equation explains about 51% of the variation, indicating that approximately half of the changes in the Word data can be predicted or accounted for by this model:

$$\frac{d^2}{dx^2}\,\mathrm{Word} = -0.579 \times \mathrm{Word} - 1.603 \times \frac{\mathrm{d}}{dx}\,\mathrm{Word}\,. \quad (4)$$

In the equation of sentence sentiments, the change in Sentence is not only dependent on the Sentence itself and its rate of change but also includes a constant term (-0.087).

This constant could represent a baseline change independent of the current state or rate of change of the Sentence. This equation explains about 47.8% of the variation, meaning nearly half of the variability in the Sentence data can be explained by the model:

$$\frac{d^2}{dx^2}Se = -0.087 - 0.590 \times Se - 1.627 \times \frac{d}{dx}Se. \quad (5)$$

Similar to the Word equation, this one relates the acceleration of change in a Paragraph to its current state and rate of change, but with different coefficients. The fact that it explains about 46.4% of the variation indicates that less than half of the changes in the Paragraph data are accounted for by this model:

$$\frac{d^2}{dx^2}Par = -0.303 \times Par - 1.485 \times \frac{d}{dx}Par. \quad (6)$$

The percentages of variation explained (51%, 47.8%, and 46.4%) refer to how much of the change in each respective level (word, sentence, paragraph) can be predicted or explained by these models.

A higher percentage indicates a better fit of the model to the data, meaning the model is more effective at explaining the changes or variations in that particular level.

These percentages also imply that there are other factors or variables not captured by these models that contribute to the changes in Words, Sentences, and Paragraphs.

These could be external or more complex internal factors not accounted for in the linear model. Both the potential benefits and limitations of this approach for modeling differential equations can be given below:

4.1 Potential Benefits

1. **Understanding Dynamic Changes:** Differential equations can model the rate and acceleration of sentiment changes over time or across different text segments. This could be particularly insightful in understanding how sentiments evolve in complex narratives or dialogues.
2. **Predictive Analysis:** By modeling how sentiments change, researchers can potentially predict future sentiment trends based on current and past data. This could be valuable in applications like market analysis, social media monitoring, and interactive storytelling.
3. **Refining Chatbot Responses:** For AI development, understanding the dynamics of sentiment can help in refining chatbot interactions, making them more sensitive and responsive to the emotional content of user inputs.
4. **Identifying Underlying Patterns:** Differential equations might reveal underlying patterns in sentiment data that are not obvious from a simple analysis. This could lead to new insights into how sentiments are expressed and perceived in language.

4.2 Limitations and Challenges

1. **The complexity of Human Sentiments:** Human emotions and sentiments are complex and often non-linear, making them difficult to accurately model with differential equations. Emotions can be influenced by a myriad of factors that are challenging to quantify.
2. **Data Quality and Variability:** The accuracy of sentiment ratings from chatbots can vary, and the data might be noisy. This variability can make it difficult to derive meaningful differential equations that accurately represent sentiment dynamics.
3. **Over-Simplification:** Reducing the rich and nuanced field of human emotions to a set of differential equations might oversimplify reality. Emotions are not just quantitative variables that can be easily modeled; they are deeply qualitative and context-dependent.
4. **Interdisciplinary Challenges:** Effectively modeling sentiments with differential equations requires an interdisciplinary approach. This combining linguistics, psychology, mathematics, and computer science. This complexity can be a barrier to research. While finding differential equations from chatbot sentiment ratings is useful for analyzing and predicting sentiment trends, it also comes with significant challenges and limitations. It's an approach that may yield valuable insights into certain applications, particularly in enhancing AI and natural language processing capabilities.

   However, researchers should be cautious of oversimplifying the complexity of human emotions and be mindful of the limitations of the data and the models used.

5 Conclusion

This research aims to understand the dynamics of sentiment evolution in textual units ranging from individual words to expansive paragraphs. The study's innovative approach to analyzing sentiment in text, especially in the context of complex literary works like Fyodor Dostoevsky's ”Notes from

Underground,” reveals significant insights into the capabilities of advanced chatbots like GPT-4.

By employing a method that bins the cumulative sum of sentiments, we uncover deeper, more intricate structures and dynamics in sentiment data, transcending the limitations of traditional raw cumulative analyses. This method is particularly valuable in understanding the nuanced, often chaotic sentiment landscapes in literature, where emotions and themes are richly layered and dynamically evolving.

The differential equations derived for word, sentence, and paragraph levels further enrich our understanding. They demonstrate how the rate and acceleration of sentiment change are influenced by their current state and rate of change.

With varying percentages of variation explained at each text level (51% for Word, 47.8% for Sentence, and 46.4% for Paragraph), these models effectively illustrate the complex, dynamic nature of sentiment evolution in literary texts.

Moreover, the fact that these models do not account for all variability suggests the presence of other factors influencing sentiment changes, possibly external influences or more intricate internal dynamics. This underscores the multifaceted nature of sentiment analysis, especially in complex narrative contexts. The analysis of modeling sentiment dynamics through differential equations reveals both potential benefits and limitations. On the one hand, differential equations can provide insights into predicting sentiment trends, understanding complex narrative arcs, and refining chatbot interactions.

The approach may uncover hidden patterns and lead to new discoveries about how sentiments are expressed in language. However, accurately quantifying and modeling human emotions through mathematical equations is extremely challenging.

Sentiments are qualitative, subjective, and dependent on nuanced contextual factors that cannot be easily captured in simplistic models. While differential equation modeling of chatbot sentiment ratings offers some utility, care must be taken not to oversimplify the richness of human emotions. Further interdisciplinary research is needed to develop more sophisticated techniques that address the complexity of sentiments and their dynamics in language.

In conclusion, this approach has merit but requires caution against oversimplification of emotions.In conclusion, this study’s findings are pivotal for enhancing the capabilities of AI-driven chatbots in sentiment analysis, particularly in dissecting and understanding the layered emotional landscapes of literary works. It demonstrates the potential of advanced analytical techniques in extracting deeper meaning from texts, a crucial step forward in the field of natural language processing and AI-driven literary analysis.

Acknowledgments

This work was supported by the Ministry of Education and Sciences of the Republic of Kazakhstan under the grant #AP14871214 “Development of machine learning methods to increase the coherence of text in summaries produced by the Extractive Summarization Methods.” The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

1. **Akhmetov, I., Gelbukh, A., Mussabayev, R. (2022).** Topic-aware sentiment analysis of news articles. Computación y Sistemas, Vol. 26, No. 1, pp. 423–439. DOI: 10.13053/cys-26-1-4179.

2. **Alexandridis, G., Michalakis, K., Aliprantis, J., Polydoras, P., Tsantilas, P., Caridakis, G. (2020).** A deep learning approach to aspect-based sentiment prediction. Artificial Intelligence Applications and Innovations, pp. 397–408. DOI: 10.1007/978-3-030-49161-1_33.

3. **Alpar, R. (2012).** Uygulamalı İstatistik ve Geçerlik Güvenirlik. Detay yayıncılık.

4. **Belsley, D. A., Kuh, E., Welsch, R. E. (1980).** Regression diagnostics: Identifying Influential data and sources of collinearity. John Wiley & Sons. DOI: 10.1002/0471725153.

5. **Birjali, M., Kasri, M., Beni-Hssane, A. (2021).** A comprehensive survey on sentiment analysis: approaches, challenges and trends. Knowl-Based Systems, Vol. 226, pp. 2–26. DOI: 10.1016/j.knosys.2021.107134.

6. **Casillo, M., Clarizia, F., D'Aniello, G., De-Santo, M., Lombardi, M., Santaniello, D. (2020).** Chat-bot: A cultural heritage aware teller-bot for supporting touristic experiences. Pattern Recognition Letters, Vol. 131, pp. 234–243. DOI: 10.1016/j.patrec.2020.01.003.

7. **Chang, M., D'Aniello, G., Gaeta, M., Orciuoli, F., Sampson, D., Simonelli, C. (2020).** Building ontology-driven tutoring models for intelligent tutoring systems using data mining. IEEE Access, Vol. 8, pp. 48151–48162. DOI: 10.1109/ACCESS.2020.2979281.

8. **Chen, Z., Liu, Y., Sun, H. (2021).** Physics-informed learning of governing equations from scarce data. Nat Commun, Vol. 12, No. 6136. DOI: 10.48550/arXiv.2005.03448.

9. **Colace, F., de-Santo, M., Greco, L. (2014).** Safe: a sentiment analysis framework for e-learning. International Journal of Emerging Technologies in Learning, Vol. 9, No. 6, pp. 37–41. DOI: 10.3991/ijet.v9i6.4110.

10. **Collomb, A., Costea, C., Joyeux, D., Hasan, O., Brunie, L. (2014).** A study and comparison of sentiment analysis methods for reputation evaluation. Rapport de Recherche RR-LIRIS-2014-002.

11. **Dietz-Uhler, B., Hurn, E. J. (2013).** Using learning analytics to predict (and improve) student success: A faculty perspective. Journal of Interactive Online Learning, Vol. 12, pp. 17–26.

12. **Do, H. H., Prasad, P., Maag, A., Alsadoon, A. (2019).** Deep learning for aspect-based sentiment analysis: A comparative review. Expert Systems with Applications, Vol. 118, pp. 272–299. DOI: 10.1016/j.eswa.2018.10.003.

13. **Duran, V. (2022).** Atatürk'ün "zabit ve kumandan ile hasb-i hâl" adlı eserinin eğitsel kavramlar açısından İncelenmesi duygu analizinin diferansiyel denklemler aracılığıyla modellenmesi. Doğumunun 141. Yılında Atatürk 2. Uluslararası Sempozyumu, pp. 48–72.

14. **Duran, V. (2023).** Modeling sentiment dynamics in terminator 3 subtitles using gpt-4 and differential equations based on fuzzy logic. 7th International Innovative Studies & Contemporary Scientific Research Congress.

15. **D'Aniello, G., Gaeta, M., La Rocca, I. (2022).** KnowMIS-ABSA: an overview and a reference model for applications of sentiment analysis and aspect-based sentiment analysis. Artificial Intelligence Review, Vol. 55, No. 3, pp. 5543–5574. DOI: 10.1007/s10462-021-10134-9.

16. **Fan, D. P., Cook, R. D. (2003).** A differential equation model for predicting public opinions and behaviors from persuasive information: Application to the index of consumer sentiment. The Journal of Mathematical Sociology, Vol. 27, No. 1, pp. 29–51. DOI: 10.1080/00222500305886.

17. **Hemmatian, F., Sohrabi, M. K. (2019).** A survey on classification techniques for opinion mining and sentiment analysis. Artificial Intelligence Review, Vol. 52, No. 3, pp. 1495–1545. DOI: 10.1007/s10462-017-9599-6.

18. **Kelsingazin, Y., Akhmetov, I., Pak, A. (2021).** Sentiment analysis of kaspi product reviews. 16th International Conference on Electronics Computer and Computation (ICECCO) Kaskelen, Kazakhstan, pp. 1–5. DOI: 10.1109/ICECCO53203.2021.9663854.

19. **Leippold, M. (2023).** Sentiment spin: Attacking financial sentiment with GPT-3.

Finance Research Letters, Vol. 55, pp. 1–6. DOI: 10.1016/j.frl.2023.103957.

20. **Liu, B. (2012).** Sentiment analysis and opinion mining. Morgan Claypool Publishers, pp. 1–168.

21. **Meškelė, D., Frasincar, F. (2020).** ALDONAR: a hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalized domain ontology and a regularized neural attention model. Information Processing & Management, Vol. 57, No. 3, pp. 1–9. DOI: 10.1016/j.ipm.2020.102211.

22. **Misuraca, M., Forciniti, A., Scepi, G., Spano, M. (2020).** Sentiment analysis for education with R: Potential benefits, methods and practical applications. DOI: 0.48550/arXiv. 2005.12840.

23. **Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De-Clercq, O., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S. M., Eryiğit, G. (2016).** SemEval-2016 task 5: aspect based sentiment analysis. Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), Association for Computational Linguistics, San Diego, pp. 19–30.

24. **Schmidt, M. D., Lipson, H. (2009).** Distilling free-form natural laws from experimental data. Science, Vol. 324, pp. 81–85. DOI: 10.1126/ science.1165893.

25. **Schouten, K., Frasincar, F. (2018).** Ontology-driven sentiment analysis of product and service aspects. The Semantic Web, pp. 608–623. DOI: 10.1007978-3-319-93417-4_39.

26. **Wankhade, M., Rao, A. C. S., Kulkarni, C. (2022).** A survey on sentiment analysis methods, applications, and challenges. Artif Intell Rev, Vol. 55, pp. 5731–5780. DOI: 10. 1007/s10462-022-10144-1.

27. **Xing, F. Z., Cambria, E., Welsch, R. E. (2018).** Natural language based financial forecasting: a survey. Artificial Intelligence Review, Vol. 50, No. 1, pp. 49–73. DOI: 10. 1007/s10462-017-9588-9.

28. **Yadav, A., Vishwakarma, D. K. (2020).** Sentiment analysis using deep learning architectures: a review. Artificial Intelligence Review, Vol. 53, No. 6, pp. 4335–4385. DOI: 10.1007/s10462-019-09794-5.

29. **Zhang, L., Wang, S., Liu, B. (2018).** Deep learning for sentiment analysis: a survey. Wiley Interdisciplinary Review, Vol. 8, No. 4, pp. 1253. DOI: 10.48550/arXiv.1801.07883.

Article received on 21/11/2023; accepted on 22/01/2024.
** Corresponding author is Alexander Gelbukh.*

ISSN 2007-9737

MiniCovid-Unet: CT-Scan Lung Images Segmentation for COVID-19 Identification

Alvaro Salazar-Urbina[1], Elías Ventura-Molina[2], Cornelio Yáñez-Márquez[*,1],
Mario Aldape-Pérez[2], Itzamá López-Yáñez[2]

[1] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

[2] Instituto Politécnico Nacional,
Centro de Innovación y Desarrollo,
Tecnológico en Cómputo,
Mexico

{asalazaru2020, cyanez}@cic.ipn.mx,
{eventuram, maldape, ilopezy}@ipn.mx

Abstract. Detection and segmentation of Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-COV2 or COVID-19) is a difficult task due the different kinds of shapes, sizes and positions of the injury. Medical institutions have vast challenges because there is an urgent need for efficient tools to improve the diagnosis of COVID-19 patients. Computer tomography images (CT) are necessary for medical specialists to diagnose the patient's condition. Nevertheless, there is a lack of both in Medical Centers, mainly in rural areas. The manual analysis of CT images is time-consuming; in addition, most images have low contrast, and it is possible to find blood vessels in the background, so the difficulty of a suitable diagnosis increases. Nowadays, deep learning methods are an alternative method to perform the detection and segmentation task. In this work, we propose a novel light model to detect and identify COVID-19 using CT images: MiniCovid-Unet. It is an improved version of U-net; main differences reside on the decoder and encoder architecture, MiniCovid-Unet needs fewer convolution layers and filters because it focuses only on COVID-19 images. Also, as a result of employing fewer parameters, it can be trained in less time, and the resulting model is light enough to be downloaded to a mobile device. In this way, it is possible to have a quick and confident diagnosis in remote areas, where there exists an absence of internet connection and medical specialists.

Keywords. Deep learning, image segmentation, COVID-19, computer tomography, Mask R-CNN, Unet, MiniCovid-Unet.

1 Introduction

SARS-CoV2, better known as COVID-19 or Coronavirus, is an acute fatal disease identified in December 2019 in Wuhan province, China. This virus spread worldwide with great speed [1], declaring itself a pandemic on March 11, 2020 [2]. As of October 31, 2020, 45,428,731 cases have been confirmed in the world, causing 1,185,721 deaths [3].

COVID-19 is spread through droplets of secretion released from the mouth and nose of an infected individual [4] and is transmitted by direct or indirect contact (through contaminated objects and surfaces) to mucosal areas of the skin such as the mouth, nose, or tear ducts. Symptoms may include dry cough, fever, headache, fatigue, shortness of breath, loss of taste or smell, and shortness of breath. Symptoms usually appear 2 to 14 days after infection [5]. An early diagnosis is important because it is one of the most effective methods to stop the disease progression [6].

There are studies that have shown that COVID19 virus mainly attacks human lungs, after that there is a possibility of an infection and a lung disease [7]. Therefore, the diagnosis using a patient's chest computed tomography (CT) is so relevant.

doi: 10.13053/CyS-28-1-4697

The main aspect in a CT image of COVID is the presence of ground glass opacity (GGO) [8,9]. Some experts have identified three main types of anomalies in CT lung images related to COVID-19: ground glass opacification, consolidation and pleural effusion [10,11].

The manual observation is the main technique to decide whether the patients are infected or not. However, the job is exhausted and there aren’t enough medical doctor’s staff to do the job. So, an automatic segmentation system is necessary in order to identify and delimit the boundary of the region of interest in the lung [12].

Deep Learning (DL), a subfield of Machine Learning, is a tool commonly used in re-search areas for speech recognition, computer vision, natural language processing, and image processing [13]. The main advantage of DL methods is that they do not require experts to perform feature extraction; it is done automatically and implicitly by multiple flexible linear and non-linear processing units in a deep architecture.

In recent years, Deep Learning has been a useful tool for classifying medical images [14], among its techniques the convolutional neural network (CNN) model [15] stands out; a neural network inspired by the connectivity of the animal visual cortex. CNN is a multi-layer neural network that uses minimal processing of convolution operations on the pixels of the images. This technique extracts the relevant features from image sets to detect features regardless of their position.

Nowadays, the computer’s power has made it possible to apply deep learning in a wide range of applications in the medical field, such as deciding whether a tumor is in a radiograph [16] or detect a cardiovascular risk. For the task about semantic segmentation, there is a constant improvement in the accuracy of segmentation with models such as Fully Convolutional Network (FCN) [17], U-net [18], Fast RCNN [19] and Mask RCNN [18] among others.

There are a lot of models that detect Covid19 cases from chest x ray images [20–22], yielding a prediction value of 90% [23]. However, this kind of model cannot provide a quantitative analysis of infection severity because they just classify between Covid19 and regular pneumonia.

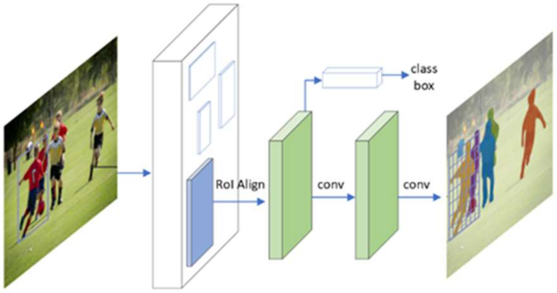


Fig. 1. Framework of the Mask R-CNN method used for detection and segmentation COVID-19 in CT images

2 Related Work

2.1 Mask R-CNN

Mask R-CNN Is a framework focused on instance segmentation. This task combines elements of object detection (classify individual objects and localize every instance with a bounding box) and semantic segmentation (classify every pixel in a set of categories).

The Figure 1 shows a representation of the Mask R-CNN framework that contains two main phases; the first one consists of a Faster R-CNN architecture [19]. It has three elements: the backbone, the region proposal network (RPN) and the object detection [18]. The backbone takes advantage of a CNN architecture for image feature extraction and generating feature maps.

The RPN uses these maps and creates proposed bounding boxes (anchors) to do the object detection task, dispersed over each feature map. These bounding boxes or anchors are classified in two classes: positive anchors or foreground, which refers to the anchors located in regions that represent features on the objects to be detected, and the negative ones or background which are located outside these objects.

The positive anchors are used to perform a task called region of interest (ROI) alignment; they are centered to the located object and mark the ROIs for the next part. The object detection is the last part and classifies every class inside each ROI. The second phase consists of a new branch in order to do the instance segmentation task over every detected object inside the image. This new branch is made by a fully convolutional mask [18].

2.2 Unet

Unet is one of most popular models for the task of image segmentation in the medical field. It was developed to understand in a visual way different types of images. And it is based on an encoder decoder neural network architecture. There are two main parts: con-tractive and expansive. The contracting one is built with several layers of convolution, filters of size 3 x 3 and strides in both directions, with ReLU layers at the end.

This part is important because it extracts the essential features of the input and the result is a feature vector of a particular dimension. The second part recover information from the contractive part by coping and cropping. However, the feature vector is built by convolutions and generate an output segmentation map. In this architecture the main component is the linking operation between the first and second part.

In this way the network gets correct information from the first part, so it could generate an accurate segmentation mask [18].

2.3 SegNet

SegNet is a deep fully convolutional neural network architecture for semantic seg-mentation [24]. It was originally designed for road and interior scene segmentation tasks. This requires the network to converge using an unbalanced dataset because the pixels of the road, sky, and buildings dominate. The main elements consist of an encoder network, a corresponding decoder followed by a pixel classification layer.

The encoder network is almost the same as the 13 convolutional layers of the VGG16 network [25]. The task of the decoder network is to map low resolution encoder feature maps to full input resolution feature maps for pixel classification. The main feature of SegNet is the way the decoder upsamples its lower resolution input feature maps; in this part, the decoder network uses clustering indices computed in the maximum clustering step of the corresponding encoder to perform non-linear upsampling.

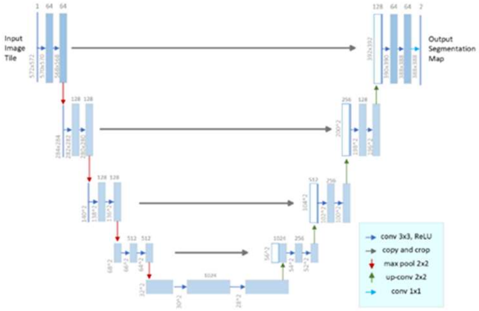


Fig. 2. U-net architecture [18]

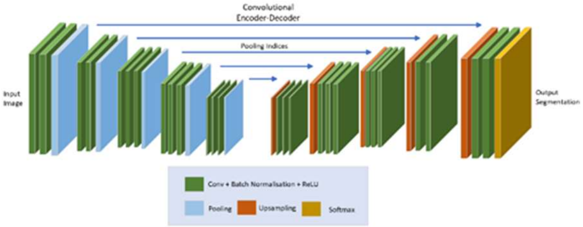


Fig. 3. SegNet architecture [24]

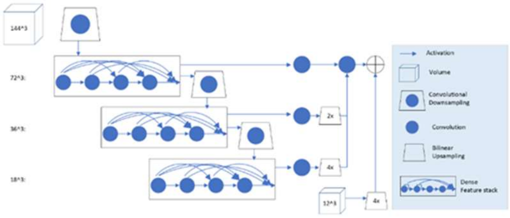


Fig. 4. Dense V-Net architecture [26]

2.4 Dense V-Net

Dense V-Net is a fully connected convolutional neural network that has performed well on the organ segmentation task. You can establish a voxel-voxel connection between the input and output images [26].

It consists of three layers of dense feature stacks whose outputs are concatenated after a convolution on the missing connection and bilinear oversampling [27]. There are 723 feature maps that are computed using a convolution step.

It then continues with a cascade of convolutions and dense feature stacks to generate activation maps with resolutions of three outputs. A convolution unit is applied on each output resolution to reduce the number of features. At the end it generates the segmentation logit.

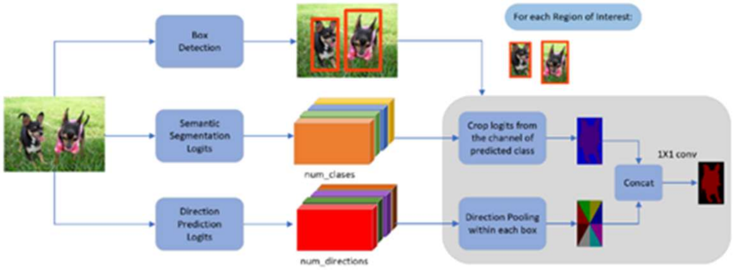

Box Detection
For each Region of Interest:
Semantic Segmentation Logits
num_clases
Crop logits from the channel of predicted class
1X1 conv
Concat
Direction Prediction Logits
num_directions
Direction Pooling within each box

Fig. 5. MaskLab architecture [29]

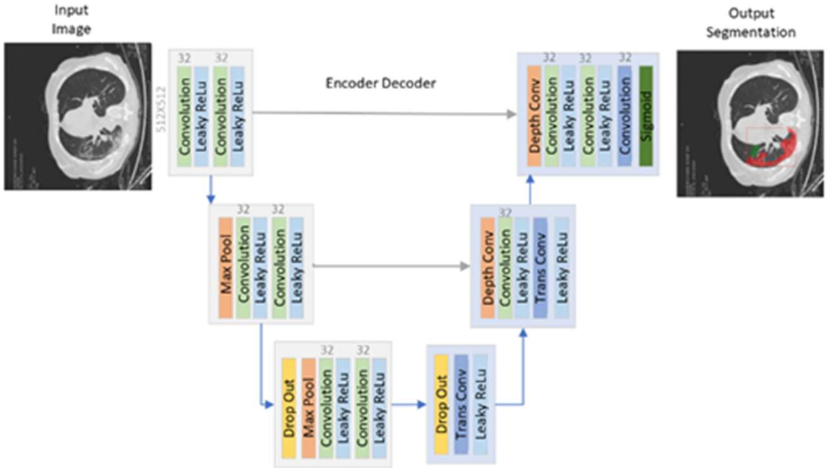


Fig. 6. MiniCovid-Unet architecture

Dense V-Net differs from V-Net [28] in several respects: the downsampling subnet-work is a sequence of three dense feature stacks connected by downsampling strided convolutions; each skip connection is a unique convolution to the output of the corresponding dense feature stack. The upsampling network comprises bilinear upsampling to the final segmentation resolution.

2.5 MaskLab

MaskLab is an instance segmentation model [29], refines object detection with ad-dress and semantic features based on Faster R-CNN [19]. This model produces three out-puts: box detection, semantic segmentation logits for pixel classification, and direction prediction logits to predict the direction of each pixel around its instance center.

Therefore, MaskLab is based on the Faster R-CNN object detector, the predicted frames provide precise location of object instances. Within each region of interest, MaskLab performs fore-ground and background segmentation by combining semantic and direction prediction.

2.6 MiniCovid-Unet

The ground glass opacities are important features of COVID-19 infection regions in CT images scans. However, these image characteristics cannot be extracted efficiently by conventional CNNs, where the original images are taken as input and the learning processes begin from pixel level features. Hence, to reflect more regional features of infections we use different filters to highlight the region of interest.

As shown in Figure 6, the proposed COVID-19 segmentation model applies the Unet like structure

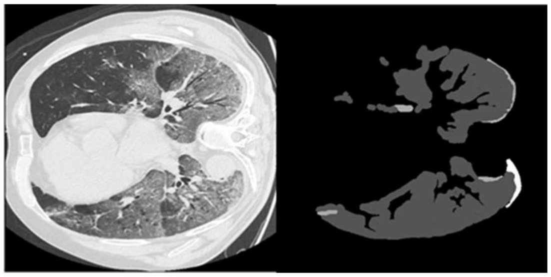


Fig. 7. Image and mask sample. CT scan (left) and labeled classes (right), where dark gray is ground glass opacities, gray is pleural effusion and white is consolidation

as backbone. There are two basic sections: contractive and expansive. We have used the activation function Leaky Rely in all blocks of layers because it is faster and it reduces the complexity of the network. Every convolution layer has 32 filters for images of 512 X 512 pixels.

There are less layers of convolution because the improvement was slightly better, however it increased the time of training and the computer resources needed. The model we proposed has good performance for computers with limited resources and is small enough to use in a mobile device.

3 Materials and Methods

Images of the dataset are Computed Tomography (CT) scans that belong to the Italian Society of Medical and Interventional Radiology [30]. The dataset contains one hundred one-slice CT scans in png format, whose dimensions are 512 x 512. There are also masks that show the region labeled by experts of the medical field [31].

In the original dataset there are three kinds of injuries related with Covid19: ground-glass opacities, consolidation and pleural effusion (Figure 7). However, we just try to identify whether an image has an injury in the lung and where it is located. The images are of people who had been infected with COVID-19.

The training of Mask R-CNN used a total of 72 of lung CT images and lung segmentation masks labels, the original image's size remained and no data enhancement was used for training. The validation set used 18 images and its masks. The training set iteration was 16 with 500 steps per iteration. The learning rate was 0.001. We set aside 10 im-ages to visualize the performance of the trained and validated model with the training and validation data sets.

For this experiment the backbone CNN architecture used was ResNet50 because of the small graphic card [32]. The experiment used pre-trained COCO weights [18,33]. The total number of parameters for Mask R-CNN is 44,662,942.

There is a problem with imbalance classes, because the task is to segment only the COVID-19 infected region. But with this configuration we have two classes: COVID-19 region and non-COVID-19. In this case, we have more pixels from healthy regions (2.4482e + 7) than from infected regions (2.119975e + 6). So, the unbalance ratio is 11 and the data set is unbalanced, that's the reason we have chosen metrics for the segmentation task.

3.1 Implementation Details

The Jupyter notebook interactive development environment was used to build and visualize the model and results. Python 3.6 was used as a programming language and the hardware configured to execute the experiments was a personal computer with a processor Intel(R) Core (TM) i7-6700 CPU @ 3.40GHz with 8 cores. NVIDIA GeForce GTX 1050 Ti (GPU 0), CUDA Toolkit 10.0 and CUDNN 7.4.1 were used to drop the time training.

Be-cause of the small GPU the training configuration was set to use one image in every step and it was needed to use a small backbone (resnet50). On average the full execution of this model took 57 minutes.

3.2 Evaluation Metrics

In order to evaluate the performance of the models, we used the following classification and segmentation measures: precision, recall, Dice coefficient and mean Intersection over Union (mIoU). These metrics are also used in the medical field, and are defined be-low.

Precision is the radio of pixels correctly predicted as COVID-19 divided by the total predicted as COVI-19:

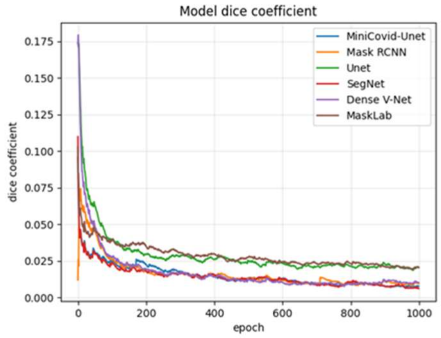


Fig. 8. Loss during the training phase

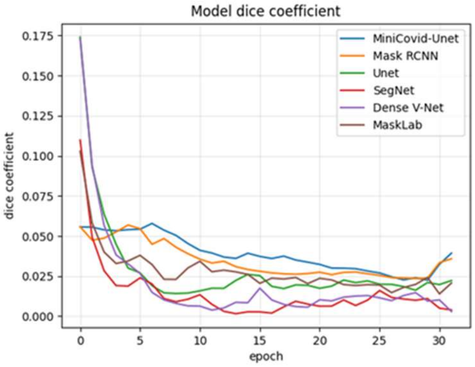


Fig. 9. Loss during the validation phase

$$Precision = \frac{TP}{TP+FP}, \quad (1)$$

where TP is the true positive (i.e., the number of pixels labeled as COVID-19 correctly) and FP refers to the false positive (i.e., the number of pixels labeled as COVID-19 wrong).

Recall is the radio of pixels correctly predicted as COVID-19 divided by total number of actual COVID-19:

$$Recall = \frac{TP}{TP+FN}, \quad (2)$$

where FN refers to the false negatives (i.e., the pixels that are labeled wrong as non-COVID-19).

However, these two measures are not frequently used as evaluation metrics because of their sensibility to segment size, in other words, they penalize errors in small segments more than in large ones [28, 34, 35].

Dice coefficient or *Dice score* (DSC) is a metric for image segmentation:

$$Dice = \frac{2|A \cap B|}{|A|+|B|}, \quad (3)$$

where A and B refers to the predicted and ground truth masks.

4 Results and Discussion

All models that we have used in this work predict a probability for every pixel and we have to set a threshold in order to identify if a pixel is in the segment of COVID-19 or is in the healthy part. So, we have decided that the threshold value of 0.9 is the best to do the Task.

We used the validation method five-fold cross validation to evaluate the segmentation performance of the models on the COVID-19 dataset. First of all, we set aside 10 im-ages to test the model after we have trained it. With the remaining 90 images, the new data set is used to apply five-cross validation.

We divided the new dataset into 5 parts, one of which was selected as the validation set and the other four parts were used for the training set in order to train the model. When the training had finished the loss, metrics were calculated and we repeated all the experiments until all the parts were used as a validation set, then the average of metrics was calculated to get the performance evaluation value of the model.

Figures 8 and 9 show the loss during the training and validation phase. At the beginning of the training phase, the difference between all the models is noticeable, but over time, all the ones converge. The models only detect where the lesion is, so we don't ask about the class of lesion.

Table 1 shows the metrics to evaluate the performance of the model. The Dice metric can be used to compare predicted segmentation pixels and their corresponding ground truth. Dense V-Net is the model that has the best performance in terms of metric accuracy. On the other hand, the proposed model achieves a better performance with respect to the Dice and Recall metric.

All models were able to detect the foreground from the back-ground; however, they were unable to detect the lesion class of the background. The best scores were obtained during the training phase compared to the validation phase, as can be seen in Figures 10 and 11.

Table 1. Performance metrics associated with different algorithms for the images in the testing dataset

Method	Dice	Precision	Recall
Mask R-CNN			
Unet	0.7801	0.7857	0.7333
SegNet	0.8202	0.6190	0.8667
Dense V-Net	0.8001	0.7667	0.8333
MaskLab	0.7905	**0.8667**	0.8467
	0.7885	0.8001	0.8402
Proposed	**0.8301**	0.8254	**0.8684**

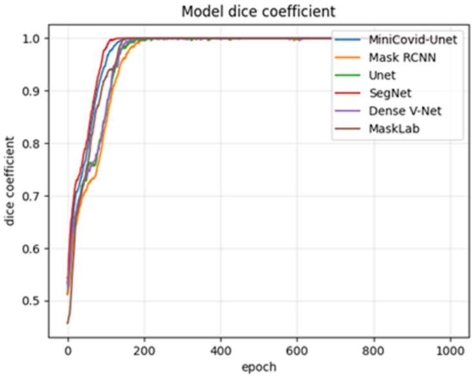


Fig. 10. Dice coefficient during the training phase

4.1 Inference

Figure 12 illustrates the segmentation results of lung infections from an example of lung CT slices taken from the test set using different segmentation networks.

The original image is on the left side (a), the expertly labeled mask is on the right side (b). All models have located the correct position on the image of the COVID-19 related injury, but do not retrieve the exact shape of the injury.

Unet misses true infected areas with small sizes. Mask RCNN works better than Unet to determine the infected region, however, some tissues close to infections are segmented incorrectly. Segnet and Dense V-Net provide good performance in segmenting medium-sized infection regions, but several overestimates of normal tissues as infections.

MaskLab cannot provide full segmentation of some regions. On the contrary, the proposed MiniCovid-Unet provides superior performance to previous methods, regarding the recognition and segmentation of small and medium infections.

The shape of the infected area was complex and could be located anywhere within the image, the contrast between the infected and healthy parts was low.

In addition, the original Mask R-CNN model has been trained with millions of images of people and different types of objects, which could explain the low score against MiniCovid-Unet.

Furthermore, the other models were unable to retrieve the exact shape of the COVID-19 lesion, as can be seen in Table 1.

5 Conclusion and Future Work

In this paper, we propose the MiniCovid-Unet network with novel structure for COVID-19 infection region segmentation in lung CT slices. We also presented other models applied to detect and segment injuries related to COVID-19.

The models were selected because it is simple to implement for a custom dataset of images. However, a GPU is necessary in order to train the model in a reasonable time.

All models were able to identify the regions where lesions were found, but had difficulties in correctly segmenting the shape of the lesion. Figure 12 shows that a healthy lung could be differentiated from a diseased one, and even completely healthy lungs could be detected. However, the results for the segmentation task were poor.

Although the models can identify the injury, it does not indicate the type of injury. We used a small dataset available for the segmentation task, however the MiniCovid-Unet's results obtained so far in this work represent an alternative to use deep learning to help in the objective diagnosis of COVID-19 using CT images of the Lung.

As future work, we want to get more images to train the framework. We also hope to be able to perform the segmentation taking into account the three existing classes in the dataset.

It is also proposed to make a comparison against other models such as U-Net++ [36], which are frameworks focused on COVID-19 medical images.

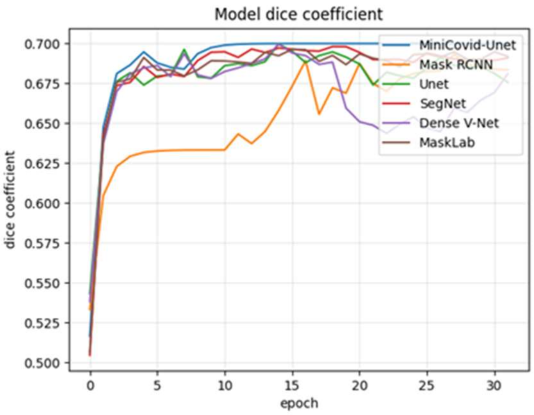


Fig. 11. Dice coefficient during the validation phase

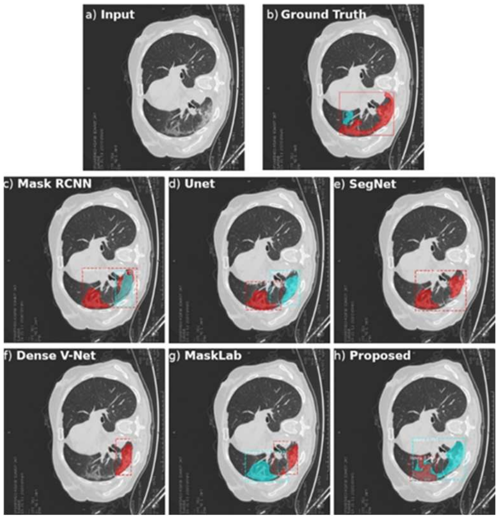


Fig. 12. Visual comparison of COVID-19 segmentation results. From left to right, (a) the first image is input CT scans, (b) the second is mask label visualization image or ground truth. The models (c) to (h) are the ones listed in Table 1. The color labeled part is the infected area

Acknowledgments

First author thanks to CONAHCYT for the scholarship granted. This research received no external funding.

References

1. **Platto, S., Wang, Y., Zhou, J., Carafoli, E. (2021).** History of the COVID-19 pandemic: Origin, explosion, worldwide spreading. Biochemical and Biophysical Research Communications, Vol. 538, pp. 14–23. DOI: 10.1016/j.bbrc.2020.10.087.
2. **WHO (2020).** Director-General's opening remarks at the media briefing on COVID-19. https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020.
3. **WHO (2020).** Coronavirus (COVID-19) Dashboard, 2020. https://covid19.who.int/.
4. **Li, S., Li, S., Disoma, C., Zheng, R., Zhou, M., Razzaq, A., Liu, P., Zhou, Y., Dong, Z., Du, A., Peng, J., Hu, L., Huang, J., Feng, P., Jiang, T., Xia Z. (2021).** SARS-CoV-2: Mechanism of infection and emerging technologies for future prospects. Reviews in Medical Virology, Vol. 31, No. 2, pp. e2168.31. DOI: 10.1002/rmv.2168.
5. **Cheng, V. C. C., Wong, S. C., Chen, J. H. K., Yip, C. C. Y., Chuang, V. W. M., Tsang, O. T. Y., Sridhar, S., Chan, J. F. W., Ho, P. L., Yuen, K. Y. (2020).** Escalating infection control response to the rapidly evolving epidemiology of the coronavirus disease 2019 (COVID-19) due to SARS-CoV-2 in Hong Kong. Infection Control & Hospital Epidemiology, 41, 493–498. DOI: 10.1017/ice.2020.58.
6. **Chen, M., Tu, C., Tan, C., Zheng, X., Wang, X., Wu, J., Huang, Y., Wang, Z., Yan, Y., Li, Z., Shan, H., Liu, J., Huang, J. (2020).** Key to successful treatment of COVID-19: accurate identification of severe risks and early intervention of disease progression. MedRxiv. DOI: 10.1101/2020.04.06.20054890.
7. **Liu, Z., Jin, C., Wu, C. C., Liang, T., Zhao, H., Wang, Y., Wang, Z., Li, F., Zhou, J., Cai, S., Zeng, L., Yang, J. (2020).** Association between Initial chest CT or clinical features and clinical course in patients with coronavirus disease 2019 pneumonia. Korean J Radiology, Vol. 21, No. 6, pp. 736–745. DOI: 10.3348/kjr.2020.0171.
8. **Zhou, X., Pu, Y., Zhang, D., Xia, Y., Guan, Y., Liu, S., Fan, L. (2022).** CT findings and dynamic imaging changes of COVID-19 in 2908 patients: A systematic review and meta-analysis. Acta Radiologica, Vol. 63, No. 3, pp. 291–310. DOI: 10.1177/0284185121 992655.
9. **Suri, J. S., Agarwal, S., Chabert, G. L., Carriero, A., Paschè, A., Danna, P. S. C., Saba, L., Mehmedović, A., Faa, G., Singh, I. M., Turk, M., Chadha, P. S., Johri, A. M., Khanna, N. N., Mavrogeni, S., Laird, J. R., Pareek, G., Miner, M., Sobel, D. W., Balestrieri, A. (2022).** COVLIAS 1.0 lesion vs. medseg: An artificial intelligence framework for automated lesion segmentation in

COVID-19 lung computed tomography scans. Diagnostics, Vol. 12, No. 5, pp. 1283. DOI: 10.3390/diagnostics12051283.

10. **Shi, H., Han, X., Jiang, N., Cao, Y., Alwalid, O., Gu, J., Fan, Y., Zheng, C. (2020).** Radiological findings from 81 patients with COVID-19 pneumonia in Wuhan, China: a descriptive study. The Lancet. Infectious Diseases, Vol. 20, pp. 425–434. DOI: 10.1016/S1473-3099(20)30086-4.
11. **Ye, Z., Zhang, Y., Wang, Y., Huang, Z., Song, B. (2020).** Chest CT manifestations of new coronavirus disease 2019 (COVID-19): A pictorial review. European Radiology, Vol. 30, pp. 4381–4389. DOI: 10.1007/s00330-020-06801-0.
12. **Al-Shehri, W., Almalki, J., Mehmood, R., Alsaif, K., Alshahrani, S. M., Jannah, N., Alangari, S. (2022).** A novel COVID-19 detection technique using deep learning-based approaches. Sustainability, Vol. 14, No. 19, DOI: 10.3390/su141 912222.
13. **LeCun, Y., Bengio, Y., Hinton, G. (2015).** Deep learning. Nature, Vol. 521, pp. 436. DOI: 10.1038/nature14539.
14. **Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., Sánchez, C. I. (2017).** A survey on deep learning in medical image analysis. Medical Image Analysis, Vol. 42, pp. 60–88. DOI: 10.1016/j.media.2017 .07.005.
15. **Gatys, L. A., Ecker, A. S., Bethge, M. (2017).** Texture and art with deep neural networks. Current Opinion in Neurobiology, Vol. 46, pp. 178–186. DOI: 10.1016/j.conb.2017.08.019.
16. **Wang, S., Yang, D. M., Rong, R., Zhan, X., Fujimoto, J., Liu, H., Minna, J., Wistuba, I. I., Xie, Y., Xiao, G. (2019).** Artificial intelligence in lung cancer pathology image analysis. Cancers, Vol. 11, No. 11, pp. 1673. DOI: 10.3390/cancers11111673.
17. **Shelhamer, E., Long, J., Darrell, T. (2017).** Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, pp. 640–651. DOI: 10.1109/TPAMI.2016.2572683.
18. **Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D. (2022).** Image segmentation using deep learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 44, pp. 3523–3542. DOI: 10.1109/TPAMI.2021.3059968.
19. **Ren, S., He, K., Girshick, R., Sun, J. (2017).** Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, pp. 1137–1149. DOI: 10.1109/TPAMI.2016. 2577031.
20. **Luján-García, J., Villuendas-Rey, Y., López-Yáñez, I., Camacho-Nieto, O., Yáñez-Márquez, C. (2021).** Nanochest-net: A simple convolutional network for radiological studies classification. Diagnostics, Vol. 11. No. 5, pp. 775. DOI: 10.3390/diagnostics11050775.
21. **Constantinou, M., Exarchos, T., Vrahatis, A. G., Vlamos, P. (2023).** COVID-19 classification on chest X-ray images using deep learning methods. International Journal of Environmental Research and Public Health, Vol. 20. No. 3, pp. 2035. DOI: 10. 3390/ijerph20032035.
22. **Teixeira, L. O., Pereira, R. M., Bertolini, D., Oliveira, L. S., Nanni, L., Cavalcanti, G. D. C., Costa, Y. M. G. (2021).** Impact of lung segmentation on the diagnosis and explanation of COVID-19 in chest X-ray images. Sensors, Vol. 21, No. 21, pp. 7116. DOI: 10.3390/s21217116.
23. **Wang, L., Lin, Z. Q., Wong, A. (2020).** COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images. Scientific Reports, Vol. 10, pp. 19549. DOI: 10.1038/s41598-020-76550-z.
24. **Badrinarayanan, V., Kendall, A., Cipolla, R. (2017).** SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, No. 12, pp. 2481–2495. DOI: 10.1109/TPAMI.2016.2644615.
25. **Jiang, Z. P., Liu, Y. Y., Shao, Z. E., Huang, K. W. (2021).** An improved VGG16 model for pneumonia image classification. Applied Sciences, Vol. 11. No. 23, pp. 11185. DOI: 10.3390/app112311185.
26. **Gibson, E., Giganti, F., Hu, Y., Bonmati, E., Bandula, S., Gurusamy, K., Davidson, B., Pereira, S. P., Clarkson, M. J., Barratt, D. C. (2018).** Automatic multiorgan segmentation on abdominal CT with dense V-Networks. IEEE Transactions on Medical Imaging, Vol. 37, No. 8, pp. 1822–1834. DOI: 10.1109/TMI.2018.2806309.
27. **Wu, J., Hu, W., Wen, Y., Tu, W., Liu, X. (2020).** Skin lesion classification using densely connected convolutional networks with attention residual learning. Sensors, Vol. 20, No. 24, pp. 7080. DOI: 10.3390/s20247080.
28. **Liu, X., Song, L., Liu, S., Zhang, Y. (2021).** A review of deep-learning-based medical image segmentation methods. Sustainability, Vol. 13. No. 3, pp. 1224. DOI: 10.3390/su13031224.
29. **Sharma, R., Saqib, M., Lin, C. T., Blumenstein, M. (2022).** A survey on object instance segmentation.

SN Computer Science, Vol. 3, No. 6, pp. 499. DOI: 10.1007/s42979-022-01407-3.

30. **COVID-19 CT Segmentation Dataset (2020).** http://medicalsegmentation.com/ covid19/.

31. **Hofmanninger, J., Prayer, F., Pan, J., Röhrich, S., Prosch, H., Langs, G. (2020).** Automatic lung segmentation in routine imaging is primarily a data diversity problem, not a methodology problem. European Radiology Experimental, Vol. 4, No. 1, pp. 1–3. DOI: 10.1186/s41747-020-00173-2.

32. **Xie, J., Pang, Y., Nie, J., Cao, J., Han, J. (2022).** Latent feature pyramid network for object detection. IEEE Transactions on Multimedia, Vol. 25, pp. 2153–2163. DOI: 10.1109/TMM.2022.3143707.

33. **Rostianingsih, S., Setiawan, A., Halim, C. I. (2020).** COCO (creating common object in context) dataset for chemistry apparatus. Procedia Computer Science, Vol. 171, pp. 2445–2452. DOI: 10.1016/j.procs.2020.04.264.

34. **Udupa, J. K., LeBlanc, V. R., Zhuge, Y., Imielinska, C., Schmidt, H., Currie, L. M., Hirsch, B. E., Woodburn, J. (2006).** A framework for evaluating image segmentation algorithms. Computerized Medical Imaging and Graphics, Vol. 30, No. 2, pp. 75–87. DOI: 10.1016/j.comp medimag.2005.12.001.

35. **Moorthy, J., Gandhi, U. D. (2022).** A Survey on medical image segmentation based on deep learning techniques. Big Data and Cognitive Computing, Vol. 6, No. 4, 117. DOI: 10.3390/bdcc 6040117.

36. **Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., Liang, J. (2020).** UNet++: redesigning skip connections to exploit multiscale features in image segmentation. IEEE Transactions on Medical Imaging, Vol. 39, No. 6, pp. 1856–1867. DOI: 10.1109/TMI.2019.2959609.

Article received on 06/09/2023; accepted on 18/10/2023.
[*]Corresponding author is Cornelio Yáñez-Márquez.

ISSN 2007-9737

Multi-Instrument Based N-Grams for Composer Classification Task

Alexander Gelbukh[1], Daniel Alejandro Pérez Alvarez[1],
Olga Kolesnikova[1], Liliana Chanona-Hernandez[*,2], Grigori Sidorov[1]

[1] Instituto Politécnico Nacional,
Centro de Invetigación en Computación,
Mexico

[2] Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco,
Mexico

lchanona@gmail.com

Abstract. In this research, we address the composer classification supervised problem from a Natural Language Processing perspective. Starting from digital symbolic music files, we build two representations: a class representation and other based on MIDI pitches. We use the technique of n-grams to build feature vectors of musical compositions based on their harmonic content. For this, we extract n-grams of size 1 to 4 in harmonic direction, differentiating between all possible subsets of instruments. We populate a term-frequency matrix with the vectors of compositions and we classify by the means of Support Vector Machines (SVM) classifier. Different classification models are evaluated, e.g., using feature filters and varying hyperparameters such as TF-IDF formula, among others. The results obtained show that n-grams based on MIDI pitches perform slightly better than n-grams based on class representation in terms of overall results, but the best result of each one of these representations is identical. Some of our best models reach accuracy results that exceed previous state of the art results based on a well-known dataset composed of string quartets by Haydn and Mozart.

Keywords. Composer classification, composer attribution, composer recognition, composer identification, composer style, n-grams, harmonic n-grams, string quartet, mozart, haydn.

1 Introduction

Starting from a list of composers and a list of compositions, the task of composer classification is defined as automatically assign each composition to the correct composer. It can be approached in unsupervised or supervised manner, but the last its a more known task. There are two types of formats to represent music digitally.

These are audio files and symbolic files. Audio files store recorded sound and are used in the field of Signal Processing. Our approach is more close to Natural Language Processing field. That's why we use symbolic files, which store information similar to a stave, as the composer intended it.

Different names have been given to this task. For example, composer style [30, 29], composer identification [21, 22], composer recognition [44, 42] and composer attribution [38, 40].

While several datasets have been tested for the symbolic domain, comparing Mozart and Haydn remains a challenge due to the similarities between these composers [39, 21]. These two composers admired and influenced each other [36, 4, 11].

Datasets involving Mozart and Haydn are often the most difficult to classify [39, 9, 43]. According to Hillewaere et al. and Kaliakatsos-Papakostas

doi: 10.13053/CyS-28-1-4903

et al. [15, 22] the models for solving classification tasks related to music (in symbolic format) can be grouped into two large categories.

The first category includes models based on global features or statistical descriptors which express each piece as a vector of features. Each feature or descriptor represents the measurement of a certain musical element throughout the entire piece, for example, frequency of major second intervals, average pitch of notes, etc.

Among the works that are part of this category we can mention [2, 38, 23]. The second category involves predictive models such as n-grams or Markovian models, which are based on the counting or prediction of local events. An event can be the interval between two consecutive notes or the duration ratio between two consecutive notes, etc. Examples of studies in this category include [44, 16, 19].

In recent years, deep learning based models for composer classification [43, 45, 7, 8, 24] may also worth a category on his own. If we look at the more abroad music classification task we find deep learning approaches based on Convolutional Neural Network (CNN) [43], Residual Neural Network [24, 8] and transformer-based architectures [45, 7, 47]. This open exciting new possibilities to represent music but this types of models usually need a lot of samples to train and many resources to process.

For example, when trying to apply CNN to a Mozart and Haydn dataset, the efforts by Verma and Thickstun [43] were unsuccessful, probably due to the small size of the dataset, which could introduce overfitting in deep learning models, and also the need for applying Leave One Out (LOO) cross validation in order to compare with previous works, which multiplies the processing resources needed.

Our approach is based on n-grams, but instead of the more common melody-based methods [44, 16, 19], we use harmony-based n-grams. Our goal is to investigate whether harmonic content can be a good predictor for the composer classification task. This document is structured as follows.

In Subsection 1.1, we present the dataset used. In Section 2, we discuss related work. In Section 3, we show our method. In Section 4, we discuss our results. Finally, in Section 5, we give our conclusions.

1.1 Dataset

The dataset we use is composed by 107 movements from string quartets by Haydn and Mozart. It is a balanced dataset, since there are 54 movements by Haydn and 53 by Mozart. This dataset was collected by Van Kranenburg and Backer [39] in **kern format. This format store musical information about pitch, duration, dynamics etc., in a similar fashion to a musical staff.

The **kern files can be converted into other popular formats of symbolic music such as MIDI [20]. The data was gathered from two different sources: Musedata[1] and KernScores[2] and generated at the Center for Computer Assisted Research in the Humanities at Stanford University[3]. According to our research, 16 more works have been built around this dataset or similar, using accuracy as the measure in all cases.

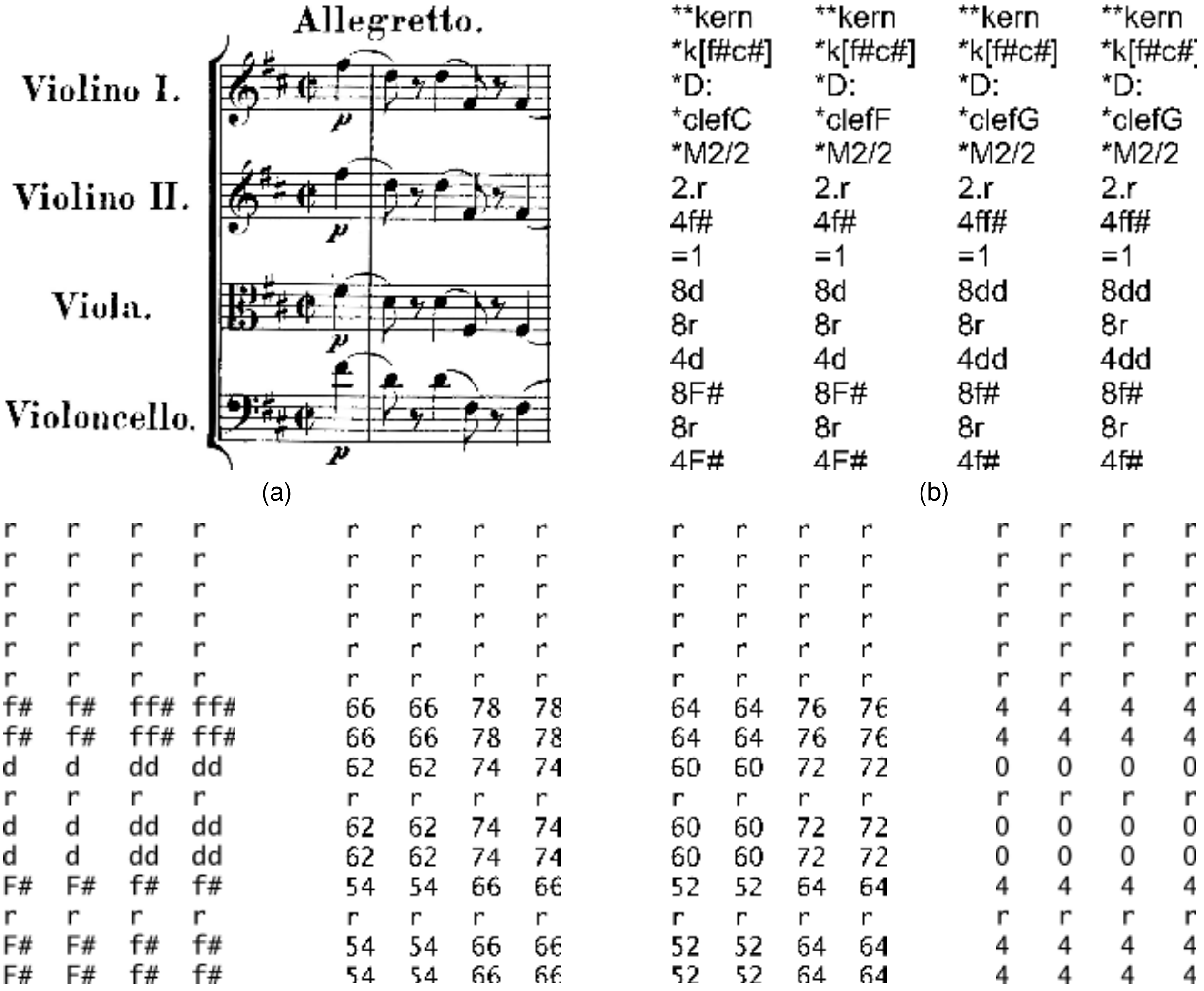
2 Related Work

In 2005, Van Kranenburg and Backer made an important contribution to the field of Music Information Retrieval MIR [39].

Instead of characterizing the style of composers from different periods [33] or differentiating the work of a composer among a group of composers [6], these researchers posed the problem of differentiating two contemporary and very similar composers such as the case of Mozart and Haydn.

Several research has been made in this regard, but solutions to this problem still has room for improvement.

[1] musedata.org
[2] kern.ccarh.org
[3] www.ccarh.org



(a)

```
**kern    **kern    **kern    **kern
*k[f#c#]  *k[f#c#]  *k[f#c#]  *k[f#c#
*D:       *D:       *D:       *D:
*clefC    *clefF    *clefG    *clefG
*M2/2     *M2/2     *M2/2     *M2/2
2.r       2.r       2.r       2.r
4f#       4f#       4ff#      4ff#
=1        =1        =1        =1
8d        8d        8dd       8dd
8r        8r        8r        8r
4d        4d        4dd       4dd
8F#       8F#       8f#       8f#
8r        8r        8r        8r
4F#       4F#       4f#       4f#
```

(b)

```
r   r   r   r
r   r   r   r
r   r   r   r
r   r   r   r
r   r   r   r
r   r   r   r
f#  f#  ff# ff#
f#  f#  ff# ff#
d   d   dd  dd
r   r   r   r
d   d   dd  dd
d   d   dd  dd
F#  F#  f#  f#
r   r   r   r
F#  F#  f#  f#
F#  F#  f#  f#
```

(c)

```
r   r   r   r
r   r   r   r
r   r   r   r
r   r   r   r
r   r   r   r
r   r   r   r
66  66  78  78
66  66  78  78
62  62  74  74
r   r   r   r
62  62  74  74
62  62  74  74
54  54  66  66
r   r   r   r
54  54  66  66
54  54  66  66
```

(d)

```
r   r   r   r
r   r   r   r
r   r   r   r
r   r   r   r
r   r   r   r
r   r   r   r
64  64  76  76
64  64  76  76
60  60  72  72
r   r   r   r
60  60  72  72
60  60  72  72
52  52  64  64
r   r   r   r
52  52  64  64
52  52  64  64
```

(e)

```
r   r   r   r
r   r   r   r
r   r   r   r
r   r   r   r
r   r   r   r
r   r   r   r
4   4   4   4
4   4   4   4
0   0   0   0
r   r   r   r
0   0   0   0
0   0   0   0
4   4   4   4
r   r   r   r
4   4   4   4
4   4   4   4
```

(f)

Fig. 1. Steps to final representations

2.1 Composer Classification

Van Kranenburg and Backer [39] evaluate the effectiveness of harmony and counterpoint-based features for detecting the style of the composers Bach, Handel, Telemann, Haydn and Mozart.

To do this, they extract 20 features, among which are: fraction of dissonant sounds, average number of active voices, number of different harmonic intervals between pairs of voices, number of parallel thirds, fourths and sixths, etc.

The k-means clustering, k-nearest neighbor and decision tree classifiers are used on this feature vector.

The researchers divide the dataset into several subsets, of which the most difficult to classify turn out to be those containing the composers Haydn and Mozart.

The best results are obtained with the Nearest neighbor classification algorithm in conjunction with the Fischer transformation for dimension reduction.

Table 1. Accuracy best results of MIDI representation for each subset of instruments

insts	results	num feat	insts	results	num feat	instrums	results	num feat
1	65.42	48	1-3	86.92	1,208	1-2-3	**88.79**	8,685
2	70.09	38	1-4	82.24	519	1-2-4	84.11	9,982
3	66.36	38	2-3	75.7	574	1-3-4	83.18	2,102
4	78.5	44	2-4	84.11	650	2-3-4	85.98	7,946
1-2	77.57	1,247	3-4	82.24	566	1-2-3-4	87.85	25,526

Kempfert and Wong [23] explore the use of features derived from the sonata form, particularly the use of primary and secondary themes in the melodic development of the pieces.

As a complement, they add to the feature vector other style markers based on the rhythm and melody of individual voices and harmonic features that capture the interaction between voices, as well as global features of the pieces, such as average and standard deviation of pitch and duration of notes or the ratio of notes and rests during the piece. Some of these features are derived from previous studies by other researchers.

They apply a selection of features based on the Bayesian Information Criterion and classify using Bayesian Logistic Regression, obtaining state of the art results for the widely used dataset of string quartets by Haydn and Mozart. Their work demonstrates the usefulness of incorporating features that take into account the sonata structure.

2.2 Harmony Based Classification

Ogihara and Li [30] seek to characterize the style of composers through the chord progressions of their works. To do this, they use n-grams of previously simplified chords and assign weight to these n-grams depending on the duration of the chords.

These researchers also use a transposition system to ensure that the key is the same for all works. With the n-grams obtained, the researchers create profiles of composers and compare them with each other using the cosine similarity measure. Using this method, the researchers managed to automatically group jazz composers hierarchically, according to a chronological order.

Something that remained to be demonstrated in this research is whether the cosine similarity measure is the most appropriate for comparing composer profiles because the n-grams that make up the profiles suffer from dispersion.

Pérez-Sancho et al. [34] face the task of classifying musical genres. For this, they collect a corpus of pieces from three genres: popular, jazz and classical and three subgenres for each of these genres. These researchers use two methods to ensure the homogeneity of the data: to transpose the entire dataset to the same key or to represent the chords as degrees of the key.

They also use feature selection procedures based on Average Mutual Information (AMI) on the chord list of the training set. To classify, they compare the performance of the n-gram technique with the Naïve Bayes Classifier, obtaining slightly better results with n-grams for the data set of three genres and with Naïve Bayes for the data set of nine subgenres.

According to [13], the representation of musical structure can have a significant influence in the quality of the results of a computational analysis on a given dataset. In most of the reviewed research that faces musical classification tasks based on harmonic content, the classic symbolic nomenclature of chord representation is used (e.g., Cmaj, Cmin, Cdim etc.), or some representation variant.

An example of this is the use of classical chord labels [30, 34, 46, 13] or a binary notation based on the present notes [46]. Other transformations include chord simplification [30, 34, 13], enharmonic representation of chords [35] or the replacement of chords by degrees or tonal functions (1st degree, 5th degree etc.) [34].

Table 2. Accuracy best results of class representation for each subset of instruments

instrums	results	num feat	instrums	results	num feat	instrums	results	num feat
1	63.55	13	1-3	76.64	148	1-2-3	86.92	1,016
2	66.36	13	1-4	71.03	169	1-2-4	75.7	1,725
3	69.16	13	2-3	79.44	140	1-3-4	79.44	1,704
4	52.34	5	2-4	68.22	150	2-3-4	86.92	638
1-2	64.49	135	3-4	62.62	121	1-2-3-4	**88.79**	2,136

Some researchers also process the duration of the chords [30, 35, 13] and the vast majority use transposition to the same key [30, 34, 46, 13]. Our representation specifies not only the notes that are included in a given chord but also the order in which these notes are produced. That is, our representation includes information about which of the instruments produces a given note within the chord. In many cases, our representation even specifies which octave each note is in.

In this way our classifier can know at all times which instrument produces the tonic note, which instrument produces the 5th degree, etc. This information could be important to characterize the style of a certain composer. The representations based on chord labels discussed in the previous paragraph do not offer access to that information.

3 Method

Some aspects of our methodology are shared with our previous work [1], for example, setting a minimum note length, which aims to transform duration information into tonal information. But the way we generate n-grams is different because we base our method in harmony instead of melody. Other aspects are similar to those that have been observed in prior studies, including feature filtering and using popular machine learning classifiers such as SVM.

3.1 Preprocessing

We found some errors in **kern files and we fix them manually (see subsection 4.1). We also remove multiple stops keeping the highest note at any moment for each instrument, as is common practice [44, 26]. Besides this, we extend the representation, we convert the pitches from **kern to MIDI format, we transpose the compositions and we transform MIDI pitches into classes in an optional fashion. We explain all this steps in detail below.

3.2 Extended Representation

Because notes can have different duration is very hard to generate a n-gram model using more than one instrument simultaneously. That's why we establish a minimum note length, thus transforming long notes into many notes of minimum length.

Our goal is to convert duration information into pitch information. For instance, we can represent a quarter note as two quavers, a half note as four quavers and a whole note as eight quavers, if we define the quaver as minimum note length. However, with such election would be impossible to represent shorter length notes such as semiquavers.

That's why a much smaller base note must be established to avoid losing too much information. The downside to this is the high number of computational resources that a small minimum note length can take to process. We define as extended representation the action of converting a regular staff with regular note lengths into a representation with only minimum length notes.

Several researchers have tried somewhat similar procedures, for instance Pape et al. [31] (they also add one hot encoding representation) and Velarde et al. [42] (they add multi hot encoding).

Table 3. Comparison with the state of art (in the last 9 works the same composers were used but with different datasets)

Comparison with SOTA	
4-grams (Class representation)	88.79
Alvarez et al. (2024) [23]	86.92
Kempfert and Wong (2020) [23]	84.11
Lostanlen (2018) [26]	82.2
Velarde et al. (2016) [42]	80.4
Van Kranenburg and Backer (2005) [39]	79.4
Hillewaere et al. (2009) [16]	79.4
Velarde et al. (2018) [41]	79.4
Hajj et. al. (2018) [10]	82.9
Kempfert and Wong (2020) [23]	82.46
Herlands et al. (2014) [14]	80.0
Hillewaere et al. (2010) [17]	75.4
Hontanilla et al. (2013) [19]	74.7
Dor and Reich (2011) [9]	73.75
Pape et al. (2008) [31]	73.5
Taminau et. al. (2010) [37]	73.0
Kaliakatsos et al. (2011) [22]	70.0

3.3 Transposition

Bringing all compositions of the dataset to the same key it's required to avoid being conditioned when classifying by the diverse original tonalities of the pieces. Otherwise, instead of classifying by composers, we probably would be classifying by tonalities.

This follows Wolkowsky's view [44], which calls for independence between the features of the feature vector and the tonalities of the pieces of dataset. There are ways of avoiding transposition, for instance using intervals between consecutive notes [44, 15, 16, 19, 10].

However, we could lose some information with this type of representation. For example, the intervals C-G and F-C are both perfect fifth intervals, so they are represented in the same way in a interval-based representation.

On the other hand, in a transposed based representation, they are represented differently. The interval C-G being equivalent to going from the tonic to the fifth degree and the interval F-C being equivalent to going from the fourth degree to the tonic.

3.4 Class Pitch vs MIDI Pitch

As an optional step in the representation, we add a class representation. To do so, we use 12 different symbols representing each of the musical notes and one special symbol for rests. We use the modulo 12 operation in order to normalize MIDI pitch into classes. This results in a reduction of the MIDI values in degrees or functions (fifth, fourth, third etc). For instance, after normalization, MIDI values 48, 60 and 72 would be represented as class 0, corresponding to tonic C, and MIDI values 55, 67 and 79 would be equivalent to class 7, corresponding to fifth degree G. This representation is compared with a representation that preserves the original MIDI values in Section 4.

3.5 Example of Feature Generation

We summarise the above steps with an example. Figure 1 shows the processing of the first two bars of the first movement of Mozart's string quartet k499. Clause a) shows the representation on the staff, as conceived by the composer. In clause b) the same piece is shown, but now in **kern format.

At the top of this format, some metadata can be observed (character '*') and at the bottom the notes can be seen. Each instrument is represented by a column and the '=' character denotes the beginning of a new measure. The length of each note is represented with a number and the pitch with a combination of letters and symbols.

Clause c) shows how the piece looks after "extending" the representation, we use a quaver as the minimum base note in this example. We discard unused information contained in the **kern format. The piece in this example begins with an anacrusis, hence the first thing represented is the dotted half note rest that precedes the quarter note f#.

Table 4. Most important features for class Haydn and class Mozart based on our two best models

MIDI	H	60-52-52, 65-62-53, 67-59-59, 83-r-r, 64-55-55, 64-58-55, r-71-67, 62-53-53, 62-57-57, 65-59-62
rep	M	60-57-50, 72-57-52, 71-59-55, 72-57-54, 74-62-r, 62-47-43, 62-50-47, 76-67-60, r-64-55, 72-60-r
Class	H	5-2-5-2, 0-4-4-0, 2-9-2-5, 0-4-4-7, 7-0-4-4, 11-5-11-7, 6-9-0-7, 2-9-9-5, 6-2-0-r, 9-0-0-5
rep	M	9-5-0-5, 2-7-11-5, r-r-2-2, 10-4-0-0, 4-7-0-r, 5-11-7-7, 2-7-11-2, 5-2-9-2, 11-7-7-4, 2-9-0-6

It can be seen that in this extended representation the eighth notes are not modified, the quarter notes are doubled and the dotted half note rest is replaced by 6 eighth note rests. Clause d) shows a representation similar to that in clause c), only the pitches in **kern format have been replaced by MIDI values.

Clause e) shows the transposition of these MIDI values to the scale of C. Since the original scale is D, we subtract 2 from each MIDI value. The rests remain unchanged. Finally, clause f) shows the optional step of converting the already transposed MIDI values into class values. To achieve this, we calculate each MIDI value by its modulo 12. Once again, the rests are not modified.

3.5.1 Instrument Filtering

Once we have all preprocessing done we can start to generate features. We use a simple n-gram alike method, but instead of words we use groups of minimum length notes in harmonic direction.

Since we have reduced the content of each instrument to a single note, in our dataset of string quartets must be a maximum of 4 voices playing (or at rest) at any given moment (see 1). Instead of always using all of these 4 voices, we build models for all combinations of instruments and we create n-grams based solely on the current subset of instruments.

For example, we can have models that use only viola, models that use only viola and cello, models that use all voices except for viola etc. In this way we can analyse based on results if there is a particular instrument, or combination of instruments that make the results stand out.

3.5.2 N-grams Generation

Continuing with the example of feature generation, in figure 1 after building MIDI representation in clause e) and class representation in clause f), we can generate $2^4 - 1 = 15$ n-grams in harmonic (horizontal) direction for each line (time).

Each one of these n-grams will be made from notes solely from a subset of instruments, so if the current model uses only first violin and viola, then we only use columns 1 and 3 to generate a 2-gram and if the current model uses all instruments except for second violin then we only use columns 1, 3 and 4 to generate a 3-gram.

We use symbol '-' to concatenate from left to right the notes within the n-gram. For instance, for a model using all instruments and class representation the following 4-grams are generated: r-r-r-r, r-r-r-r, r-r-r-r, r-r-r-r, r-r-r-r, r-r-r-r, 4-4-4-4, 4-4-4-4, 0-0-0-0... etc.

For a model based on MIDI representation using only first violin and viola the following 2-grams are generated: r-r, r-r, r-r, r-r, r-r, r-r, 64-76, 64-76, 60-72, ..., etc. and we ignore the information from remaining columns (for that particular model).

Counting the number of occurrences of each of the n-grams generated we fill the feature vector that identifies each composition.

3.6 Feature Filtering

Once each sample is represented as a vector of occurrences of n-grams of groups of instruments, we can optionally apply a simple feature selection criterion. To do this, we set 3 thresholds, a threshold that filters the most frequent n-grams, a threshold that filters the most infrequent n-grams

Table 5. Misclassified scores

MIDI	H	op103-01, op20n3-01, op20n6-04, op50n1-04, op64n1-03, op64n4-04, op71n1-04, op76n4-01, op76n4-03
rep	M	k138-03, k159-02, k168-04
Class	H	op1n0-05, op20n3-03, op20n6-04, op33n6-02, op50n2-01, op64n1-02, op64n4-04, op76n4-01
rep	M	k159-03, k168-02, k168-03, k465-03

and a threshold that limits the number of features. This is somewhat similar to the way text is processed by removing stop words and rare words. We can also alter the TF-IDF formula or the type of normalization of the feature vector.

Thus, we can compare models with different degrees of filtering, different norm and different variations of TF-IDF formula. For more on this, see subsection 4.1.

3.7 Classification Parameters

We accommodate some NLP concepts to the field of music with the goal of building the Term-document matrix [27, 28]. These concepts are enumerated below:

1. Term frequency (TF): $\mathrm{tf}(t,\, c)$, the frequency of occurrence of term t in composition c.
2. Inverse document frequency (IDF): $\log[n/\,\mathrm{df}(t)] + 1$, where n is the total number of compositions and $\mathrm{df}(t)$ is the number of compositions in which term t is present.
3. TF-IDF: $\mathrm{TF}(t,\, c) \times \mathrm{IDF}(t)$, TF multiplied by IDF.
4. Sublinear scaling: $1 + \log(\mathrm{TF})$, is used as optional replacement for TF.
5. IDF smoothing: $\log\Big((1+n)/\big(1+\mathrm{df}(t)\big)\Big) + 1$, is used as optional replacement for IDF.
6. Normalization L1: set the sum of values of the feature vector equal to 1.
7. Normalization L2: set the sum of squares of values of the feature vector equal to 1.

As can be observed, we have replaced documents (for which the original formula was created) for musical compositions. I the case of terms, we have defined them as minimum length groups of notes in harmonic direction, but it also could be intervals between notes, note duration or any other element taken from compositions.

4 Results and Discussion

4.1 Quartet Classification

We selected SVM with linear kernel, a classifier commonly used for text classification, for quartets classification. Given the computationally expensive process resulting from vectors with high dimensionality, we avoided SVM with RBF kernel.

To ensure that the vector size is uniform for all samples, n-grams which have not been observed in a particular composition are added to the vector with an occurrence of zero. We chose to transpose all the movements to the key of C major.

Changes in tonality may occur within each movement, that is a peculiarity of string quartet format. These changes were carefully considered during the transposition process. The **kern files were found to contain a certain amount of encoding errors.

For example, a file whose lines do not match the humdrum encoding, files or some bars of files with more instruments than the established ones (we removed the extra column), files with duplicate instruments (we ignored the extra columns) and files with duration errors (see [16, 17].

We use Python as the programming language for our method.

We use scikit-learn [32] and nltk [3] libraries, as well as Support Vector Classifier (SVC), CountVectorizer and TfidfTransformer methods of these libraries. We established the values **C** for SVC as **C=1,000** and minimum note length **g** as **g=192**. We created models for both MIDI and class representation.

For each representation, we developed models for each of the 15 subsets of instruments (first violin, second violin, viola, cello, first violin and second violin, first violin and viola etc.).

For each subsets of instruments, we tested models with and without IDF, sublinear scaling and IDF smoothing, different types of norms (L1 and L2), values of 0, 5 and 10 for minimum document frequency and values of L, L-5 and L-10 for maximum document frequency, where L is the length of the dataset. We use leave one out cross validation to compare our results with previous research (See Table 3).

4.2 Analysis of Results

The code and the results for the present work can be seen at the following address[4]. Here we present the best results for each each subset of instruments and each representation. Table 1 shows the best results obtained for each instrument subset using the MIDI pitch representation. Columns of the table show in order the instruments used, the results and the average number of features.

Most of the models listed in this table obtained good results, above 80% of accuracy. The best result derived from the MIDI representation (88.79) is based on a model formed by the union of the first and second violin and viola. The parameters of this model are as follows:

L2 norm with sublinear scaling in conjunction with IDF smoothing, a minimum document frequency equal to 0 and a maximum document frequency equal to 97.

On the other hand, the cello was the most predictive instrument among the models made up of individual instruments, with a wide difference over the rest. It is worth highlighting the result of the model formed by the first violin and viola (86.92) with just over 1200 features. Several of the models presented in this table outperform previous state of the art results. Table 2 shows the best results obtained for each instrument subset using the class representation. Compared to the MIDI pitch representation, most results are inferior.

The best result of this representation (88.79) equals the best result of the MIDI representation but with a smaller number of features. Among the parameters of this best result, we used the L2 norm and sublinear scaling in conjunction with the inverse document frequency, and we also set the minimum document frequency to 5 and the maximum document frequency to 107.

Among other notable results, we can mention the mixture of second violin and viola (79.44) with only 140 features and the union of second violin, viola and cello (86.92) with only 638 features. Several models based on this representation also surpass the results of the state of the art.

Table 3 shows the state of the art results for the task of composer classification. The first part of the table shows the results of previous research using the same dataset we use. The second part of the table shows the research in which the same composers (Haydn and Mozart) are classified but using datasets different than ours. The results obtained in the present work outperform the results of the state of the art.

4.3 Musicological Analysis

Table 4 presents the 10 most important features for Haydn and Mozart based on our two best models. This was calculated using ELI5 Python library and taking into account all iterations of the cross validation process. The features are sorted by importance. First part of the table shows the features derived from trigrams of all instruments except for violoncello and MIDI representation.

It's difficult to make conclusions about the chords involved, since we are missing the lower voice but we can recognize in the Mozart's row a likely A minor in second position, a G major in sixth position and a C major in eighth position. In the case of features derived from the class representation model, it's easier to identify chords.

[4] github.com/dapalvarez/harmonic_ngrams

So, in the case of Haydn, we recognize a couple of D minor chords in first inversion, a couple of C major chords in second and first inversion, and an F major. In the case of Mozart, we can see an F major, a G major 7 in third inversion, a C major, a G major in second inversion, a D minor, an E minor and a D major 7 in first inversion.

Most of this chords are shared by both composers but differences arise in the order of the notes. When we compare the features from the two models, applying modulo 12 operation to the MIDI pitches and comparing them with the three first notes of the class derived features, we find some coincidences. So, for Haydn, feature '60-52-52' matches '0-4-4-0' and '0-4-4-7', feature '65-62-53' matches '5-2-5-2' and feature '62-57-57' matches '2-9-9-5'. For Mozart, feature '76-67-60' matches '4-7-0-r'. Besides, there are no coincidences between MIDI-based features belonging to one composer and class-based features belonging to the other composer, so we can glimpse points of contact between the procedures of both models.

Table 5 lists the compositions that were misclassified by the two models with which we obtained the best results. The first part of the table shows the pieces misclassified by the model formed by the union of first violin, second violin and viola, based on the MIDI pitch representation.

Most of the pieces listed in this segment are composed by Haydn. Among these is the fugal finale (4th movement) opus 20 number 6. This movement has similarities with another Mozart fugue, the fourth and final movement of the k168 quartet, also misclassified.

Heartz [12] asserts that Mozart's inclusion of a fugue at the conclusion of the K168 quartet was a result of his familiarity with Haydn's opus 20. Brown [5] refutes this, noting out that composing fugal ends is not just a Haydn practice, but rather a Viennese one.

On the other hand, Hontanilla [18] points out that fugues are popular throughout the Baroque era, which makes it difficult to classify because both composers use them to the same extent. The second part of the table 5 shows the compositions misclassified by the model formed by the union of all the instruments and based on the class representation.

As with the MIDI-based model, most of the misclassified pieces belong to Haydn. Several compositions were misclassified by both models. These are the fourth movement opus 64 number 4, the first movement opus 76 number 4 and the fourth movement opus 20 number 6, all by Haydn.

The third movement of the quartet k465, also misclassified, belongs to the "Haydn Quartets", a group of 6 string quartets that Mozart composed in honor of Haydn. Bonds [4], suggests that with the "Haydn Quartets", Mozart intended to show himself as the master Haydn's successor rather than attempting to copy his style.

Maybe that's why only one movement turned out to be misclassified among the 18 movements in the dataset derived from the "Haydn Quartets". Regarding this 3rd movement k465, La Rue [25] identifies a direct influence of Haydn on it.

The movements k168-02, k168-03, k168-04 included in the "Viennese Quartets" were composed by Mozart in 1973 in the city of Vienna. There are studies on similarities between the "Viennese Quartets" and other contemporary quartets by Haydn [5]. For example Heartz [12], identifies the quartets opus 9 and opus 17 as probable sources of inspiration for Mozart.

5 Conclusions

In the present manuscript, we expose a new approach to the supervised problem of composer recognition. We adapt to music field some concepts of NLP domain to create a vectorial representation of musical pieces based on n-grams extracted in harmonic direction. We compare the pros and cons of representing pitches using MIDI values or class values. We use the SMV classifier to achieve state of the art results on a dataset of string quartets by Mozart and Haydn.

We compare models with different subsets of hyperparameters such as norm, feature filtering values etc. and we give some musical insight about the misclassified scores. We think it would be interesting, as future work, to evaluate models which can combine different n-gram sizes. That is, models which integrate unigrams with trigrams, bigrams with 4-grams, etc.

Another proposal would be to apply our method to datasets used in other music classification tasks, such as emotion recognition or genre recognition. Finally, we propose to adapt recent discoveries from NLP field, for example transformer models such as BERT, to tackle the problem of composer classification.

Acknowledgments

The present work was done with partial support from the Mexican Government through the grant A1-S-47854 of CONACYT, Mexico, grants 20232138, 20231567 and 20232080 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico.

The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

References

1. **Alvarez, D. A. P., Gelbukh, A., Sidorov, G. (2024).** Composer classification using melodic combinatorial n-grams. Expert Systems with Applications, Vol. 249. DOI: 10.1016/j.eswa.2024.123300.

2. **Backer, E., van-Kranenburg, P. (2005).** On musical stylometry—a pattern recognition approach. Pattern Recognition Letters, Vol. 26, No. 3, pp. 299–309. DOI: 10.1016/j.patrec.2004.10.016.

3. **Bird, S., Klein, E., Loper, E. (2009).** Natural language processing with Python: Analyzing text with the natural language toolkit. O'Reilly Media, Inc.

4. **Bonds, M. E. (2007).** Replacing haydn: Mozart's "Pleyel" quartets. Music and Letters, Vol. 88, No. 2, pp. 201–225. DOI: 10.1093/ml/gcl150.

5. **Brown, A. P. (1992).** Haydn and Mozart's 1773 stay in Vienna: Weeding a musicological garden. Journal of Musicology, Vol. 10, No. 2, pp. 192–230. DOI: 10.2307/763612.

6. **Buzzanca, G. (2002).** A supervised learning approach to musical style recognition. Additional Proceedings of the Second International Conference on Music and Artificial Intelligence, Vol. 2002, pp. 167.

7. **Chou, Y. H., Chen, I., Chang, C. J., Ching, J., Yang, Y. H. (2021).** MidiBERT-piano: Large-scale pre-training for symbolic music understanding. arXiv. DOI: 10.48550/arXiv.2107.05223.

8. **Deepaisarn, S., Buaruk, S., Chokphantavee, S., Chokphantavee, S., Prathipasen, P., Sornlertlamvanich, V. (2022).** Visual-based musical data representation for composer classification. 17th International Joint Symposium on Artificial Intelligence and Natural Language Processing, pp. 1–5. DOI: 10.1109/iSAI-NLP56921.2022.9960254.

9. **Dor, O., Reich, Y. (2011).** An evaluation of musical score characteristics for automatic classification of composers. Computer Music Journal, Vol. 35, No. 3, pp. 86–97.

10. **Hajj, N., Filo, M., Awad, M. (2018).** Automated composer recognition for multi-voice piano compositions using rhythmic features, n-grams and modified cortical algorithms. Complex and Intelligent Systems, Vol. 4, No. 1, pp. 55–65. DOI: 10.1007/s40747-017-0052-x.

11. **Hatten, R. S. (2017).** Mozart's music of friends: Social interplay in the chamber works, Vol. 39. DOI: 10.1093/mts/mtx010.

12. **Heartz, D. (1995).** Haydn, Mozart, and the Viennese School, 1740-1780. WW Norton New York.

13. **Hedges, T., Roy, P., Pachet, F. (2014).** Predicting the composer and style of jazz chord progressions. Journal of New Music Research, Vol. 43, No. 3, pp. 276–290. DOI: 10.1080/09298215.2014.925477.

14. **Herlands, W., Der, R., Greenberg, Y., Levin, S. (2014).** A machine learning approach to musically meaningful homogeneous style classification. Vol. 28, No. 1. DOI: 10.1609/aaai.v28i1.8738.

15. **Hillewaere, R., Manderick, B., Conklin, D. (2009).** Global feature versus event models for folk song classification. Proceedings of the International Society for Music Information Retrieval Conference, Vol. 2009, pp. 729–733.

16. **Hillewaere, R., Manderick, B., Conklin, D. (2009).** Melodic models for polyphonic music classification. Proceedings of the Second International Workshop on Machine Learning and Music, pp. 19–24.

17. **Hillewaere, R., Manderick, B., Conklin, D. (2010).** String quartet classification with monophonic models. 11th International Society for Music Information Retrieval Conference, pp. 537–542.

18. **Hontanilla, M., Pérez-Sancho, C., Inesta, J. M. (2017).** Music style recognition with language models–beyond statistical results. Proceedings of the 10th International Workshop on Machine Learning and Music, pp. 31–36.

19. **Hontanilla, M., Pérez-Sancho, C., Iñesta, J. M. (2013).** Modeling musical style with language models for composer recognition. Iberian Conference on Pattern Recognition and Image Analysis, Pattern Recognition and Image Analysis, pp. 740–748. DOI: 10.1007/978-3-642-38628-2_88.

20. **Huron, D. (2002).** Music information processing using the humdrum toolkit: Concepts, examples, and lessons. Computer Music Journal, Vol. 26, No. 2, pp. 11–26. DOI: 10.1162/014892602760137158.

21. **Kaliakatsos-Papakostas, M. A., Epitropakis, M. G., Vrahatis, M. N. (2010).** Musical composer identification through probabilistic and feedforward neural networks. Proceedings of the European Conference on the Applications of Evolutionary Computation, pp. 411–420. DOI: 10.1007/978-3-642-12242-2_42.

22. **Kaliakatsos-Papakostas, M. A., Epitropakis, M. G., Vrahatis, M. N. (2011).** Weighted Markov chain model for musical composer identification. European Conference on the Applications of Evolutionary Computation, pp. 334–343. DOI: 10.1007/978-3-642-20520-0_34.

23. **Kempfert, K. C., Wong, S. W. K. (2020).** Where does Haydn end and Mozart begin? Composer classification of string quartets. Journal of New Music Research, Vol. 49, No. 5, pp. 457–476. DOI: 10.1080/09298215.2020.1814822.

24. **Kim, S., Lee, H., Park, S., Lee, J., Choi, K. (2020).** Deep composer classification using symbolic representation. Proceedings of the International Society for Music Information Retrieval Conference, pp. 1–3. DOI: 10.48550/arXiv.2010.00823.

25. **La-Rue, J. (2001).** The haydn-dedication quartets: Allusion or influence?. Journal of Musicology, Vol. 18, No. 2, pp. 361–373. DOI: 10.1525/jm.2001.18.2.361.

26. **Lostanlen, V. (2018).** Eigentriads and eigenprogressions on the tonnetz. arXiv. DOI: 10.48550/arXiv.1810.00790.

27. **Manning, C., Schutze, H. (1999).** Foundations of statistical natural language processing. MIT press.

28. **Manning, C. D., Raghavan, P., Schütze, H. (2009).** Introduction to information retrieval. Cambridge University Press.

29. **Mearns, L., Tidhar, D., Dixon, S. (2010).** Characterisation of composer style using high-level musical features. Proceedings of 3rd International Workshop on Machine Learning and Music, pp. 37–40. DOI: 10.1145/1878003.1878016.

30. **Ogihara, M., Li, T. (2008).** N-gram chord profiles for composer style representation. Proceedings of the 9th International

Conference on Music Information Retrieval, pp. 671–676.

31. **Pape, L., de-Gruijl, J., Wiering, M. (2008).** Democratic liquid state machines for music recognition. Speech, Audio, Image and Biomedical Signal Processing using Neural Networks, pp. 191–215. DOI: 10.1007/978-3-540-75398-8_9.

32. **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011).** Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, Vol. 12, No. 85, pp. 2825–2830. DOI: 10.48550/arXiv.1201.0490.

33. **Pollastri, E., Simoncelli, G. (2001).** Classification of melodies by composer with hidden Markov models. Proceedings of the 1st International Conference on WEB Delivering of Music, pp. 88–95. DOI: 10.1109/wdm.2001.990162.

34. **Pérez-Sancho, C., Rizo, D., Iñesta, J. M. (2009).** Genre classification using chords and stochastic language models. Connection Science, Vol. 21, No. 2–3, pp. 145–159. DOI: 10.1080/09540090902733780.

35. **Rohrmeier, M., Graepel, T. (2012).** Comparing feature-based models of harmony. Proceedings of the 9th International Symposium on Computer Music Modelling and Retrieval, pp. 357–370.

36. **Schmid, E. F., Sanders, E. (1956).** Mozart and Haydn. The Musical Quarterly, Vol. 42, No. 2, pp. 145–161. DOI: 10.1093/mq/XLII.2.145.

37. **Taminau, J., Hillewaere, R., Meganck, S., Conklin, D., Nowé, A., Manderick, B. (2010).** Applying subgroup discovery for the analysis of string quartet movements. Proceedings of 3rd International Workshop on Machine Learning and Music, Association for Computing Machinery, pp. 29–32. DOI: 10.1145/1878003.1878014.

38. **van-Kranenburg, P. (2006).** Composer attribution by quantifying compositional strategies. The International Society for Music Information Retrieval, pp. 375–376.

39. **van-Kranenburg, P., Backer, E. (2005).** Musical style recognition—a quantitative approach. Handbook of Pattern Recognition and Computer Vision, World Scientific, pp. 583–600. DOI: 10.1142/9789812775320_0031.

40. **van-Nuss, J., Giezeman, G. J., Wiering, F. (2017).** Melody retrieval and composer attribution using sequence alignment on RISM incipits. Proceedings of 9th International Conference on Technologies for Music Notation and Representation, pp. 1–7.

41. **Velarde, G., Cancino-Chacón, C., Meredith, D., Weyde, T., Grachten, M. (2018).** Convolution-based classification of audio and symbolic representations of music. Journal of New Music Research, Vol. 47, No. 3, pp. 191–205. DOI: 10.1080/09298215.2018.1458885.

42. **Velarde, G., Weyde, T., Chacón, C. E. C., Meredith, D., Grachten, M. (2016).** Composer recognition based on 2D-filtered piano-rolls. Proceedings of the 17th International Conference on Music Information Retrieval, pp. 115–121.

43. **Verma, H., Thickstun, J. (2019).** Convolutional composer classification. arXiv. DOI: 10.48550/arXiv.1911.11737.

44. **Wołkowicz, J., Kulka, Z., Kešelj, V. (2008).** N-Gram-based approach to composer recognition. Archives of Acoustics, Vol. 33, No. 1, pp. 43–55.

45. **Yang, D., Ji, K., Tsai, T. J. (2021).** A deeper look at sheet music composer classification using self-supervised pretraining. Applied Sciences, Vol. 11, No. 4, pp. 1387. DOI: 10.3390/app11041387.

46. **Yoshii, K., Goto, M. (2011).** A vocabulary-free infinity-gram model for nonparametric bayesian chord progression analysis.

Proceedings of the International Society for Music Information Retrieval Conference, pp. 645–650.

47. **Zeng, M., Tan, X., Wang, R., Ju, Z., Qin, T., Liu, T. Y. (2021).** MusicBERT: Symbolic music understanding with large-scale pre-training. Findings of the Association for Computational Linguistics, pp. 791–800. DOI: 10.48550/arXiv. 2106.05630.

Article received on 29/09/2023; accepted on 14/12/2023.
** Corresponding author is Liliana Chanona-Hernandez.*

ISSN 2007-9737

Datos mercadológicos del uso y consumo de las compras por internet (*e-commerce*) de los empresarios en pequeña escala en México

Lidia Ramírez-Lemus[*,1], Carlos Alberto Rodríguez-Rodríguez[2], José Miguel Barrón-Adame[3]

[1] Universidad Tecnológica del Suroeste de Guanajuato,
Licenciatura en Innovación de Negocios y Mercadotecnia,
México

[2] Universidad Politécnica de Guanajuato,
Licenciatura en Administración de Empresas,
México

[3] Universidad Tecnológica del Suroeste de Guanajuato,
México

{mbarrona, lramirez}@utsoe.edu.mx, carodriguezr@upgto.edu.mx

Resumen. Los avances tecnológicos, han generado cambios drásticos para muchos tipos de organizaciones, en especial para las Micros y pequeñas empresas (MiyPES). El objetivo que se plasma en esta investigación es analizar las herramientas que utilizan los empresarios en pequeña escala principalmente en el uso y consumo de las compras por internet (*e-commerce*) de productos y servicios en México. Se utilizó el método descriptivo-correlacional, con un instrumento de investigación de 36 *ítems* con escalas tipo *Likert*, para ello se contó con una muestra finita de 550 elementos. De acuerdo a los hallazgos encontrados se utilizó el modelo de regresión lineal como estadístico para correlacionar los datos. Se concluye que, el uso de la tecnología por los empresarios generó una correlación de .550, el consumo de programas tecnológicos fue de .157 y la habilidad en el uso del internet fue de .488, estas fueron las variables seleccionadas y prioritarias, que mostraron una correlación directa moderada con las compras por internet, con más del (62%) de confiabilidad de los datos, que fueron representativos de acuerdo al modelo que definen el perfil de compra para los usuarios como estrategia comercial.

Palabras clave. Uso y consumo, tecnologías, e-commerce, compras y empresarios.

Marketing Data on the Use and Consumption of Internet Purchases (E-Commerce) by Small-Scale Entrepreneurs In Mexico

Abstract. Technological advances have generated drastic changes for many types of organizations, especially for Micro and small businesses (MiyPes). The objective of this research is to analyze the tools used by small-scale entrepreneurs mainly in the use and consumption of internet purchases (e-commerce) of products and services in Mexico. The descriptive-correlational method was used, with a 36-item research instrument with Likert-type scales, for which a finite sample of 550 elements was used. According to the findings, the linear regression model was used as a statistic to correlate the data. It is concluded that the use of showed a moderate direct correlation with online purchases, with more than (62%) reliability of the data, which were representative according to the model that defines the purchase profile for users as a commercial.

Keywords. Use and consumption, technologies, e-commerce, purchases and entrepreneurs.

doi: 10.13053/CyS-28-1-4476

1. Introducción

La innovación ha tenido cambios muy radicales, debido al impulso de las nuevas tecnologías y las demandas propias de los consumidores; es por ello que las empresas han tenido que hacer ajustes en sus productos, procesos y estrategias mercadológicas, tales como el uso del internet, derivándose las plataformas virtuales, redes sociales, *internet on line,* comercio electrónico *(e-commerce), internet* en la nube; *apps* de aplicaciones sociales, entre otros.

Por lo que, el internet ha provocado la compra y venta de productos y servicios en diferentes partes del mundo, con mayores facilidades para los usuarios que continuamente utilizan esta herramienta y que requieren al instante de un bien o servicio de forma rápida y sencilla (Palomino Pita et al., 2020).

En el siglo XXI el comercio electrónico (*e-commerce*), se fue desarrollando de manera rápida y con la confianza de los clientes lograron conseguir lealtad hacia las empresas de acuerdo con Angenu et al. (2015); así se lograría la retención de los clientes, en especial con las compras por internet según Sánchez-Alzate y Montoya-Restrepo (2016).

Las TIC´s (Tecnologías de la Información y Comunicación) son ejemplo de esto, han trascendido drásticamente y han generado ventas altas en los últimos años, lo que ha ocasionado que los consumidores adopten nuevas formas de compra de una manera más rápida y eficiente en poco tiempo, lo que resulta atractivo y fácil para muchos a nivel mundial como lo aportan Dhanapal *et al*. (2015).

En México se han observado indicadores asociados al internet, se reflejó un alza del 24% en el año 2017 y 2018, alrededor de 83 millones de internautas, un 74% pertenecían a la tercera edad de los cuales fueron consumidores y comparadores por vía on line de acuerdo a Riquelme (2018).

En otro estudio, se encontró que los jóvenes realizan sus compras por plataformas vía *on line* con un 62%; lo que les permite mayor accesibilidad de manera inmediata, que el estar esperando tiempo para hacer pagos y recibir sus productos de manera física (Espinoza Delgado et al., 2020).

Así también, se observó en una publicación que, en México, el comercio electrónico (*e-commerce*) se colocó en la posición número 90 dentro de los 144 lugares empresas dedicadas a este rubro, sobre todo el índice de B2C (Business to Consumer) de comercio electrónico (NACIONES UNIDAS, 2017).

En este sentido, la presente investigación, está divida en 4 apartados: primeramente, se explora el contexto actual; así como, la importancia de contar con las tecnologías innovadoras que ayuden a dinamizar los procesos de compras de los pequeños empresarios que cuentan con escasas herramientas tecnológicas; sin embargo, estos hacen un esfuerzo por utilizarlas ya sea de manera personalizada o con ayuda de alguien.

En el segundo plano, se hace énfasis en la búsqueda de literatura, abarcando los temas adheridos a las tecnologías, plataformas *on line, e-commerce,* redes sociales, entre otros.

En una tercera parte se menciona los métodos donde se explica de manera detalla que esta investigación se realizó en campo eligiendo una muestra representativa, principalmente con empresarios de pequeña escala, posteriormente se presentan los resultados mediante un estadístico de regresión lineal planteando un modelo que determinará el comportamiento de los datos de las compras por internet y finalmente se termina con una conclusión haciendo alusión a las mejoras encontradas en esta investigación.

2. Revisión literaria

Con la pandemia causada por la COVID-19, se generó un cambio radical en los consumidores, pues se vieron obligados a no salir de sus hogares y hacer las compras por vía internet mediante el uso de la app móvil y/o computadora personal mediante estrategias comerciales como el *e-commerce* (Rodríguez et al., 2020)*;* éste último definido por la Procuraduría Federal del Consumidor (PROFECO), como un proceso de intercambio de bienes y servicios con la red del internet (PROFECO, 2016).

Así también considerado como fuente para la compra y venta de productos de la canasta básica

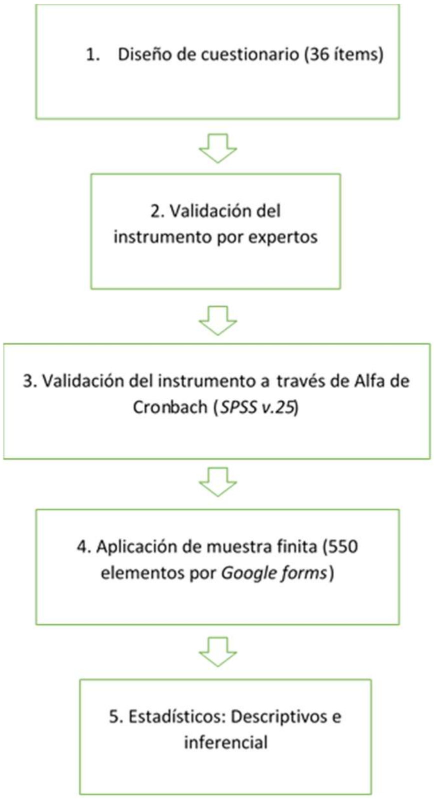

1. Diseño de cuestionario (36 ítems)
2. Validación del instrumento por expertos
3. Validación del instrumento a través de Alfa de Cronbach (*SPSS v.25*)
4. Aplicación de muestra finita (550 elementos por *Google forms*)
5. Estadísticos: Descriptivos e inferencial

Fig. 1. Estructura de la investigación: Fuente: Elaboración propia

Tabla 1. Estadísticos de fiabilidad

Alfa de Cronbach	N de elementos
.861	36

Fuente: Diseño propio mediante software SPSS v.25

Tabla 2. Estadísticos descriptivos

	Media	Desviación típica	N
COMPRAS_INTERNET_ECOMMERCE	29.2564	6.39599	550
USO_TECNOLOGÍA_INTERNET	2.3161	1.06964	550
CONSUMO_PROGRAMAS_TECNOLOGÍA	1.5106	.33641	550
HABILIDAD_INTERNET	3.2076	1.15635	550

Fuente: Diseño propio mediante software SPSS v.25

a nivel mundial (Palomino Pita et al., 2020); por lo que, los empresarios siguen estrategias de marketing acordes a los cambios momentáneos de los consumidores, para lograr continuidad en los negocios y el *e-commerce* constituye una ventaja de comercio por internet (Viramontes-Olivas et al., 2015); ya que se involucra la oferta y la demanda para los productos y servicios, asumiendo los riesgos como una buena estrategia de mercadotecnia (Sanz *et al*., 2018; Ramos Carrasco y Altamirano Morra., 2021).

Ahora bien, los dispositivos de IoT (Internet de las cosas) como una computadora, un celular, impresoras, cámaras de video, entre otros; están

Tabla 3. Correlaciones

		COMPRAS_ INTERNET _ECOMMERCE	USO_TECNOLOGÍA_ INTERNET	CONSUMO_PROGRAMAS_ TECNOLOGÍA	HABILIDAD_ INTERNET
Correlación de Pearson	COMPRAS_INTERNET _ECOMMERCE	1.000	.550	.157	.488
	USO_TECNOLOGÍA_ INTERNET	.550	1.000	.056	.415
	CONSUMO_PROGRAMAS _TECNOLOGÍA	.157	.056	1.000	.078
	HABILIDAD_INTERNET	.488	.415	.078	1.000

Fuente: Diseño propio mediante software SPSS v.25

Tabla 4. Variables introducidas/eliminadas[a]

Modelo	Variables introducidas	Variables eliminadas	Método
1	USO_TECNOLOGÍA_INTERNET	.	Por pasos (criterio: Prob. de F para entrar <= .050, Prob. de F para salir >= .100).
2	HABILIDAD_INTERNET	.	Por pasos (criterio: Prob. de F para entrar <= .050, Prob. de F para salir >= .100).
3	CONSUMO_PROGRAMAS_TECNOLOGÍA	.	Por pasos (criterio: Prob. de F para entrar <= .050, Prob. de F para salir >= .100).

a. Variable dependiente: COMPRAS_INTERNET_ECOMMERCE

Fuente: Diseño propio mediante software SPSS v.25

Tabla 5. Resumen del modelo[d]

Modelo	R	R cuadrado	R cuadrado corregida	Error típ. de la estimación	Durbin-Watson
1	.550[a]	.302	.301	5.34662	
2	.620[b]	.384	.382	5.02999	
3	.629[c]	.396	.393	4.98503	1.714

a. Variables predictoras: (Constante), USO_TECNOLOGÍA_INTERNET

b. Variables predictoras: (Constante), USO_TECNOLOGÍA_INTERNET, HABILIDAD_INTENET

c. Variables predictoras: (Constante), USO_TECNOLOGÍA_INTERNET, HABILIDAD_INTERNET, CONSUMO_PROGRAMAS_TECNOLOGÍA

d. Variable dependiente: COMPRAS_INTERNET_ECOMMERCE

Fuente: Diseño propio mediante software SPSS v.25

siendo cada día funcionales al estar conectados de diferentes maneras enviando información y el estar procesándola a través de la nube (Biggs et al., 2015).

Para los sistemas digitales de pagos se están utilizando las tarjetas de crédito y débito, estas son rápidas para hacer transacciones, pues reducen costos y tiempos; para los pagos en línea y permiten mayor acceso a cualquier tipo empresa (Humphrey et al., 2003).

Por otro lado, el comercio en pequeña escala ha tenido cambios inesperados para el empresario

Tabla 6. ANOVA[a]

Modelo		Suma de cuadrados	gl	Media cuadrática	F	Sig.
1	Regresión	6793.521	1	6793.521	237.649	.000b
	Residual	15665.331	548	28.586		
	Total	22458.853	549			
2	Regresión	8619.322	2	4309.661	170.337	.000c
	Residual	13839.531	547	25.301		
	Total	22458.853	549			
3	Regresión	8890.476	3	2963.492	119.253	.000d
	Residual	13568.377	546	24.851		
	Total	22458.853	549			

a. Variable dependiente: COMPRAS_INTERNET_ECOMMERCE

b. Variables predictoras: (Constante), USO_TECNOLOGÍA_INTERNET

c. Variables predictoras: (Constante), USO_TECNOLOGÍA_INTERNET, HABILIDAD_INTERNET

d. Variables predictoras: (Constante), USO_TECNOLOGÍA_INTERNET, HABILIDAD_INTERNET, CONSUMO_PROGRAMAS_TECNOLOGÍA

Fuente: Diseño propio mediante software SPSS v.25

y el consumidor, el vínculo que los une para el nuevo mercado ha sido el ciberespacio (Bocanegra Gastelum y Vázquez Ruiz, 2021).

Así con estos cambios comerciales, los establecimientos han ofrecido ventas las 24 horas del día y en todos los meses del año, prueba de ello, es que se examinan las características de compra y consumo de manera personalizada y digital con métodos de pago con una simple tarjeta electrónica con acceso al internet desde la casa hasta la oficina con un aparato móvil, smart phone, tablet, computadora, Ipad entre otras (Tham et al., 2019).

México ha estado en crecimiento continuo y apegado a las leyes de acuerdo al comercio por internet para proteger a los consumidores de acuerdo al capítulo 19 del T-MEC, Asociación Latinoamericana de Internet (Ríos, 2021).

Estudios demuestran que, se ha visto una tendencia de las compras por internet mediante el comercio electrónico (e-commerce) a nivel minorista con un 12%, en su mayoría de la población de jóvenes con educación, en conocimientos y del manejo de las TIC´s.; también por las reiteradas compras individuales con diferentes empresas nacionales y extrajeras ejemplo de ellas, destacan: Amazon, Mercado Libre, Walmart, Office Depot, Home Depot, eBay, Expedia, etc. (Bocanegra Gastelum y Vázquez Ruiz, 2021).

De acuerdo a Díaz y Valencia (2015) se realizó un estudio de la oferta de comercio electrónico en un conjunto de micro y pequeñas empresas (MiyPes) ubicados en diferentes distritos dentro de la ciudad de Lima, Perú, con el objetivo de identificar la realidad de la oferta del e-commerce de esa región (Cáceda Salazar, 2014).

En España se encontró que, en el 2020, se realizó un estudio sobre social network donde el 87% de los internautas utilizan las redes sociales en edades de 16 a 65 años para hacer sus compras on line y se encontró que hubo disminuciones en cuanto a marcas y actividades publicitarias del 81% en el 2018 y un 52% en el 2020, lo que se observa que existe un mayor consumo masivo de las aplicaciones sociales que en las propias ventas tradicionales y dejando atrás el valor de las marcas (De-Frutos-Torres et al., 2021).

3. Materiales y métodos

Para esta investigación, primeramente, se diseñó un instrumento con 36 ítems con escalas tipo Likert, que fue validado por expertos en el tema de marketing, para su confiabilidad se empleó la herramienta de Alfa de Cronbach, posteriormente se aplicaron las encuestas través de Google forms, para ello se contó con una muestra finita de 550 elementos con un nivel de

Tabla 7. Coeficientes[a]

Modelo		Coeficientes no estandarizados		Coeficientes tipificados	t	Sig.	Intervalo de confianza de 95.0% para B		Estadísticos de colinealidad	
		B	Error típ.	Beta			Límite inferior	Límite superior	Tolerancia	FIV
1	(Constante)	21.639	.544		39.767	.000	20.570	22.708		
	USO_TECNOLOGÍA_INTERNET	3.289	.213	.550	15.416	.000	2.870	3.708	1.000	1.000
2	(Constante)	17.881	.677		26.426	.000	16.552	19.210		
	USO_TECNOLOGÍA_INTERNET	2.511	.221	.420	11.381	.000	2.077	2.944	.828	1.208
	HABILIDAD_INTERNET	1.734	.204	.313	8.495	.000	1.333	2.134	.828	1.208
3	(Constante)	14.887	1.127		13.205	.000	12.673	17.102		
	USO_TECNOLOGÍA_INTERNET	2.492	.219	.417	11.393	.000	2.062	2.921	.827	1.209
	HABILIDAD_INTERNET	1.693	.203	.306	8.358	.000	1.295	2.091	.825	1.213
	CONSUMO_PROGRAMAS_TECNOLOGÍA	2.096	.635	.110	3.303	.001	.850	3.343	.993	1.007

a. Variable dependiente: COMPRAS_INTERNET_ECOMMERCE

Fuente: Diseño propio mediante software SPSS v.25

confianza del 95% y un margen de error de 5%. El marco de muestreo está enfocada principalmente a empresarios en pequeña escala, lo que significa que tienen menos de 30 empleados, y están clasificados como Micro y pequeñas empresas (MiyPes) en México.

Para ello, se utilizó el método cuantitativo de tipo descriptivo y correlacional, utilizando estadísticos descriptivos e inferenciales mediante la herramienta de regresión lineal con el Modelo de pasos sucesivos, utilizando el software SPSS versión 25.

Para conocer el comportamiento de la información de acuerdo a los criterios expuestos sobre el e-commerce como lo afirma Josept Schumpeter en (Croitoru, 2012), (Ver Figura I).

4. Resultados y discusión

A continuación, se presentan los estadísticos de tipo descriptivo e inferencial que se generaron en la investigación de acuerdo al instrumento aplicado por *Google Forms*, y capturados en el programa *SPSPS v.25*, se diseñó la variable independiente y las variables dependientes que influyen en las compras por internet (e-commerce), a continuación, se explica lo siguiente:

4.1. Estadísticos Descriptivos

En este apartado, se analizan cada una de los elementos, que se encuentra en interacción con las compras por internet; así como las variables que se correlacionarán. Para validar los *ítems*, de acuerdo al instrumento de 36 preguntas, con 550 casos, se utilizó el programa *SPSS v.25*, utilizando la herramienta de *Alfa de Crobach,* con un .861, lo que significa que, se encuentra por arriba de lo normal estipulado por (Hernández-Samipieri et al., 2014) que considera debe ser mayor a .7, lo que muestra una fiabilidad adecuada de acuerdo al instrumento planteado (Ver tabla 1).

En tabla 2. Se muestra los descriptivos, con las variables que resultaron seleccionadas, aquí se

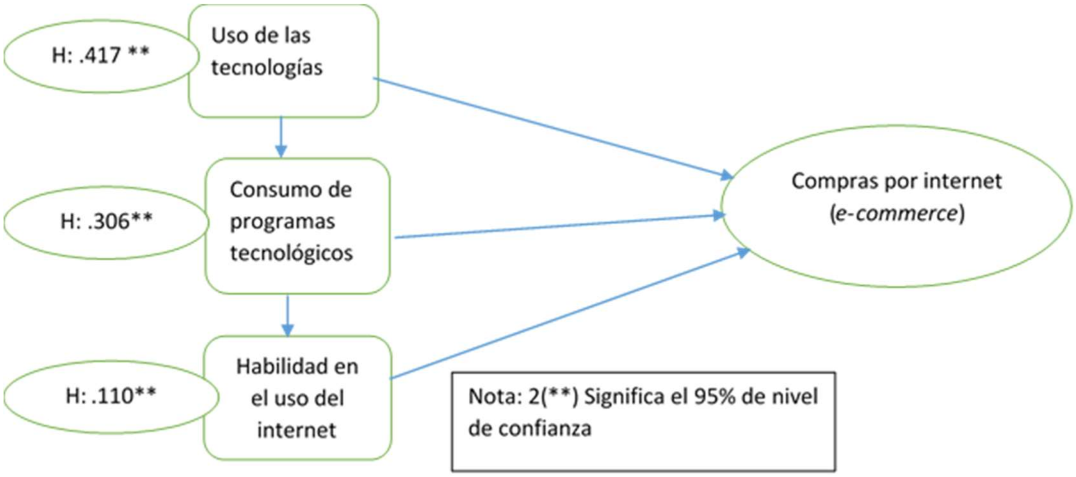

H: .417 **
Uso de las tecnologías
H: .306**
Consumo de programas tecnológicos
H: .110**
Habilidad en el uso del internet
Compras por internet (*e-commerce*)
Nota: 2(**) Significa el 95% de nivel de confianza

Fig. 2. Elaboración propia del Modelo teórico de variables involucradas

calculó el valor de la media de los datos. Donde se obtuvo que las Compras por internet dio como resultado de 29.2564 como variable independiente, el uso de la tecnología con internet con un 2.3161, consumo de Programas tecnológicos de 1.5106 y la habilidad del internet fue de 3.2076, éstas últimas como variables dependientes:

$\bar{X}$ = 29.25

Valor de la media es $\overline{X1}$ = 2.3161, $\bar{X}2$=1.5106, $\overline{X3}$ = 3.2076

4.2. Estadístico inferencial

Para medir la relación de los datos, se utilizó la correlación de *Pearson*, los valores deben ser positivos dado que deben estar dentro del rango de confiabilidad del 95% el $p \leq [.05]$ es significativo. Dentro de las correlaciones de Pearson que se realizaron, se determina la relación que existe entre las tres variables más importantes, y a su vez se determinan los resultados más exactos entre cada variable.

Los datos arrojados muestran correlaciones inferiores a 1, esto es, existe una correlación positiva directa entre las variables estudiadas (Ver tabla 3), las cuales resultaron moderadamente significativas, desde el punto de vista estadístico, para la regresión lineal.

Los factores que están correlacionados con las Compras de internet fueron: Uso de la tecnología con .550, consumo de los programas tecnológicos .157 y habilidad sobre el internet de .488. Se observa que la habilidad del internet y las compras en internet (*e-commerce*) se asocian moderadamente entre sí.

Por consiguiente, se desarrolló el Modelo de pasos sucesivos para seleccionar las variables introducidas. Las que se incluyeron en el modelo son: uso de la tecnología con internet, la habilidad del internet y el consumo de programas tecnológicos. (Ver tabla 4).

Para el resumen del modelo (Tabla 5), la importancia que tuvo la variable, es aquella que resultó mayor que el resto, en esta, se generaron tres modelos de los cuales solo el Modelo 3 con R corregida dio como resultado de .393 y las variables predictoras del modelo R^c representado con un 629% de confiabilidad en los datos.

Para el ANOVA del modelo 3 la prueba de F es de 119.253 y la significancia fue de .000, lo que estadísticamente es significativo.

Para la comprobación de la hipótesis en la tabla 7, se consideraron los Coeficientes tipificados esto, para comprobar las hipótesis con los valores de *Beta*, en el Modelo 3: *Beta*1 Uso de la tecnología .417 es significativo al **95%, por lo tanto, se aceptan las hipótesis, *Beta* 2 Habilidad en el internet .306 es significativo al **95%, lo cual se aceptan las hipótesis y *Beta 3* Consumo de programas tecnológicos .110 es significativo al **95%, por lo tanto, se aceptan las hipótesis.

En los valores de *t* resultaron para el uso de la tecnología de 11.393, habilidad en el internet es de 8.358 y en consumo de programas tecnológicos es de 3.303. La significancia es de .000 y .001. Las otras hipótesis quedaron rechazadas.

Finalmente, se muestra un diagrama simplificado (*Figura II*), de acuerdo al Modelo teórico, donde se manifiestan las variables independientes y dependiente; así como la comprobación que nuestras hipótesis que fueron aceptadas con un nivel de confianza del 95% y correlacionadas entre sí, dando un resultado favorable.

Por lo tanto, se comprueba nuestra ecuación matemática de regresión lineal, con los Coeficientes no estandarizados con los valores de *Beta* que fueron los siguientes: la Constante 0 vale 14.88, Uso de tecnología β1 = 2.492, habilidad del internet β 2 = 1.693 y los valores de consumo de programas tecnológicos fue de β3= 2.096. Entonces nuestra ecuación matemática queda comprobada con:

$$Y = \beta 0 + \beta 1(X1) + \beta 2(X2) + \beta 3(X3) =$$
$$Y = 14.88 + 2.423(23.161) + 1.693(1.5106) + 2.096\ (3.2076) = 29.25$$

5. Conclusión

Con los nuevos desafíos de las tecnologías digitales que han llegado a México. Las nuevas tecnologías traen cambios significativos para los usuarios en pequeña escala, que experimentan a través del internet, los diferentes mecanismos electrónicos como el comercio electrónico (*e-commerce*), a su vez las plataformas virtuales, los programas tecnológicos que han tenido interacciones con las compras y ventas de productos y servicios de primera mano, donde los empresarios de las MiyPes, han sabido aprovechar como estrategia de mercadotecnia el gestionar sus compras y abastecer sus inventarios en poco tiempo; así como atraer nuevos clientes potenciales y poder definir un perfil de compra.

Los hallazgos encontrados permitieron definir el perfil de compra de los empresarios en pequeña escala, sustentándose en el modelo planteado anteriormente, de acuerdo a los constructos formulados, dieron como resultado que los empresarios utilizan las tecnologías en un 55%, el consumo de los programas tecnológicos 15% y la habilidad en internet con un 48%; las variables mostraron estar moderadamente correlacionadas entre sí, con las compras por internet (*e-commerce*), con un 62%, lo que significa que los usuarios que cuentan con negocios o empresas pequeñas, hacen un esfuerzo por adentrarse a las nuevas tendencias tecnológicas, algunos con experiencia propia, otros por capacitaciones recibidas dentro de su propia empresa o con algún familiar, por lo que queda comprobado que estas herramientas son las más requeridas para las transacciones comerciales.

Referencias

1. **Angenu, B. B., Quansah, F., Okoe, A. F. (2015).** Determinants of online banking adoption among Ghanaian University Students. Journal of Service Science and Management, Vol. 8, No. 2, pp. 183–190. DOI: 10.4236/jssm.2015.82020.
2. **Biggs, P., Garrity, J., LaSalle, C., Polomska, A. (2015).** smartnet.niua.org/ sites/default/files/resources/Harnessing-IoT-Global-Development.pdf
3. **Bocanegra, C., Vázquez, M. (2021).** México y China: Comercio minorista electrónico y perfil del consumidor. Revista Chilena de Economía y Sociedad, Vol. 15, No. 1, pp. 56–77.
4. **Cáceda-Salazar, H. (2014).** Los obstáculos del eCommerce en Perú y estrategias para atraer clientes a mi tienda virtual. Comisión de Promoción del Perú para la Exportación y el Turismo.
5. **Croitoru, A. (2012).** Schumpeter, J.A., 1934 (2008), The Theory of Economic Development: An inquiry into profits, capital, credit, interest and the business cycle. Journal of Comparative Research in Anthropology and Sociology, Vol. 3, No. 2, pp. 137–148.
6. **De-Frutos-Torres, B., Pastor-Rodríguez, A., Martín-García, N. (2021).** Consumo de las plataformas sociales en internet y escepticismo a la publicidad. El Profesional de La Información, Vol. 30, No. 2, pp. 1–11. DOI: 10.3145/epi.2021.mar.04.

7. **Dhanapal, S., Vashu, D., Subramaniam, T. (2015).** Perceptions on the challenges of online purchasing: A study from "baby boomers", generation "X" and generation "Y" point of views. Contaduria y Administracion, Vol. 60, pp. 107–132. DOI: 10.1016/j.cya.2015.08.003.

8. **Díaz, D., Valencia, B. (2015).** Estudio exploratorio de la oferta de comercio electrónico en un conjunto de micro y pequeñas empresas (Mypes) localizadas en diversos distritos de Lima Metropolitana. Pontificia Universidad Católica del Perú. http://tesis.pucp.edu.pe/repositorio/handle/123456789/6769.

9. **Espinoza-Delgado, J., Puente-Valero, V., Flores-Rueda, I., Tristán-Monrroy, B. (2020).** Percepción de estudiantes sobre el riesgo en compras por internet. SUMMA. Revista disciplinaria en ciencias económicas y sociales, Vol. 2, No. 1, pp. 83–103.

10. **Hernández-Sampieri, R., Fernández-Collado, C., Baptista-Lucio, P. (2014).** Selección de la muestra. Metodología de la investigación: 6ta edición, pp. 170–196.

11. **Humphrey, D., Willesson, M., Lindblom, T., Bergendahl, G. (2003).** What does it cost to make a payment? Review of Network Economics, Vol. 2, No. 2, pp. 159–174. DOI: 10.2202/1446-9022.1024.

12. **NACIONES UNIDAS. (2017).** Informe sobre la economía de la información 2017 digitalización, comercio y desarrollo. Conferencia de las Naciones Unidas sobre comercio y desarrollo, UNCTAD. https://unctad.org/es/system/files/official-document/ier2017_es.pdf.

13. **Palomino-Pita, A. F., Mendoza-Vargas, C., Oblitas-Cruz, J. F. (2020).** E-commerce and its importance in times of covid-19 in Northern Peru. Universidad Privada del Norte. Vol. 25, No. 3, pp. 253–266. DOI: 10.37960/rvg.v25i3.33367.

14. **PROFECO. (2016).** Capítulo 14 comercio electrónico. Procuraduría Federal del Consumidor. https://www.gob.mx/cms/uploads/attachment/file/86482/14._Comercio_Electr_nico.pdf.

15. **Ramos-Carrasco, A. M., Altamirano-Morra, P. (2021).** La confianza del consumidor y el comercio electrónico en Lima Metropolitana años 2015-2020. Universidad San Ignacio de Loyola.

16. **Ríos-Ruiz, A. Á. (2021).** Capítulo 19 del T–MEC: Implicaciones para el comercio electrónico en México: Alma de los Ángeles Ríos Ruiz. Perfiles de las Ciencias Sociales, Vol. 8, No. 16.

17. **Riquelme, R. (2018).** Comercio electrónico en México. El Economista. https://www.eleconomista.com.mx/tecnologia/Comercio-electronico-en-Mexico-desacelera-crecimiento-porsegundo-ano-consecutivo-20181205-0090.html

18. **Rodríguez, K., Ortiz, O., Quiroz, A., Parrales, M. (2020).** El e-commerce y las Mipymes en tiempos de Covid-19. Espacios, Vol. 41, No. 42, pp. 100–118. DOI: 10.48082/espacios-a20v41n42p09.

19. **Sánchez-Alzate, J. A., Montoya-Restrepo, L. A. (2016).** Factors affecting the consumer trust for shopping through electronic media. Revista Científica Pensamiento y Gestión, Vol. 40, pp. 159–186. DOI: 10.14482/pege.40.8809.

20. **Sanz, S., Ruiz, C., Pérez, I. (2018).** Factores determinantes de las relaciones de intercambio comercial en México. Un estudio centrado en la compra en línea. Administración y Organizaciones. Universidad Autónoma Metropolitana, Vol. 21, No. 41, 75–90. DOI: 10.24275/uam/xoc/dcsh/rayo/2018v21n41/Sanz.

21. **Tham, K., Dastane, O., Johari, Z., Ismail, N. (2019).** Perceived risk factors affecting consumers' online shopping behaviour. The Journal of Asian Finance, Economics and Business, Vol. 6, No. 4, pp. 246–260. DOI: 10.13106/jafeb.2019.vol6.no4.249.

22. **Viramontes-Olivas, O., Hernández-Perea, J. J., Flores-Morales, C. R. (2015).** Factores que influyen en el rendimiento académico de los estudiantes de la FCA-UACH que trabajan. XVII Congreso internacional sobre innovaciones en docencia e investigación en ciencias económico administrativas, pp. 1–5.

Article received on 10/01/2023; accepted on 13/11/2023.
[*]Corresponding author is Lidia Ramírez-Lemus.

ISSN 2007-9737

The Impact of Training Methods on the Development of Pre-Trained Language Models

Diego Uribe*, Enrique Cuan, Elisa Urquizo

Tecnológico Nacional de México,
Instituto Tecnológico de La Laguna,
Mexico

{duribea, ecuand, eurquizob}@lalaguna.tecnm.mx

Abstract. The focus of this work is to analyze the implications of pre-training tasks in the development of language models for learning linguistic representations. In particular, we study three pre-trained BERT models and their corresponding unsupervised training tasks (e.g., MLM, Distillation, etc.). To consider similarities and differences, we fine-tune these language representation models on the classification task of four different categories of short answer responses. This fine-tuning process is implemented with two different neural architectures: with just one additional output layer and with a multilayer perceptron. In this way, we enrich the comparison of the pre-trained BERT models from three perspectives: the pre-training tasks in the development of language models, the fine-tuning process with different neural architectures, and the computational cost demanded on the classification of short answer responses.

Keywords. Language models, pre-training tasks, BERT, fine-tuning.

1 Introduction

Currently, the development and deployment of Large Language Models (LLMs) is a common scenario in the sphere of NLP due to the development paradigm known as Self-Supervised Learning (SSL). This learning paradigm, also known as a process of two steps: pre-training and fine-tuning, outlines a generic framework for transferring knowledge [18, 2].

While pre-training a LLM produces semantic representations by processing unlabeled data, fine-tuning makes use of such representations for a particular downstream learning task.

In this way, the performance of this new task depends significantly on the quality of the semantic representations, which in turn depend on the quality of the training methods for the development of a LLM. Thus, how to produce good quality representations? We focus our attention on the analysis of the training methods for producing semantic representations to be transferred to make the definition of a learning model, for a particular downstream language task, a non-complex issue.

As a result of research on representation learning, a semantic vector known as embedding is nowadays the building block for a wide range of NLP tasks.

Since this semantic vector denotes a point in high-dimensional space, modeling similarity between words is straightforward. Two main types of word embeddings have been developed: static and contextual embeddings. Static embeddings are also known as context independent embeddings, as such representations are unique for each word and ignore the word's context.

Glove [15] and word2vec [14] are classic examples of this kind of embeddings. On the other hand, contextual embeddings are also known as context-dependent embeddings, as each word is represented by a different vector for each context in which it is used.

In other words, contextual embeddings allow us to represent multiple senses of a particular word. ELMo [16] and BERT [4] are examples of contextual embeddings. The mechanism for acquiring these embeddings is known as pre-training, a process defined as the computation

doi: 10.13053/CyS-28-1-4718

of large document collections in order to learn the semantic vectors corresponding to words or sentences. Actually, pre-trained language models denote the mechanism for acquiring these semantic representations that have been developed by using two deep learning architectures: recurrent neural networks (RNNs) and transformer networks.

ELMo is an example of a pre-trained language model based on RNNs, whereas BERT is a classic example of a pre-trained language model based on transformer networks. In this work, we examine BERT, a pre-trained language model based on a bidirectional transformer encoder which is characterized by a bidirectional self-attention mechanism to produce contextual embeddings.

There are many BERT models, all variants on the original BERT, available to perform some downstream task like classification or tagging. From the model collection available at TensorFlow Hub [22], we analyze three BERT models: the original, the universal, and the compact BERT model.

In terms of representation learning, what makes one BERT model better than another? Is there any significant difference in the quality of the contextual embeddings between these three BERT models? To answer these questions, we first analyze the pre-training process of a bidirectional language model as BERT, and then the fine-tuning process to transfer the embedded knowledge to a downstream language task.

2 Motivation Behind the Work

The guide to conducting our study is clearly defined with the following research question: what is the impact of the training methods for each BERT model on the quality of the linguistic representations produced by these models?

Thus, the motivation behind the training methods for each BERT model is to perceive the similarities and differences between the various training techniques to produce semantic knowledge to be embedded via fine-tuning.

Since BERT is a bidirectional encoder, and thus it is able to attend to the whole context of a particular input element (left and right of the current input), the training method is based on a cloze task [21]. Masked Language Model (MLM) is the original unsupervised training method where the model learns to predict the missing words of a text. By learning to predict the masked words, the model produces suitable word-level representations.

Another unsupervised task for the training of BERT is to deal with the relationship between pairs of sentences. Next Sentence Prediction (NSP) is an unsupervised training method where the model learns to predict such connection between pairs of sentences. Now, the pre-training method for the universal BERT model is a bit different.

The purpose is to improve the semantic representations at sentence level by implementing a dual encoder based on the combination of the BERT original training methods: the integration of NSP with MLM training is denominated by its authors as the Conditional Masked Language Model (CMLM) [28].

The third language model studied in this work is the compact BERT model. As LLMs have a high computational cost, this small model was created with the purpose of not only reducing the computational cost but also using the same self-supervised learning paradigm in its development [24].

We then conduct an empirical evaluation via fine-tuning to transfer the embedded knowledge to a downstream language task as classification. The representations obtained from the BERT models are transferred to a classifier model, commonly represented as a simple multiperceptron, to be fine-tuned to the peculiarities of a downstream task as short answer responses classification.

The collection of short answer responses was created with the intention of automated assessment of written responses [3]. Each instance in the collection denotes a short answer corresponding to a particular story of a specific domain where the grade is defined in terms of levels of quality.

In other words, the fine-tuning process performs a downstream task as multi-class classification where a short answer is assigned into one of the multiple rubrics of the responses. Thus, we have described the perspective from which a learning paradigm known as Self-Supervised Learning is analyzed. The primary contributions of our work are summarised as follows:

- To provide insights about the impact of training methods in the development of pre-trained models. The pre-training process for each BERT model is described to consider similarities and differences between them.
- To conduct an empirical evaluation on semantic linguistic representations. The fine tuning process is implemented on a downstream classification task with a learning model defined in terms of the semantic representations produced by each BERT model.
- To offer additional insight into the computational resources demanded by the language models. The experimentation carried out allows us to detail the computational cost incurred by each BERT model.

3 BERT Pre-training and Language Models

We describe in this section the language models with which BERT has been trained for learning meaning representations for words and sentences: MLM [4], NSP [4], CMLM [28] and Distillation [24]. But we first briefly take a look at BERT and its self-attention mechanism that has impacted the world of NLP.

3.1 BERT: Bidirectional Encoder Representations from Transformers

In its broadest sense, the transformer consists of an encoder-decoder architecture. However, BERT is a transformer model that includes only the encoder component.

Unlike other popular embedding models (e.g., word2vec) that produce static embeddings irrespective of the context, BERT generates dynamic embeddings based on the context so multiple embeddings are produced for the multiple contexts in which a particular word can be used [4]. In order to generate context-based embeddings, the attention mechanism of the transformer plays a crucial role in the encoding process.

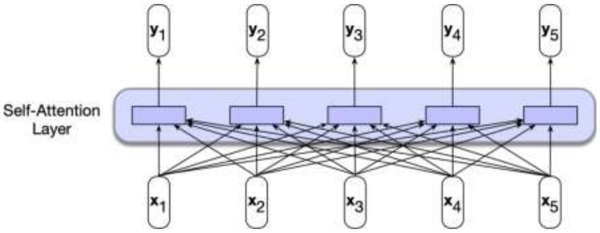


Fig. 1. Bidirectional self-attention model. This figure corresponds to [10]

Self-attention, a special type of attention, emerged as a more efficient alternative to overcome the limitations of the RNNs: capturing long-term dependencies is one of the major challenges with RNNs [25].

Self-attention takes a holistic approach to the analysis of the linguistic elements: instead of considering only the previous elements in the input, self-attention compares each element with all the sequence elements in order to understand how words relate to each other over long distances.

Given a sequence of input elements $(x_1, \ldots, x_n)$, Figure 1 shows how the output of a particular element y_i depends on the comparisons between the input x_i and the preceding and following elements x_j.

In other words, the self-attention mechanism is responsible for considering each element of the entire input sequence and mapping them to contextualized output vectors. A formal description of the output values (vector y) is based on three concepts:

- **Query**: The current focus of attention.
- **Key**: Preceding and following input to be compared with the current focus of attention.
- **Value**: Computation of the output for the current focus of attention.

In this way, each element of the input vector **x** is represented in terms of these concepts and the corresponding weights:

$$\begin{aligned} q_i &= W^Q x_i, \\ k_i &= W^K x_i, \\ v_i &= W^V x_i. \end{aligned} \tag{1}$$

Then, the output y_i corresponding to each input element x_i is:

$$y_i = \sum_{j=i}^{n} \alpha_{ij} v_j, \tag{2}$$

where the alpha weights represent the proportional relevance of each input to the current focus of attention:

$$\alpha_{ij} = \frac{\exp(\text{score}_{ij})}{\sum_{k=1}^{n} \exp(\text{score}_{ik})}, \tag{3}$$

$$\text{score}_{ij} = q_i\, k_j. \tag{4}$$

Thus the comparison of each element with the rest of the sequence elements take place in parallel. This means simultaneous access to all sequence elements and therefore simultaneous computation of the relevance of each sequence element. In this way, the step-by-step processing of intermediate recurrent connections is eliminated.

3.2 BERT Training Techniques

We describe in this section the language models with which BERT has been trained for learning meaning representations for words and sentences: MLM [4], NSP [4], CMLM [28] and Distillation [24].

3.2.1 Masked Language Modeling (MLM)

Masked Language Modeling is the approach to training a deep bidirectional transformer as BERT to learn contextual word-level representations [4].

MLM is basically a cloze task [21]: some percentage of the input tokens are masked in a random way, in order to figure out those masked tokens. More precisely, each token of the sequence can be:

- masked
- replaced with another token from the vocabulary
- left unchanged

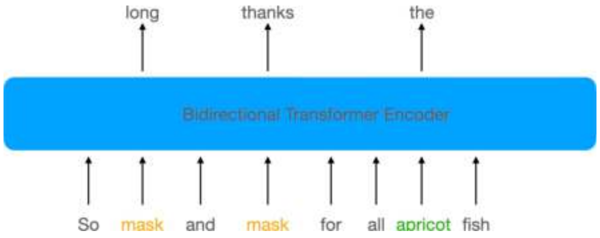


Fig. 2. Masked Language Model training

Figure 2 shows this training task. In this example, three of the input tokens are selected, two of which are masked (long and thanks) and the third (the) is replaced with a tangential token from the vocabulary.

The purpose is to predict the original words for each of the masked tokens as well as the tangential token and in this way to reproduce the original input sequence. MLM is an unsupervised learning method as a large corpus of unannotated input sequences is used for training.

The output vector for each of the masked tokens (h_i) is multiplied by a learned set of classification weights W_v in order to take a softmax to produce a probability distribution over the vocabulary:

$$y_i = \text{softmax}(W_v\, h_i). \tag{5}$$

3.2.2 Next Sentence Prediction (NSP)

Next Sentence Prediction (NSP) is another unsupervised task for the training of BERT on how to deal with the relationship between pairs of sentences [4].

As many applications such as paraphrase detection or entailment demand determining how close or distant two sentences are, NSP is an unsupervised training method where the model learns to predict such connection between pairs of sentences.

In the particular case of BERT, 50% of the training pairs denote adjacent sentences whereas the other 50% of the pairs denote unrelated sentences as the second sentence is randomly selected. In addition to the input elements of the sentences, two new tokens are added to conduct a proper training: the token [CLS] is prepended to the input sentence pair, and the token [SEP] is

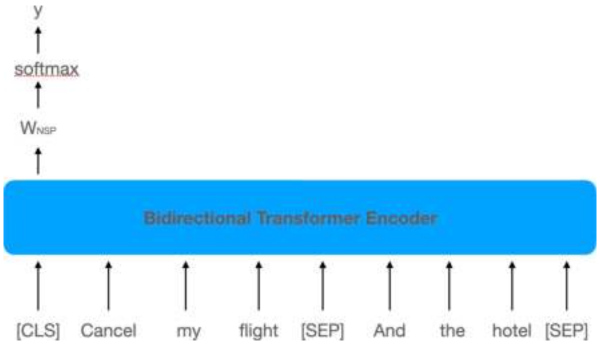


Fig. 3. Next Sentence Prediction training

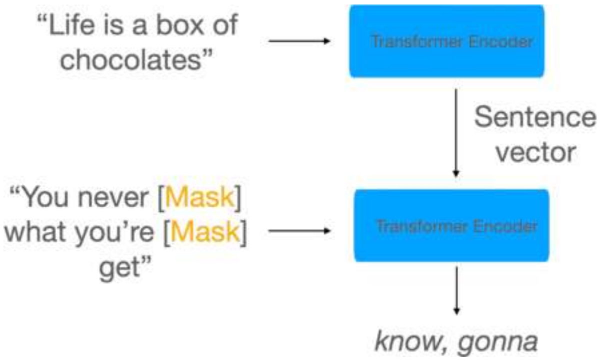

"Life is a box of chocolates"
Transformer Encoder
Sentence vector
"You never [Mask] what you're [Mask] get"
Transformer Encoder
know, gonna

Fig. 4. Conditional MLM training

placed between the sentences and after the final token of the second sentence. Figure 3 shows this training task. While the role of the token [SEP] is obvious, the token [CLS] represents the output vector associated with the final layer of the transformer.

And it is precisely this output vector that denotes the next sentence prediction. The output vector for each training pair (h_i) is multiplied by a learned set of classification weights W_{NSP} in order to take a softmax to produce a two-class prediction:

$$y_i = \mathrm{softmax}(W_{\mathrm{NSP}}\, h_i). \tag{6}$$

This NSP task was inspired by the framework developed by Logeswaran and Lee for learning sentence representations from unlabeled data [13]. The key point of their work was the replacement of a generation objective, that is, the generation of a context sentence given an input sentence.

Instead, they replace the decoder with a classifier to predict the target sentence from a set of candidate sentences. In this way, the NSP training task takes advantage of this antecedent work to allow BERT to be able to produce sentence-level representations.

3.2.3 Conditional Masked Language Modeling (CMLM)

Conditional Masked Language Modeling is an alternative approach to training a deep bidirectional transformer as BERT to learn effective sentence-level representations [28]. Basically, CMLM is a training method that combines two training tasks: Next Sentence Prediction (NSP) and MLM.

The main idea of CMLM is learning sentence representations by optimizing the performance on the MLM task. The architecture of CMLM is based on the use of two transformer encoders and the processing of pair of sentences such as the NSP method does. From each pair of sentences, the first sentence becomes the input into an encoder that produces a sentence vector.

This sentence representation is then provided to the second encoder to perform the MLM task on the second sentence by making use of the learning weights generated by the first encoder to produce the sentence representation. Since the sentence vector is projected into N spaces, the MLM of the second sentence can result from observing more than one representation.

In this way, the optimization of the MLM task depends on the sentence vector representation of the adjacent sentence. Last but not least, this dependency of the MLM task on the sentence vector representation of the adjacent sentence is the reason to include the word "conditional" in the name of this language model: Conditional Masked Language Model. Figure 4 shows the architecture of this training task.

This CMLM training task was inspired by the Skip-Thought work developed by Kiros et al. for learning generic sentence representations from a large training corpus of contiguous text [11]. The key point of their work was the replacement of composition operators based on the mapping of word embedding to sentence representations. Instead, they replace the composition operator with a sentence encoder to encode a sentence to predict the sentences around it: the previous and the next sentence. In this way, the CMLM training task takes advantage of this antecedent work to allow BERT to be able to improve sentence-level representations.

3.2.4 Knowledge Distillation

Building a compact model revolves around knowledge distillation: the standard technique for model compression [8].

Since LLMs have a high computational cost, research on the development of a small model was guided by not only reducing the computational cost but also by using the same self-supervised learning paradigm in its development.

Indeed, building a compact model proved to be possible by applying the standard pre-training and fine-tuning process but a different training strategy, based on a compression technique known as knowledge distillation, was implemented.

Basically, this distillation technique consists of a student-teacher training method where the teacher, a robust LM, transfers knowledge to the student, a small LM to be developed, through its predictions for unlabeled training examples.

Figure 5 shows the knowledge distillation process incorporated in the development and implementation of a compact BERT model [24]. The training resources demanded by the process are the following:

- **Teacher**: The teacher is a LLM which can be either a BERT-base or a BERT-large pre-trained language model.
- **Student**: The student is the compact model to be built. Whereas the total number of parameters is 110 million in BERT-base, the initial size for a tiny model is 4 million parameters.
- **Label data (D_L)**: A set of N training examples $(x_1, y_1), \ldots, (x_N, y_N)$, where x_i is an input and y_i is a label.

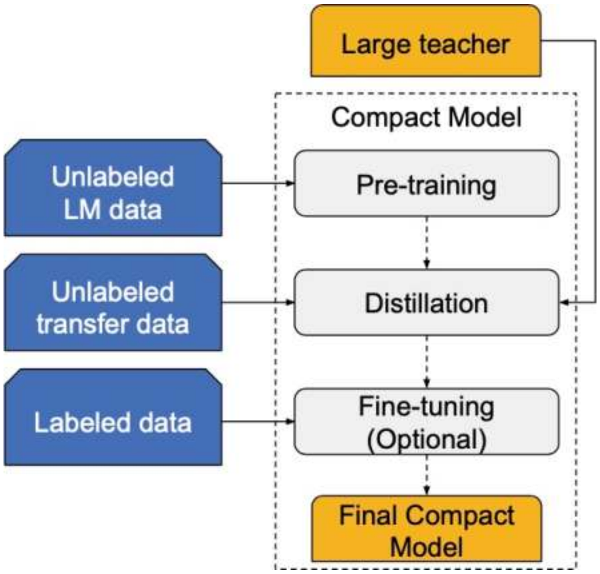


Fig. 5. Knowledge Distillation process. This figure corresponds to [24]

- **Unlabeled training data (D_T)**: A set of M input examples $x_1^{'}, \ldots, x_M^{'}$ obtained from a distribution not necessarily identical to the distribution of the labeled set.

  This dataset is used by the teacher for the transfer of knowledge to the student by making available its predictions for instances $x_m^{'}$.
- **Unlabeled language model data (D_{LM})**: it is an unannotated text collection for unsupervised learning of text representation by using MLM as training method. And a procedure for a sequence of three training operations executed by the algorithm (Figure 1).
- **Pre-training on D_{LM}**: pre-training of the compact model with MLM as training method (Line 1).
- **Distillation on D_T**: transfer knowledge to the student. Once the student is prepared, the teacher transfer its knowledge to the student via its predictions to strengthen the compact model.

  Line 3 shows the estimation of the cross-entropy loss between teacher and student predictions, this loss is then used to update the student model. (Line 4).

Table 1. BERT models and unsupervised training methods

BERT Model	MLM	NSP	CMLM	Distillation
Original	×	×		
Universal	×	×	×	
Compact	×			×

Algorithm 1 Knowledge Distillation algorithm. This figure corresponds to [24]

```
Require: student $\theta$, teacher $\Omega$, unlabeled LM, data $\mathcal{D}_{LM}$, unlabeled transfer data $\mathcal{D}_T$, labeled data $\mathcal{D}_L$
1: Initialize $\theta$ by pre-training and MLM$^+$ on $\mathcal{D}_{LM}$
2: for each $x \in \mathcal{D}_T$ do
3:     Get loss $L \leftarrow -\sum_y P_\Omega(y|x) \log P_\theta(y|x)$
4:     Update student $\theta \leftarrow \mathrm{BACKPROP}(L, \theta)$
5: end for
6: Fine-tune $\theta$ on $\mathcal{D}_L$                    ▷ Optional step.
7: return $\theta$
```

- **Fine-tuning on** D_L: Line 6 shows this optional step. The compact model is fine-tuned on end-task labeled data. In other words, the similarity between the distribution of the transfer and labeled datasets is perceived in this step.

This compact model is compared with two contemporary works that also use distillation for transfer knowledge. Both works initialize the student with a BERT model truncated, that is, the bottom layers of a 12-layer BERT model are used for the initialization of the student.

However, the distillation process is different. Whereas Patient Knowledge Distillation performs task-specific distillation [20], DistillBert makes use of a more expensive LM teacher as distillation is performed on general-domain data [19].

3.3 BERT Models

As we previously said, the motivation behind this work is to study three pre-trained BERT models and their corresponding unsupervised training tasks.

The previous section describes each unsupervised training task and Table 1 shows similarities and differences between the BERT models in terms of the training methods used in their development.

As we see in Table 1, MLM and NSP are unsupervised training tasks that characterize the development of the **O**riginal BERT model [5]. This model[1] consists of $L = 12$ encoder layers, a hidden size of $H = 768$, and $A = 12$ attention heads representing a total of 110M parameters.

On the other hand, the development of the **U**niversal BERT model is based on CMLM, an unsupervised training task that integrates MLM and NSP in order to optimize the semantic representations at sentence-level [27].

This model[2], that extends the BERT transformer architecture, maps text into high dimensional vectors to capture sentence-level semantics. Last but no least, we have a very different trained model:

The **C**ompact BERT model based on an initial model trained on MLM (the student) to eventually improve its performance by knowledge distillation from the teacher [23].

This model[3] consists of $L = 4$ encoder layers, a hidden size of $H = 512$, and $A = 8$ attention heads representing a total of 28M parameters.

4 Experimental Evaluation

Once we have described the training methods for each BERT model, we want to know its behavior on a particular text-processing task. So, the experimentation conducted is detailed in this section.

First, we explain the fine-tuning process of the pre-trained language models previously mentioned to perform a downstream task as sequence classification. Then, the dataset characteristics are exposed and the results of each BERT model are exhibited.

[1] bert_en_uncased_L-12_H-768_A-12

[2] universal-sentence-encoder-cmlm/multilingual-base

[3] small_bert/bert_en_uncased_L-4_H-512_A-8

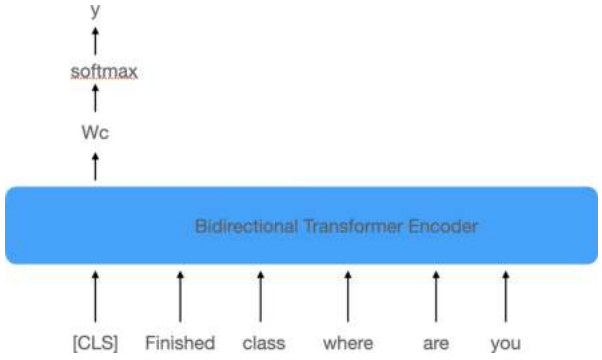


Fig. 6. Sequence classification with a bidirectional transformer encoder

4.1 Fine-Tuning

The process to make use of the representations produced by the pre-trained language models is known as fine-tuning. These semantic representations are helpful to build a sort of pipeline application to cope with NLP tasks such as named entity tagging or sequence classification.

In the case of sequence classification, the key point is the representation of the entire input sequence. Whereas in RNNs the hidden layer corresponding to the last input element denotes the entire sequence, an additional vector in the transformer encoder captures the entire sequence.

This is the reason why this additional vector is called the sentence embedding. The additional vector is symbolized by the [CLS] token which is prepended to the input sequences.

Figure 6 shows the architecture of a transformer encoder for sequence classification where the output of the encoder represented by [CLS] is provided to a neural network classifier that makes the category decision.

By using a labeled dataset, the sequence classification task entails to learn a set of weights (W_C) in order to map the output vector (Y_{CLS}) to a set of categories:

$$y = \mathrm{softmax}(W_C\, Y_{\mathrm{CLS}}). \tag{7}$$

4.2 Data

The dataset used in this experimentation is part of an ambitious research project denominated the Automated Student Assessment Prize (ASAP) [7] for automated grading of student-written responses sponsored by The William and Flora Hewlett Foundation.

The purpose is to explore new forms of testing and grading methods and to reduce the cost of human graders by automating the student assessment. Three stages set up the ASAP project:

- **Phase 1**: Analysis of essays: Long form response.
- **Phase 2**: Analysis of short answers: Short form response.
- **Phase 3**: Analysis of charts/graphs: Symbolic mathematical/logical reasoning.

The focus of our attention is the collection of short-answers corresponding to the phase 2 [3]. Each instance in the collection denotes a short answer corresponding to a reading passage from a broad range of disciplines: from English Language Arts to Science.

More specifically, the dataset is divided into 10 collections, where each one is described by a particular reading passage corresponding to a particular discipline and where the grade is defined in terms of levels of quality or categories.

For instance, the following text is an example of a short answer response where the range of the score is three: 0 (not proficient), 1 (partially proficient), or 2 (proficient).

“Paul is shocked that Mr. Leonard didn’t tell him that he broke all the records he did, and that he won the 400 meter hurdles at nationals when he was only a freshman. Paul also realizes that Mr. Leonard had been trying to help him because he too, was good at something, but couldn’t do it because he didn’t get good enough grades, because he couldn’t read.”

The average length of each answer is approximately 50 words and most training sets contain around 1,800 responses that have been

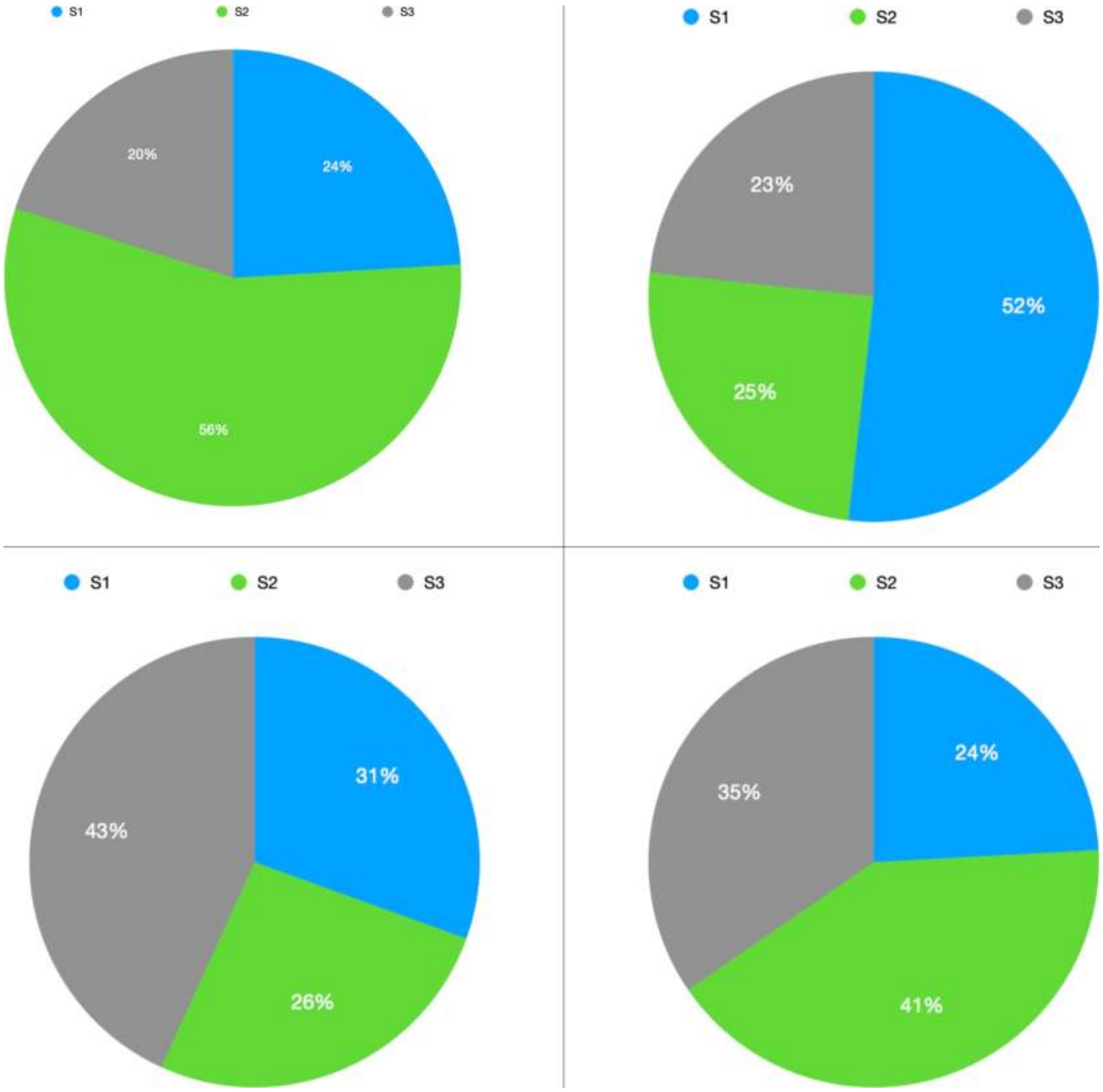


Fig. 7. Data distribution of the short-answers collections 3, 7, 8, 9

randomly selected from a sample of approximately 3,000. From the 10 training collections available in the dataset, we select four training sets where three levels of quality define the grade of each answer.

In other words, the fine-tuning process implemented in our experimentation performs a downstream task as multi-class classification where a short answer is assigned into one of the multiple rubrics of the responses.

The distribution of responses to rubrics corresponding to each training collection is shown in Figure 7.

4.3 Results

In our experiments, we adopt two strategies to the downstream task: the simple use of the embeddings obtained from the pre-trained model, and a more refined optimization of such embeddings via an added classic neural network.

Table 2. Results corresponding to each BERT model for each network architecture and each dataset

BERT model	Architecture	Dataset 3	Dataset 7	Dataset 8	Dataset 9
Compact	Simple	0,64	0,60	0,62	0,65
Compact	Layers	**0,71**	**0,66**	**0,64**	**0,72**
Original	Simple	0,60	0,55	0,56	0,62
Original	Layers	0,61	0,53	0,54	0,63
Universal	Simple	0,69	0,68	0,66	0,72
Universal	Layers	**0,74**	**0,74**	**0,72**	**0,79**

Although there are more sophisticated neural network models such as CNN and RNN, we consider these two simple and basic options as our purpose is to perceive the quality of the embeddings produced by different training methods rather than to obtain a high precision on the downstream task. Thus, the downstream network architectures implemented are:

– Simple: a simple dense layer is used to adjust the pre-trained embeddings obtained from pooled_output. For example, since the number of hidden units of the original BERT model is 768, and our experimentation performs a downstream three-class classification, the number of parameters to be adjusted is 2,307.

– Layers: three dense layers are used to adjust the pre-trained embeddings obtained from pooled_output. The first and second layers contain 64 and 32 hidden units respectively, and since the number of hidden units of the original BERT model is 768, and our experimentation performs a downstream three-class classification, the number of parameters to be adjusted is 51,395.

As the size of the short-answers collections is small, the performance evaluation of the pre-trained models was conducted by the cross-validation method to use all the responses corresponding to a particular domain.

We train our downstream learning models with an Adam optimizer with a learning rate of 0.001, three-fold cross-validation and 25 epochs.

We also apply dropout with $\rho = 0.2$ across layers of the downstream networks to prevent overfitting. Table 2 shows the results obtained in the fine-tuning process where classification of the collection of short-answers is the downstream task implemented for the analysis of the semantic representations obtained from the pre-trained BERT models.

The results are expressed in terms of the F1 score corresponding to each BERT model for each network architecture and each training set. For example, the first row shows a F1 score of 0.64 obtained with the **C**ompact model and a simple network architecture for dataset 3. A deep analysis of the results is carried out in the next section.

5 Discussion

A starting point for our discussion section is the definition of the baseline as a reference point for the obtained results. As it has been described in the data section 4.2, the data collection used in our experimentation is part of a competition for automated grading of student-written responses (ASAP) [7].

Unfortunately, the information available on the competition portal only mentions the winners of the competition but no methodology implemented or obtained results are provided.

But taking into account that our purpose is to perceive the quality of the embeddings produced by different training methods rather than obtain high precision on the downstream task, we define the original BERT model as the baseline model.

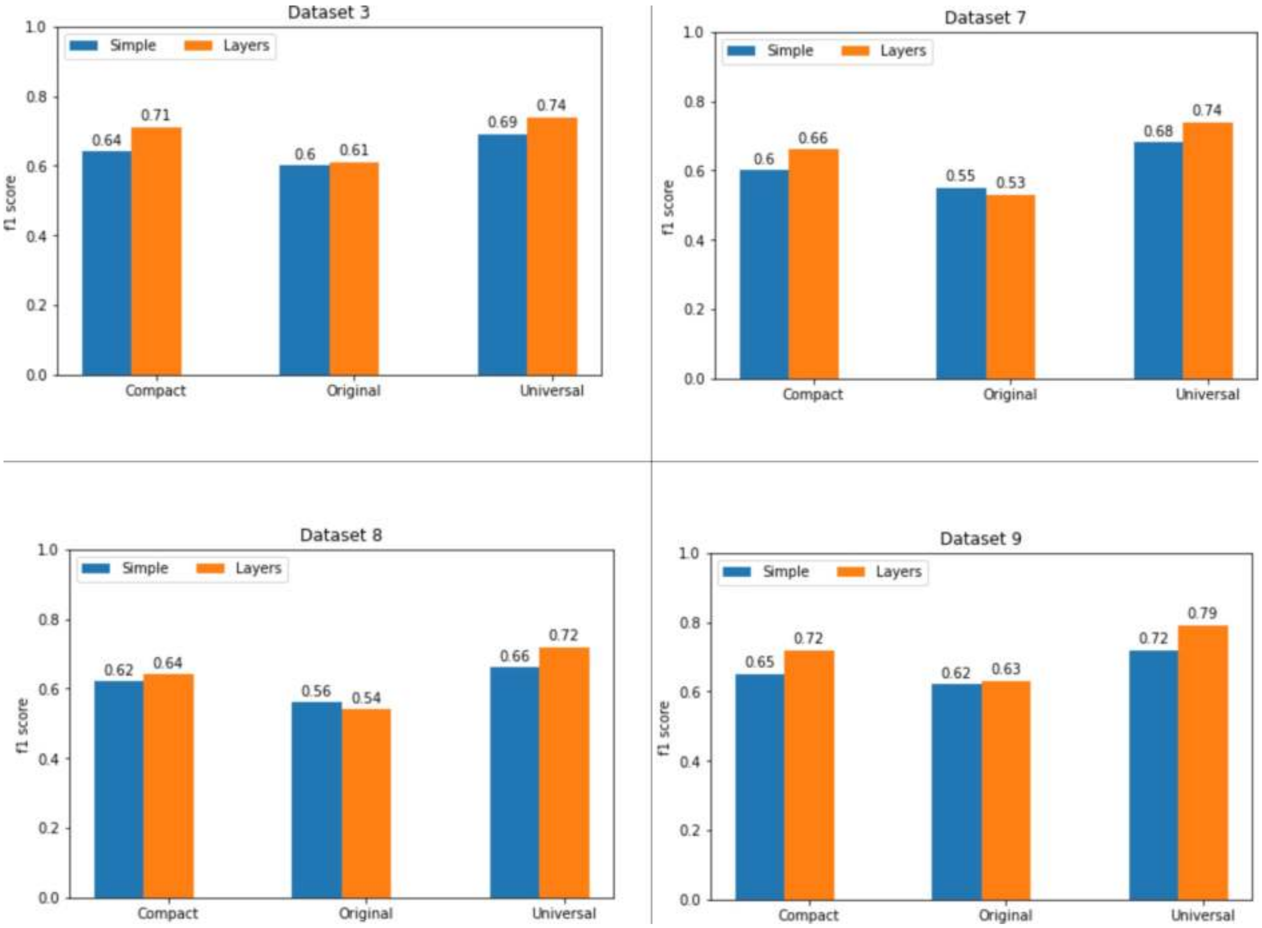


Fig. 8. Results corresponding to each BERT model for each network architecture and each dataset

For the sake of clarity, Figure 8 shows a graphic perspective on the obtained results from Table 2.

5.1 Pre-Trained BERT Models: Unsupervised Training Methods

The research question that guides our work is: what is the impact of the unsupervised training methods on the quality of the semantic representations produced by the pre-trained BERT models? Based on the experimental results, Figure 8 shows how the baseline performance differs from the Compact and Universal models: we can see how the performance of these extended models exceeds that of the original model.

In other words, the obtained results exhibit how the unsupervised training variants contribute to a positive effect on the performance. For example, the highest F1 score obtained for all datasets by the Universal model underpins its argument about the optimization of sentence-level representations.

In fact, the integration of the NSP and MLM training methods, where the MLM task depends on the sentence level representation produced by the NSP task, entails a sort of tradeoff: to perform good MLM, good sentence representations are required.

As we described in section 3.2.3, the CMLM training method of the Universal model makes use of adjacent sentences where the concatenation of the token embeddings of s_2 with the embeddings of the first sentence s_1, are provided to the transformer encoder for the prediction of the masked tokens in s_2. In this way, this training method proves to be the best option to learn and produce sentence level representations.

As for the results obtained by the Compact model, the use of knowledge distillation as training method proves to be a plausible option for transfer learned knowledge to particular tasks.

The F1 score obtained by this Compact model for all datasets have surpassed the corresponding scores obtained by the baseline model. Thus, these results underpins its argument about the successful development of compact models under the self-supervised pre-training paradigm.

In section 3.2.4, we describe how the pre-trained distillation method of the Compact model defines three training operations: initialize a small model (i.e. the student) by pre-training under the MLM task, transfer learned knowledge (i.e. distillation of the teacher knowledge) and the optional fine-tuning on a particular linguistic task such as classification.

In this way, compared to the use of compression techniques on large language models [20, 19], this distillation training method proves to be a well-performing model developed under the self-supervised pre-training paradigm.

5.2 Pre-Trained BERT Models: Fine-Tuning Model Architectures

As suggested by Goodfellow et al. [6], a good representation is one that makes a subsequent learning task easier.

This is the reason why, in order to know about the strengths and weaknesses of the semantic representations extracted from the pre-trained BERT models studied in this work, we implement the fine-tuning process on a downstream classification task.

In other words, we transfer the acquired knowledge obtained from the pre-trained BERT models to solve automated grading of student written responses. Then, we need to figure out which of these representations demand further training to cope with this classification task.

This is the reason why we implement two fine-tuning model architectures: a simple and a forward neural network named in this work as layers. As we described in previous section 4.3, a simple architecture is just a softmax layer whereas our layers architecture is defined in terms of a small forward neural network to determine whether tuning is worth implementing.

Figure 8 highlights important points to be noticed. First, we see how the tuning of the embeddings produced by the Universal and Compact models has been worth of implementing. For all the observed datasets, the F1 score obtained by the use of the layers architecture is higher than the score obtained by the simple architecture. An average increase of 6 points in the F1 score is observed.

On the other hand, we see how the tuning of the embeddings produced by the Original BERT model has not been worth of implementing. For datasets 3 and 9, the F1 score obtained by the use of the layers architecture is a bit higher than the score obtained by the simple architecture (just one point is the difference).

However, for datasets 7 and 8, the F1 score obtained by the use of the layers architecture is lower than the score obtained by the simple architecture. Thus, two points stand out with the use of the semantic representations produced by the Original BERT model: for all the observed datasets, the lowest F1 score has been obtained, and the tuning of the embeddings has not been worth of implementing.

In summary, the embeddings produced by the extended BERT models, Universal and Compact models, have optimized the downstream task. On the other hand, regardless of the fine-tuning learning model implemented, simple or layers architectures, the F1 score obtained with the Original BERT model was lower than the one obtained with the pre-trained BERT variants.

The use of complex downstream network architectures such as CNN or Bi-LSTM could possibly improve the performance of the Original BERT model, but two previous works do not consider this option as a plausible alternative. Zhao et al., in their work about the use of pre-trained LLMs for toxic comment classification, prove that using a basic linear downstream architecture outperforms complex ones such as CNN or Bi-LSTM [29]. Also, in their work about the analysis of multiple embeddings methods for text classification, Wang et al. implement CNN and Bi-LSTM as downstream network architectures and the difference in performance was not significant [26]. For the authors, the difference in performance lies in the characteristics of the data rather than the network architectures of the learning model.

5.3 Pre-Trained BERT Models: Computational Resources

What is the computational cost demanded by the pre-trained BERT models? Since determining the runtime and memory requirement of the pre-trained BERT models is highly platform-dependent, we do not describe the computational cost in absolute terms.

We describe rather the computational cost as a degree of runtime. In order to make a viable explanation for the computational cost incurred by each BERT model, we define a baseline as a reference point for the running time demanded by each model in the fine-tuning process.

So, taking into account the longest running time demanded, we define the Universal BERT model as the baseline model. Since the different downstream network architectures (simple or layers architectures) do not show any discrepancy in terms of the time consumed, we attribute the difference in time to the structure and training of each particular BERT model.

For example, the use of the Universal model gives rise to a tradeoff between classification performance and time: the Universal model demands more time but obtains the best F1 score for all datasets.

As we describe in section 3.3, this Large Language Model is based on the BERT transformer architecture that consists of L=12 encoder layers, a hidden size of $H = 768$, and $A = 12$ attention heads representing a total of 110M parameters. By contrast, the running time demanded by the Compact model is really amazing: this model requires only a third of the time required by the Universal model. And the classification performance is also good: this model achieves better F1 score than the Original model.

As we describe in section 3.3, this Small Language Model is based on a knowledge distillation architecture that consists of $L = 4$ encoder layers, a hidden size of $H = 512$, and $A = 8$ attention heads representing a total of 28M parameters.

In summary, and based on the evidence provided by our experimentation, we conclude this discussion section by considering the Compact model as a plausible alternative when the classification task can tolerate slight faults. Otherwise, and despite the running time demanded, the Universal model is the best option.

6 Related Work

Based on the taxonomy proposed by Qiu et al. for a deep examination of pre-trained language models for NLP [17], we focus our attention in this section on the type of pre-training tasks. More specifically, and given that in this work we address the analysis of three pre-trained BERT models and their corresponding pre-training tasks such as MLM, NSP and Distillation, in this section we make a brief description of pre-training tasks related to those previously mentioned. For example, we start with Dynamic MLM as it is a pre-training task closely related to MLM.

6.1 Dynamic MLM

This pre-training task is implemented in the development of a variant of BERT known as RoBERTa [30]. The purpose of this pre-training method is the optimization of the static masking implemented by MLM in which unique and different maskings are generated for each sequence, so each sequence with the same masking is observed more than once.

Instead, Dynamic MLM generates a unique masking every time a sequences is transferred to BERT training. In this way, a wide diversity of masking patterns is available for the training of BERT. Besides this training method optimization, the training of RoBERTa was implemented with bigger batches, longer sentences and the use of NSP was omitted. In this way, RoBERTa performance achieves state-of-the-art results on a benchmark such as GLUE.

6.2 SOP: Sentence Order Prediction

This pre-training task is implemented in the development of a variant of BERT known as ALBERT [12]. As a sequel to BERT breakthrough, some studies on BERT development suggest the use of next sentence prediction (NSP) as an ineffective training method. In the development of ALBERT, SOP is then introduced to replace NSP. In order to take care of inter-sentence modeling, SOP focuses on coherence between pairs of sentences in a different way to NSP.

Instead of using sentence pairs from different documents as negative examples, SOP makes use of the same two consecutive sentences, used as positive examples in BERT, but with their order swapped. In addition to this new training method, ALBERT implements two parameter reduction techniques to cope with the huge computational resources demanded by BERT.

First, the separation of the hidden layers from the vocabulary embedding to increase the hidden layers without increasing the size of the vocabulary embedding. Second, to share all parameters across layers as a way to improve parameter efficiency. In this way, ALBERT performance achieves state-of-the-art results on a benchmark such as GLUE.

6.3 Transformer Distillation

This pre-training method implements a distillation knowledge technique to reduce the computational overhead of BERT while retaining its performance. A variant of BERT known as TinyBERT is the language model obtained by implementing this transformer distillation technique [9]. Transformer distillation performs layer-to-layer distillation with embedding outputs, hidden states and self-attention distributions. Basically, layer-to-layer distillation consists in choosing M out of N layers from the teacher model where a mapping function is defined for transfer learning from a particular layer of student model to a particular layer of a teacher model. The development of TinyBERT consists of two learning stages: general distillation and task-specific distillation. General distillation makes use of the pre-trained BERT as the teacher to train a smaller student called general TinyBERT with only 4 hidden layers instead of the standard 12.

Because of this significant reduction in the number of hidden layers, general TinyBERT performance is lower than BERT. Now, the purpose of the task-specific distillation is to strengthen the power of TinyBERT by applying again transformer distillation but now having as teacher the knowledge of fine-tuned BERT.

This process makes use of a data augmentation method on a task dataset in order to expand the task-specific training dataset. In this way, TinyBERT performance achieves state-of-the-art results on a benchmark such as GLUE.

7 Conclusion and Future Work

In this paper, we analyze the influence of unsupervised training methods on the development of pre-trained language models for learning linguistic representations. In particular, we study three pre-trained BERT models and their corresponding unsupervised training tasks such as MLM, NSP, CMLM and Distillation.

A broad outline of the pre-training process for each BERT variant allows to consider similarity and differences between them. We conduct fine-tuning as an empirical evaluation on a downstream classification task with a learning model defined in terms of the semantic representations produced by each BERT model. In this way, our experimentation provides empirical evidence of the quality of the embeddings produced by these pre-trained language models.

For example, the results show how the tuning of the embeddings produced by the Universal and Compact models has been worth of implementing as the F1 score obtained by the use of the layers architecture is higher than the score obtained by the simple architecture whereas the tuning of the embeddings produced by the Original BERT model has not been worth of implementing.

Finally, we obtain insight into the computational resources demanded by the BERT models analyzed in this work. The efficiency of the Compact model was rather astonishing. Based on the work about the identification of linguistic properties of data for which contextual embeddings contribute with a significant improvement on performance [1], our future work will explore the linguistic properties of data for which pre-trained models improve performance during downstream task. Said in another way, we will identify linguistic properties of data for which pre-trained models will exhibit the strengths and weaknesses of their corresponding unsupervised training methods.

References

1. **Arora, S., May, A., Zhang, J., Ré, C. (2020).** Contextual embeddings: When are they worth it? Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 2650–2663. DOI: 10.18653/v1/2020.acl-main.236.

2. **Balestriero, R., Ibrahim, M., Sobal, V., Morcos, A., Shekhar, S., Goldstein, T., Bordes, F., Bardes, A., Mialon, G., Tian, Y., Schwarzschild, A., Wilson, A. G., Geiping, J., Garrido, Q., Fernandez, P., Bar, A., Pirsiavash, H., LeCun, Y., Goldblum, M. (2023).** A cookbook of self-supervised learning. DOI: 10.48550/ARXIV.2304.12210.

3. **Barbara, Hamner, B., Morgan, J., lynnvandev, L., Shermis, M. (2012).** The Hewlett foundation: Short answer scoring.

4. **Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019).** BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 4171–4186.

5. **Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019).** bert_en_uncased.

6. **Goodfellow, I., Bengio, Y., Courville, A. (2016).** Deep learning. MIT Press.

7. **Hamner, B., Morgan, J., lynnvandev, L., Shermis, M., Vander-Ark, T. (2012).** The Hewlett foundation: Automated essay scoring.

8. **Hinton, G., Vinyals, O., Dean, J. (2015).** Distilling the knowledge in a neural network. DOI: 10.48550/ARXIV.1503.02531.

9. **Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., Liu, Q. (2020).** TinyBERT: Distilling BERT for natural language understanding. Findings of the Association for Computational Linguistics: Empirical Methods in Natural Language Processing, pp. 4163–4174. DOI: 10.48550/ARXIV.1909.10351.

10. **Jurafsky, D., Martin, J. H. (2023).** Speech and language processing.

11. **Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., Fidler, S. (2015).** Skip-thought vectors. Proceedings of the 28th International Conference on Neural Information Processing Systems, Vol. 2, pp. 3294–3302. DOI: 10.48550/ARXIV.1506.06726.

12. **Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. (2019).** ALBERT: A lite BERT for self-supervised learning of language representations. Proceedings of the 8th International Conference on Learning Representations, pp. 1–17.

13. **Logeswaran, L., Lee, H. (2018).** An efficient framework for learning sentence representations. Proceedings of the 6th International Conference on Learning Representations, pp. 1–16. DOI: 10.48550/ARXIV.1803.02893.

14. **Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J. (2013).** Distributed representations of words and phrases and their compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems, Vol. 2, pp. 3111–3119. DOI: 10.48550/ARXIV.1310.4546.

15. **Pennington, J., Socher, R., Manning, C. (2014).** GloVe: Global vectors for

word representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543. DOI: 10.3115/v1/d14-1162.

16. **Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L. (2018).** Deep contextualized word representations. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 2227–2237. DOI: 10.48550/ARXIV.1802.05365.

17. **Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X. (2020).** Pre-trained models for natural language processing: A survey. Science China Technological Sciences, Vol. 63, No. 10, pp. 1872–1897. DOI: 10.1007/s11431-020-1647-3.

18. **Rani, V., Nabi, S. T., Kumar, M., Mittal, A., Kumar, K. (2023).** Self-supervised learning: A succinct review. Archives of Computational Methods in Engineering, Vol. 30, No. 4, pp. 2761–2775. DOI: 10.1007/s11831-023-09884-2.

19. **Sanh, V. (2019).** Smaller, faster, cheaper, lighter: Introducing DistilBERT, a distilled version of BERT.

20. **Sun, S., Cheng, Y., Gan, Z., Liu, J. (2019).** Patient knowledge distillation for BERT model compression. DOI: 10.48550/ARXIV. 1908.09355.

21. **Taylor, W. L. (1953).** Cloze procedure: A new tool for measuring readability. Journalism and Mass Communication Quarterly, Vol. 30, No. 4, pp. 415–433. DOI: 10.1177/107769905303000401.

22. **TensorFlow (2023).** Tensorflow hub.

23. **Turc, I., Chang, M. W., Lee, K., Toutanova, K. (2019).** small_bert/bert_en_uncased.

24. **Turc, I., Chang, M. W., Lee, K., Toutanova, K. (2019).** Well-read students learn better: On the importance of pre-training compact models. DOI: 10.48550/ARXIV.1908.08962.

25. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017).** Attention is all you need. Neural Information Processing Systems 17: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 6000–6010. DOI: 10.48550/ARXIV.1706.03762.

26. **Wang, C., Nulty, P., Lillis, D. (2020).** A comparative study on word embeddings in deep learning for text classification. Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval, pp. 37–46. DOI: 10.1145/3443279.3443304.

27. **Yang, Z., Yang, Y., Cer, D., Law, J., Darve, E. (2021).** universal-sentence-encoder-cmlm.

28. **Yang, Z., Yang, Y., Cer, D., Law, J., Darve, E. (2021).** Universal sentence representation learning with conditional masked language model. Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 6216–6228. DOI: 10.18653/v1/2021.emnlp-main.502.

29. **Zhao, Z., Zhang, Z., Hopfgartner, F. (2021).** A comparative study of using pre-trained language models for toxic comment classification. Companion Proceedings of the Web Conference, pp. 500–507. DOI: 10.1145/3442442.3452313.

30. **Zhuang, L., Wayne, L., Ya, S., Jun, Z. (2021).** A robustly optimized BERT pre-training approach with post-training. Proceedings of the 20th Chinese National Conference on Computational Linguistics, pp. 1218–1227.

Article received on 14/10/2023; accepted on 14/12/2023.
** Corresponding author is Diego Uribe.*

ISSN 2007-9737

Resurrection: The Khazar Language Reconstruction Using Computer Science Technologies

Elina Makipova[1], Iskander Akhmetov[1,2], Alexander Gelbukh[*,3]

[1] KIMEP University, College of Humanities and Education,
Kazakhstan

[2] Insitute of Information and Computational Technologies, Almaty,
Kazakhstan

[3] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

elina.makipova@kimep.kz, i.akhmetov@ipic.kz,
gelbukh@cic.ipn.mx

Abstract. Decrypting or reconstructing extinct languages is challenging, especially when the objective is to reconstruct a language with no or very few texts left, such as the Khazar language or early Slavic and Ugric languages. In this paper, we lay out the historical perspective of the Khazar people, their language, and contemporary descendant ethnic groups, namely the Chuvash and Tatar people. Then we discuss ways Computer Science can help researchers in language reconstruction and decryption. Finally, we pilot an approach to find Khazar/Bulgar word candidates in Chuvash and Tatar languages by (1) normalizing the words of two languages and (2) comparing them, accounting for the semantic concepts to solve the homonymy problem, and (3) excluding common Turkic words and borrowings from the Russian language.

Keywords. Khazar, language reconstruction, extinct languages, historical linguistics.

1 Introduction

Nowadays, there are more than thousands of different languages that can disappear.

Of the approximately 6,000 existing languages in the world, more than 200 have become extinct during the last three generations, 538 are critically endangered, 502 are severely endangered, 632 are definitely endangered, and 607 are unsafe [14]. However, some people think it is unimportant, because languages are much easier nowadays than before, so there is no need to learn and study them.

Moreover, it can seem unnecessary because no one speaks these languages, so there is no need to recognize them. However, reconstructing or decrypting an extinct language can significantly benefit by filling up the gaps in our historical knowledge and linguistics.

For instance, the language of ancient Egyptians can seem useless because people in Egypt do not speak this language anymore and use Arabic instead. Nevertheless, there are many scriptures in the Pyramids of Giza which scientists decrypt to learn more about the history, life, and tradition of the people living millennia before in the region.

The Voynich manuscript decryption is another example of the ancient language deciphering task which is not been solved to date [21]. However, what can we do about an ancient language that left no written artifacts to decrypt?

One of the approaches from historical linguistics is called Language Reconstruction, when we use a language or a set of languages

doi: 10.13053/CyS-28-1-4902

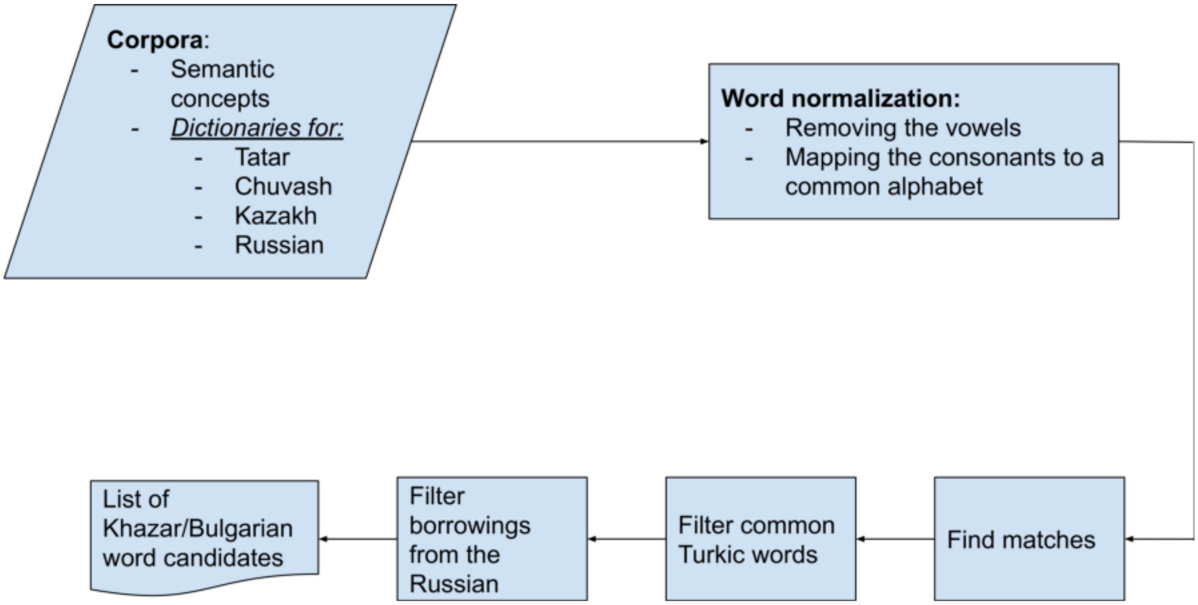

Corpora:
- Semantic concepts
- *Dictionaries for:*
- Tatar
- Chuvash
- Kazakh
- Russian
Word normalization:
- Removing the vowels
- Mapping the consonants to a common alphabet
Find matches
Filter common Turkic words
Filter borrowings from the Russian
List of Khazar/Bulgarian word candidates

Fig. 1. Screening for Khazar/Bulgar candidate words

known to be descending from an ancient language and try to reconstruct the ancestor language seeking for language anomalies and comparing the languages to discover common lexicon.

Thus, there are many works on reconstructing the proto-Indoeuropean language known to be the ancestor of all Indoeuropean languages and the works on reconstructing the proto-Turkic language [4]. In this article, we attempted to reconstruct some Khazar words which once were used in the Khazar khaganate.

The country spread from the Aral Sea in the East to the Crimean peninsula in the West in early Medival times. Scientists know very little about these mysterious people; still, their language has left no written evidence other than some personal names and toponyms we can find from the Arabian and Byzantium historians' works [5].

There are a lot of linguists and scientists who tried to unravel this language. However, they could know only a bit.

This example shows that an extinct language is a key to understanding the natural history of a particular nation. Reconstructing extinct languages is a challenging problem of an interdisciplinary nature, touching such areas of research as history, geography, linguistics, Computer Science (Artificial Intelligence, Computational Linguistics, Natural Language Processing), and others.

Our approach employed a comparative method of language reconstruction using Chuvash, Tatar, and Kazakh languages. It consisted of (1) normalizing the words by eliminating the vowel characters and mapping consonant characters of the compared languages to a standard alphabet and (2) finding matches between normalized Chuvash and Tatar words, which additionally share the same semantic concept to tackle the homonymy problem, (3) filter out the common Turkic words by eliminating the matches between Chuvash and Kazakh languages (as the Kazakh language is known to have no Khazar/Bulgar background), (4) filter out the words borrowed from the Russian language; see Figure 3.

The contribution of this work to the scientific knowledge is in (1) the approach and algorithm for discovering the Khazar/Bulgar word candidates in modern Chuvash and Tatar languages and (2) dataset with normalized words for Chuvash, Tatar, Kazakh and Russian languages[1].

[1] The code and data are available at github.com/iskander-akhmetov/Khazar-language-resurrection

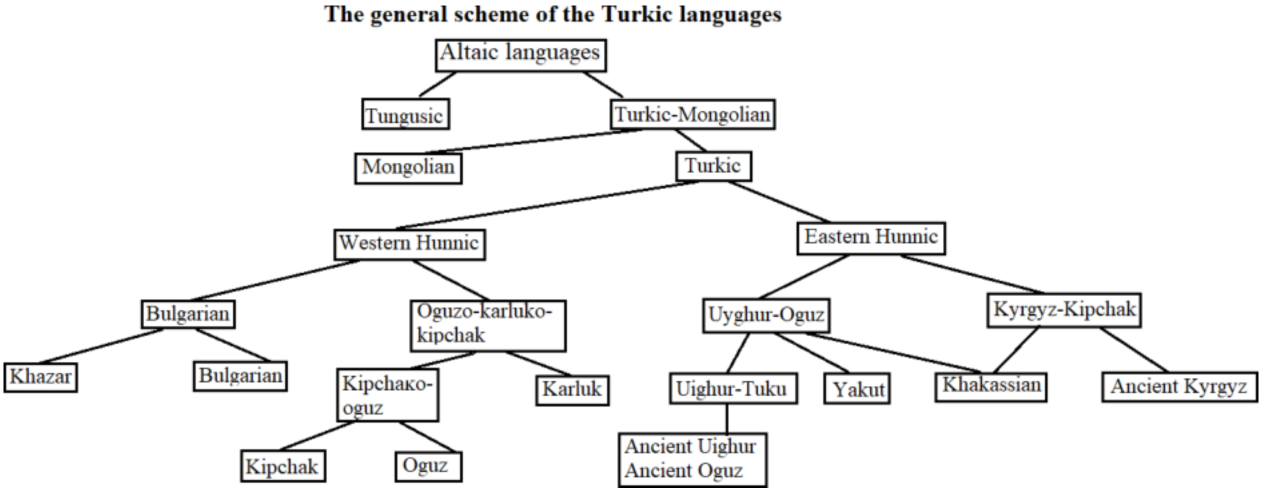


Fig. 2. The general scheme of the Turkic languages

In the following sections of the article, we will give an overview of the Khazar history and language and talk about the descendants of the Khazar people living nowadays and their ethnic groups. Then we will talk about the use of modern technologies in language reconstruction, data, methodology, experiments, and results.

2 Khazar Language History

Before observing the Khazar language, we must see how this ancient nation lived. At the very beginning, we have to observe their culture. We do it for a purpose because language cannot exist without history. We must mention that the Khazar language and culture are similar to the Tatar, Bulgarian, and Chuvash ones.

That is why it is essential to analyze the history of its neighbors and languages too. Based on the information about Khazars and their neighbors, we can find the common features between them.

At the very beginning need to start with the history of the Khazar nation, mainly how it was founded. There are many issues about this exciting nation like Khazar.

Some scientists consider that their language belonged to the Semitic language family; others attribute it to the Bulgarian branch of the Turkic language family. Still, there are plenty of questions about their history and culture.

2.1 Bulgars

History. The first step of the beginning of Great Bulgaria was not an easy job. The Bulgars nation decided to create their own country when they tried to escape from the powerful Khazar Khaganate. During some time, When the Bulgars finally created their own "Empire", the ruling elite formed a unique ethno-political identity and culture.

However, their country did not exist for an extended period of time. Bulgars became the dominant tribe and formed the military service elite of society [10].

Language. The Bulgarian language is a part of the Turkic languages. Today this language does not exist anymore. The Bulgarian language was widespread in the 13th-14th centuries in the Volga region. Arabo-graphic epitaphs were found first on the territory of Volga-Kama Bulgaria.

The Bulgar language and the modern Chuvash language make up the Bulgar group of Turkic languages. Their main regularity lies in the transition from *r'>r, as well as *-d->-r-, by the transition *-l'>-l at the end of the syllable [12].

Bulgar, like all ancient languages, used the runic alphabet. Moreover, the Bulgar language has two main dialects: the Bulgar language and the

Table 1. NorthEuraLex corpora word content by the language used in this study

Language	Number of words
Chuvash	1,210
Tatar	1,149
Kazakh	1,312
Russian	1,037

Suvar language. The second one is nowadays the Chuvash language.

Additionally, scientists consider the Khazar language similar to the Bulgar language. People from Southern Bulgaria could understand people from the Khazaria. However, nowadays, we can see their footprints only in the Chuvash language [20].

2.2 Khazars

History. First, we must mention that different linguists have different points of view about the Khazar language. Some scientists and linguists consider that the Khazar Khaganate has the same roots as the Uighur Khaganate.

"Based on the fact that the Chinese name of the Khazars = k'o-sa closely resembles the name of six of the nine Uighur tribes of Kesa, some researchers classify the Khazars as Uighurs and believe that they appeared in Europe together with the Huns or after them in the VI century" [2, 15].

However, the author refutes this version. The language of Khazar khaganate is similar to the Bulgarian language, and it is close to the Turkic languages [2, 7]. Later there was a battle between the Armenian ruler and the Khazar nation. Whereas as a result, Armenian won [2].

Language. In the previous section, we talked about the history of the Khazar nation, and here we will see what the Khazar language looked like. Khazar language does not have many texts, so we must reconstruct it. Moreover, scientists still cannot understand what kind of language it is. That is still a question. The only source of Khazar words are names of kings and toponyms of Khazaria from non-Khazar historical manuscripts available to researchers [16].

Ibn Hordabeh states that the Khazar language is identical to the Bulgar language but different from the Burtas, Persian, and Russ (people of Scandinavian origin, known as Vikings or "varyags" languages [8]. From this information, we can conclude that it belongs to the family of Turkic languages; see Fig. 2, or, more specifically, to its oldest branch, which separated from the general Turkic unity first of all [16].

Nevertheless, the Khazar language presumably belongs to the Bulgarian group of languages. However, unfortunately, we have only one alive language from the Bulgarian language family. This alive language is the Chuvash language, which is commonly spoken in the Chuvash Republic in Russia.

"That is why the data of the Chuvash language is essential for studying the question of the Khazar language. In addition, the analysis of the early Turkisms in the Hungarian language, many of which are borrowings from the Khazar language, testifies in favor of the version about the Turkic affiliation of the Khazar language" [16].

Moreover, two types of alphabets were used by Khazars. The Don letter, represented by the inscriptions of the Mayak settlement, and the Kuban letter, are the only monuments that are inscriptions found during archaeological research of the Humarin fortress [16].

Furthermore, some linguists consider that this language can be similar to the Ossetian language. "The text written by this hand should be read in a language close to the Digor dialect of the modern Ossetian language. Thus, the language of the texts written in Runic script, distributed on the territory of the Khazar Khaganate, is not Turkic but Iranian in origin. That is, it is not a proper Khazar language." [16, 13].

2.3 Contemporary Descendants

2.3.1 Chuvash

History. The nation that can attract people's attention is Chuvash. Today we can observe the territory of the Chuvash in the Middle Volga region. The Chuvash speak the Turkic language, a linguistic relic of the Western ancient Turkic language also called "Bulgar" or "Ogur".

Table 2. NorthEuraLex semantic concepts

id	Name	English	German	Russian
1	EYE	eye [[anatomy]]	Auge [[Anatomie]]	глаз [[анатомия]]
2	EAR	ear [[anatomy]]	Ohr [[Anatomie]]	ухо [[анатомия]]
3	NOSE	nose [[anatomy]]	Nase [[Anatomie]]	нос [[анатомия]]
4	MOUTH	mouth [[anatomy]]	Mund [[Anatomie]]	рот [[анатомия]]
5	TOOTH	tooth [EX:human incisor]	Zahn [BSP: Schneidezahn]	зуб [НАПР:человека]

Their neighbors are speakers of Eastern Turkic, Finn-Ugric, and Slavic languages, and historically in contact as in the Iranian world, the Chuvash, in many respects, is an excellent, illustrative example of the complexity of ethnogenesis, the mixing of ethnic groups, languages, and cultures that make up the people.

In the past, the Chuvash led a fairly diverse lifestyle, following various economic pursuits (sedentary agrarian lifestyle, pastoral nomadic lifestyle, hunting, and gathering) in the steppe, forest-steppe, and forest zones into clans, tribes, tribal unions, states, and sometimes empires. The Chuvash rarely engaged in any business alone often, they joined groups for this [1]. Some scientists firmly believe that the ancestors of the Chuvash were known as Savirs/Suvars [19].

Language. Scientists say that the ancestors of the Chuvash were Turkish nomads, and they immigrated from the West to middle Asia and moved off to Eastern Europe. This country's language is unique because it resembles the Mongolian and Finno-Ugric languages. However, scientists still argue that the Chuvash language belongs to the Turkic languages. Bulgarian Turks, the ancestors of Chuvash people, were the first Turkic clan that immigrated to the West and separated from the Central Asia Turkic community.

This immigration is thought to have happened at the beginning of the first centuries AD. For this reason, the Chuvash language, among the Turkic languages, is the oldest and represents Turkic all by itself. Because this language has Mongolian and Finno-Ugric characteristics, some scientists consider that this language was connected with the Mongolian and had similar culture and language in the past.

However, after some time, this language started to develop itself due to historical events [23]. In the past times, people used the same alphabet (runic alphabet as Bulgars did), and here there is a modern Chuvash alphabet: Аа, Ăă, Бб, Вв, Гг, Дд, Ее, Ёё, Ĕĕ, Жж, Зз, Ии, Йй, Кк, Лл, Мм, Нн, Оо, Пп, Рр, Сс, Çç, Тт, Уу, Ӳ̆ӳ̆, Фф, Хх, Цц, Чч, Шш, Щщ, ъ, Ыы, ь, Ээ, Юю, Яя .

2.3.2 Tatars

History. After the breaking of the Eastern Turkic khaganate, Kimaks and Kipchaks created their khaganate and called it "Kimak khaganate". At the same time, their powerful neighbor Bulgars created their own country and called it "Great Bulgaria".

After some time, when Great Bulgaria was broken and divided into two parts, "Danube Bulgaria and Volga-Kama Bulgaria", Danube Bulgaria combined with Slavic nations and accepted Orthodox religion meanwhile another part Volga-Kama Bulgaria combined with Turkic and Ugric tribes and accepted Islam religion.

After it, Volga-Kama Bulgaria, was conquered by the Mongols and used to be a part of the Golden Horde. When the Golden Horde was separated into several independent states such as Astrakhan, Crimea, and Kazan khanates, all of these gradually became the part of Russian Empire on its rise, and contemporary Tatar ethnic groups formed within it in the 19th century as local Muslim and Turkic communities [11].

Language. The Tatar language is widely spoken in the Tatarstan Republic. This language has several dialects, and all of these dialects are different. At the beginning of the 20th century, Tatar nations were combined. Additionally, this language

Table 3. Results sample of list of possible Khazar/Bulgar words

Norm.	Tatar	Chuvash	Concept
бс [bs]	бәс [bæs]	пас [pas]	HOARFROST
сдсб [sdsb]	савыт-саба [savɨt-saba]	савӑт-сапа [savət-sapa]	DISHWARE
бсбк [bsbk]	башмак [baʂmak]	пушмак [puʂmak]	SHOE
бндр [bndr]	мендәр [mendær]	минтер [minter]	PILLOW
ср [sr]	чир [tʃir]	чир [tʃir]	DISEASE
сд [sd]	оста [osta]	ӑста [əsta]	MASTER
сл [sl]	усал [usal]	усал [usal]	EVIL
кскр [kskr]	кычкыру [kɨʧkɨru]	кӑшкӑр [kəʃkər]	SHOUT
сл [sl]	сулау [sulau]	сывла [sɨvla]	BREATHE
рд [rd]	ярату [jaratu]	юрат [jurat]	LOVE
сдр [sdr]	өстерәү [østeræw]	сӗтӗр [sətər]	DRAG

is a part of the Turkic languages and its Kipchak branch; they have three dialects of their language (Western, Eastern, and Middle).

In the middle is Zakamsky, Paranginsky, Nagorny, Menzelinsky, Birsky, Perm, Nokratsky, Kasimov; In the west people speak Sergachsky, Drozhzhanovsky, Chistopolsky, Melekessky, Temnikovsky, Kuznetsky; Finally, in the east there are Tobolo-Irtysh, Tyumen, Barabinsky, and Tomsk. During the creation of the Tatar Republic, their language was mixed and interacted with other languages.

Its neighbors are Bashkirs, Finno-Ugric, Mordovian, Mari, Udmurt, and Slavic languages [18]. Tatars traditionally adopted the Arabic alphabet, which was replaced for a short time by the Latin alphabet used by all Turkic people, and finally converted to a Cyrillic alphabet adaptation [22]. That is a modern version of the Tatar alphabet:

А а, Ә ә, Б б, В в, Г г, Д д, Е е, Ё ё, Ж ж, Җ җ, З з, И и, Й й, К к, Л л, М м, Н н, Ң ң, О о, Ө ө, П п, Р р, С с, Т т, У у, Ү ү, Ф ф, Х х, Һ һ, Ц ц, Ч ч, Ш ш, Щ щ, Ъ ъ, Ы ы, Ь ь, Э э, Ю ю, Я я.

3 Computer Science and Extinct Languages

Computers and different technologies can help us to solve many problems. One of them is the decryption of extinct languages. It can be complex and lengthy work if done manually; meanwhile, the technologies can solve it faster.

Let us see how it works. Instead of spending half of their life trying to get something from an extinct language, for instance, as people did with the Egyptian language, computers can take just several hours for this work. For example, utilizing computer technologies, it was possible to decrypt the Ugaritic language for several hours [6].

First, if we want to decrypt the target language, we need to know which languages can be similar to the target language. In the case of the Ugaritic language, scientists discovered that the most similar language is Hebrew. Without this comparison, it would be hard for the computer to find common features.

Computers can also help a lot with the computation of statistical features of a language, such as character or word distributions, word co-occurrences, and many others. The main thing scientists can do for the extinct language is to find out the "possible" language family of the target non-decrypted language [6].

Table 4. Examples of Out of Vocabulary (OV) words in Kazakh language

Norm.	Tatar	Chuvash	Kazakh OV	Concept
жр [ʐr]	җир [ʒir]	çӗр [stʲer]	жер [ʐer]	SOIL
д [d]	ут [ut]	вут [vut]	от [ot]	FIRE
сск [ssk]	чәчәк [tʃætʃæk]	чечек [tʃetʃek]	шешек [ʂeʂek]	FLOWER
срк [srk]	сарык [sarɨk]	сурӑх [surəx]	сарық [sarɨq]	SHEEP
кккк [kkkk]	кәккүк [kækkyk]	куккук [kukkuk]	көкек [køkek]	CUCKOO

Additionally, nowadays, people speak around 6,000 languages; meanwhile, in the past, people spoke approximately 31,000 languages. As a result, people started to lose history. Via the languages, people can know the history.

In 2010 people had to know the relationships between languages to decrypt extinct languages with the help of AI. Today, machines can decrypt it without any comparison, in other words, no need to know the language family of the extinct language if we want to decrypt it in AI. To sum up, technology can help in linguistic research affairs in many ways.

Even linguists use it to know the history of the past. Machines, without any doubt, are developing year by year. However, it will take more time. With the help of machines, people can decrypt or reconstruct extinct languages faster. That is why we need to incorporate computer technologies in our research [9].

4 Data

NorthEuraLex 0.9 corpora[2] amongst 107 languages of Northern Eurasia contains datasets for Tatar, Chuvash, Kazakh and Russian languages (Table 1, and includes orthographic form of words with International Phonetic Alphabet (IPA) transcription and the semantic concept labels (Table 2).

Corpora contains 1,016 semantic concept tags explained in English, German, and Russian languages [3].

5 Methodology

The linguistic reconstruction task is to recover the lexicon, grammar, and syntax of an extinct language with no written text artifacts (unattested language) but known to be the ancestor of one or more live languages. A word rooting down to a proto-language is called reflex, and reflexes from the same root are cognate. The task can be approached in two major ways:

1. **Internal Reconstruction** exploits single language anomalies and irregularities to infer about earlier stages of language development, collecting the facts within the language studied.

   In internal reconstruction, the language is compared with itself, as it has changed over time, and we are looking for anomalies in morphology and grammar that may indicate linguistic features of the proto-language.

2. **Comparative Reconstruction** is finding a common ancestor for two or more languages from the same language group using the comparative method. The ancestor language is referred to as the proto-language of a given language family.

   The most famous examples of Proto-languages are Proto-Indo-European, Proto-Semitic, Proto-Turkic, and Proto-Dravidian because they are the most popular and common proto-languages that are being constantly researched by the scientific community.

   Languages, that are thought to have a common proto-language, are grouped together according to following criteria [17]:

[2] www.northeuralex.org/

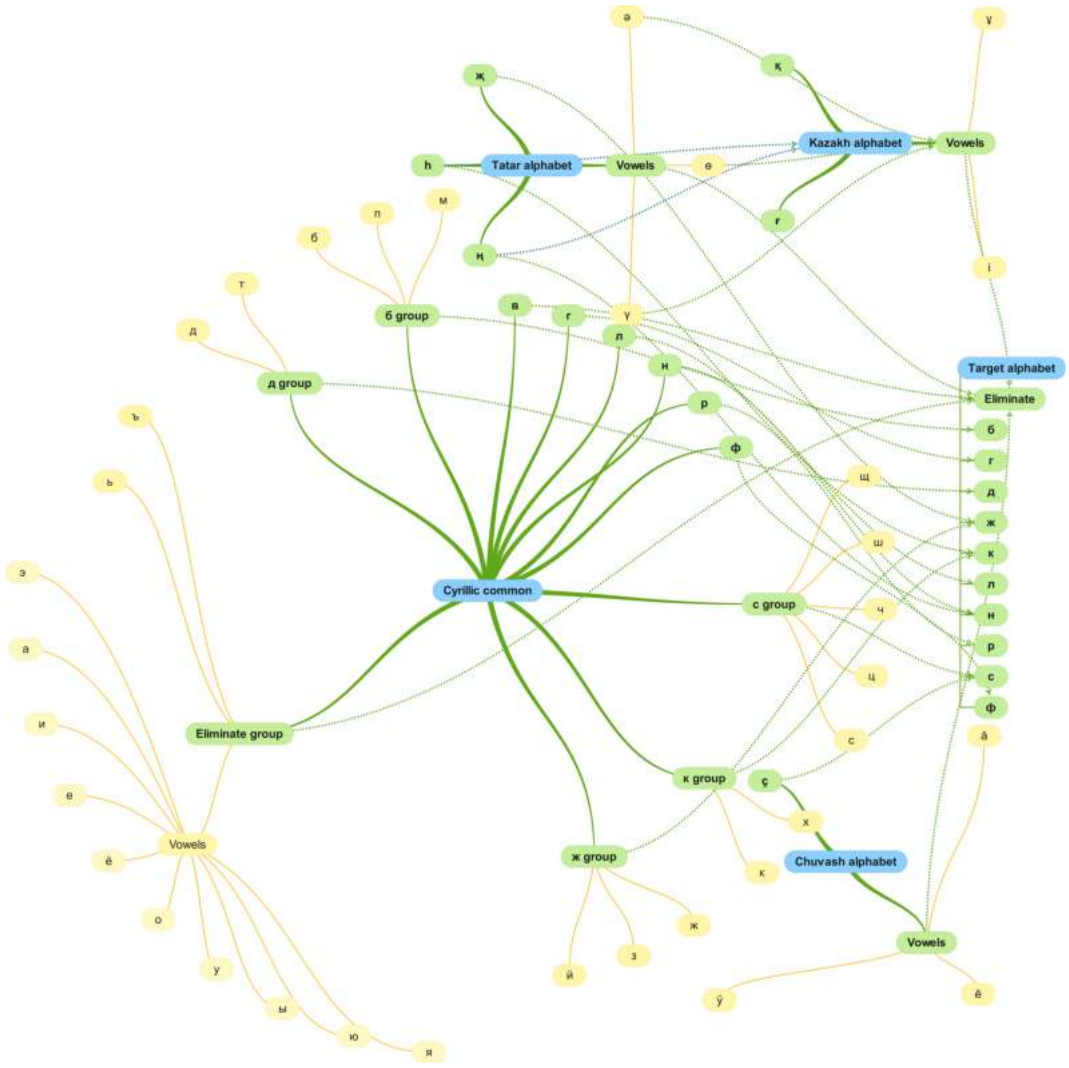


Fig. 3. Character mapping rules

- **Shared Innovation** meaning that the languages show common changes throughout time.
- **Shared Retention** which is opposite to the first criterion, meaning that languages preserve common features.

Comparative reconstruction exploits two major principles [24]:

- **The Majority Principle**, which observes that if cognates display a pattern, similar to repeating letter appearing in certain position within a word, then it is possible that the pattern was retained from the proto-language.

- **Most Natural Development Principle** proposes commonly appearing changes in languages throughout the time:

  - Omitting of final vowel in a word.
  - Consonants at the end of words become voiceless.

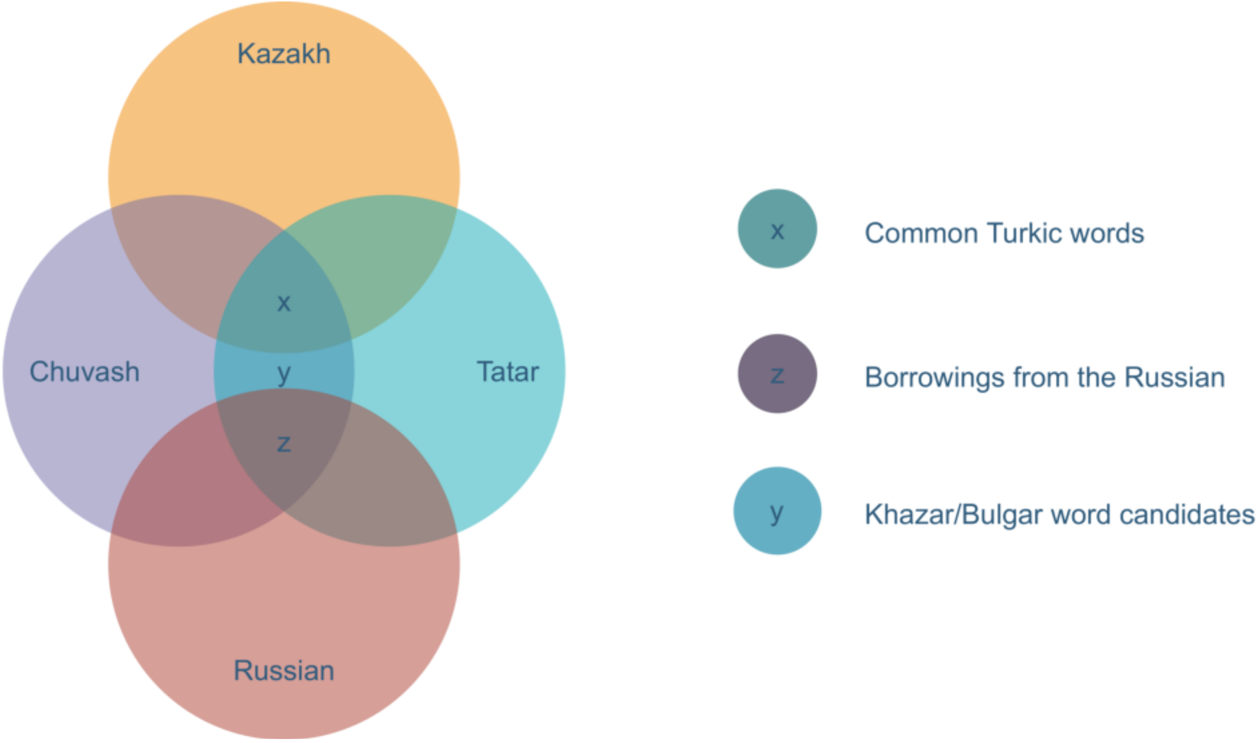


Fig. 4. Khazar/Bulgar word candidates selection in Venn diagram view

- Voiceless sounds appearing between vowels become voiced.
- Phonetic termination becomes fricative.

6 Experiment

Using the comparative method of language reconstruction, we have compared Chuvash and Tatar languages to find common words of possible Khazar/Bulgar origin.

1. Normalizing the words:
   - Remove vowels.
   - Character mapping rules; see Figure 3.
2. Find matching of normalized words in Tatar and Chuvash languages, with matching concept.
3. Exclude common Turkic words which match with the Kazakh language.
4. Exclude borrowed words from the Russian language.
5. Obtain the list of Khazar / Bulgar word candidates.

The overall process of obtaining the Khazar/Bulgar word candidates can be expressed by the Venn diagram shown in Figure 4.

7 Results

Some 185 normalized word and concept matches between Tatar and Chuvash languages were found (Figure 4 X, Y, and Z combined). Furthermore, 64 matches were left after filtering out common Turkic words (matches with the Kazakh language) and borrowings from the Russian language ((Figure 4 Y only); see Table 3 for a sample of 10 words of possible Khazar/Bulgar origin.

8 Discussion

8.1 Validity of Filtering Common Turkic Words

Briefly, the experiment included the stage where we filtered out presumably common Turkic words, which were indicated by the matching between normalized words in Chuvash and Kazakh language datasets. We assumed that the Kazakh language has no traces of Khazar/Bulgarian origin.

However, there might be some interactions as the Khazar khanate included some parts of modern Kazakhstan territory and bordered Khorezm in the past, which imposes 2 crucial questions:

- How different was the Khazar/Bulgar language from all the other Turkic languages back then and from the contemporary Turkic languages?
- How to differentiate words of Khazar/Bulgar origin in contemporary Turkic languages?

We also noticed that among those 64 words we obtained, there are still common Turkic words for which we have analogs in Kazakh, but they were not in the Kazakh language dataset we used; see Table 4. Therefore, we must repeat our experiments for all four languages on much larger corpora.

8.2 Finn-Ugric Components in Chuvash and Tatar Languages

Chuvash and Tatar languages might also share a lexicon borrowed from their Finn-Ugric neighbors: Mari, Udmurt, and Mordva people. Therefore to better distill the results, we need to account for the possible admixture from their languages and filter them out. Moreover, the neighbors could also borrow these words from ancient Bulgars or Khazars. We will need to compare their languages with their language family members who have no known contact with Khazars fixed in the history.

On the other hand, we might get better results by adding Karaim, Kumyk, and Balkar languages to the comparison, benefiting from the fact that these ethnic groups are also closely related to Khazars and Bulgars have no or little contact with Finn-Ugric people. However, they might have words from the Arabic, Persian, and neighboring Caucasian languages.

9 Conclusion

In conclusion, we want to emphasize the importance of the research in the direction of reconstruction and decrypting of extinct languages. Because it allows us to understand ancient scripts and, at the same time, makes it possible to look at the world with the eyes of our ancestors through the prism of their language. For future works, we plan:

1. Perform the experiments on significantly larger corpora.
2. Include the Karaim, Kumyk, and Balkar languages in the analysis.
3. Search for Khazar/Bulgar words in non-Turkic languages, such as Hungarian, Russian, Ukrainian, Bulgarian, and Chechen.
4. Use Bulgar vocabulary to find analog words in other Turkic and non-Turkic languages and then train a classifier model to find other possibly Khazar/Bulgar words.
5. Perform etymological analysis of the candidate Khazar/Bulgar words.

Acknowledgments

This work was supported by the Ministry of Education and Sciences of the Republic of Kazakhstan under the grant #AP14871214: “Development of machine learning methods to increase the coherence of text in summaries produced by the Extractive Summarization Methods”.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

1. **Anton, S. (2014).** Savirs - Bulgars - Chuvash. LAP Lambert Academic publishing.
2. **Artamonov, M. (1962).** Khazars’ history. Hermitage.

3. **Dellert, J., Daneyko, T., Münch, A., Ladygina, A., Buch, A., Clarius, N., Grigorjew, I., Balabel, M., Boga, H. I., Baysarova, Z., Mühlenbernd, R., Wahle, J., Jäger, G. (2019).** NorthEuraLex: A wide-coverage lexical database of northern Eurasia. Language Resources and Evaluation, Vol. 54, No. 1, pp. 273–301. DOI: 10.1007/s10579-019-09480-6.

4. **Doerfer, G. (1976).** Proto-turkic: Reconstruction problems. Belleten-Türk Dili Araştırmaları Yıllığı, Vol. 23-24, No. 1975-1976, pp. 1–59.

5. **Golden, P. B. (2007).** Khazar studies: Achievements and perspectives. New Perspectives. Selected Papers from the Jerusalem 1999 International Khazar Colloquium, Brill, pp. 7–57. DOI: 10.1163/ej.9789004160422.i-460.7.

6. **Hardesty, L. (2010).** Computer automatically deciphers ancient language.

7. **Khamidullin, S. I. (2021).** Relations between the Bashkirs and the Volga Bulgars in the 10th–13th centuries. Ural Historical Journal, Vol. 71, No. 2, pp. 137–145. DOI: 10.30759/1728-9718-2021-2(71)-137-145.

8. **Khordadbeh, I. (1889).** The Book of Ways and Countries. Palmarium.

9. **Kumar, V. (2020).** Deciphering extinct ancient languages with machine learning. Analytics Insight.

10. **Leon, I. (2012).** The formation of the Volga Bulgaria: From tribe to state. St. Petersburg Slavic and Balkan Studies.

11. **Marjani Institute (2013).** Tatars' history.

12. **Mudrak, O. (2004).** Bulgar language.

13. **Mudrak, O. (2016).** Notes on the foreign language vocabulary of Khazar-Jewish documents. Space-2000 Moscow, Vol. 14, pp. 349–379.

14. **Papia, S. (2009).** Endangered languages: Some concerns. Economic and Political Weekly, Vol. 44, No. 32, pp. 17–19.

15. **Parker, E. (1895).** A thousand years of the tartars. Wentworth Press, 1st edition.

16. **Rashkovsky, B. (2014).** Khazars and judaism in the biblical commentaries of Yefet ben Ali. A New Medieval Jewish Source for Eastern European History, Vol. 1, No. 3, pp. 210–230. DOI: 10.14653/ju.2014.13.

17. **Reiss, C., Fox, A. (1996).** Linguistic reconstruction: An introduction to theory and method. Language, Vol. 72, No. 2, pp. 387. DOI: 10.2307/416657.

18. **Safarov, A., Gabrakhmanov, G., Galimova, E., Zagidullina, D., Izamaylov, I., Salikhova, A., Sitdikov, A., Shklyaeva, L. (2019).** Tatar language and written culture: From word to book. Tatar World, pp. 1–392.

19. **Salmin, A. (2013).** Ethnographical sources about the origin of the Chuvash. Japanese Slavic and East European Studies, Vol. 34, pp. 95–104. DOI: 10.5823/jsees.34.0_95.

20. **Salmin, A. (2015).** The Bulgarian language in the context of the history of the Chuvash. Bulletin of the Chuvash University.

21. **Sapargali, E., Akhmetov, I., Pak, A., Gelbukh, A. (2021).** Determining the relationship between the letters in the Voynich manuscript splitting the text into parts. Proceedings of the Mexican International Conference on Artificial Intelligence, Advances in Soft Computing, pp. 163–170. DOI: 10.1007/978-3-030-89820-5_13.

22. **Tatar, M. (2021).** Tatar alphabet: From ancient runes to modern Cyrillic. Billion Tatars.

23. **Yilmaz, E. (2002).** Chuvash and chuvash language. The Turks.

24. **Yule, G. (2010).** The study of language. Cambridge University Press.

Article received on 20/07/2023; accepted on 02/12/2023.
** Corresponding author is Alexander Gelbukh.*

ISSN 2007-9737

Un modelo de programación entera para la generación de horarios universitarios: Un caso de estudio

Luis E. Urbán-Rivero*,[1], Myrna H. Lezama-León[2],
Eduardo Cruz-Aldana[2], Noé D. Mares-Ortega[3],
Luisa F. Loera-Díaz[3]

[1] Universidad Autónoma Metropolitana Azcapotzalco,
Departamento de Sistemas,
México

[2] Universidad Politécnica Metropolitana de Hidalgo,
México

[3] Centro de Investigación en Matemáticas,
Unidad Aguascalientes,
México

leur@azc.uam.mx, {mlezama, ealdana}@upmh.edu.mx,
{noe.mares, luisa.loera}@cimat.mx

Resumen. En este trabajo se presenta un modelo de programación entera, específicamente de programación entera binaria para la asignación de profesor a curso y horario, para un caso de estudio en una universidad mexicana siguiendo los requerimientos de operación de la institución. Se sabe que esta actividad requiere de un tiempo considerable para su desarrollo debido a distintos factores (internos como externos a la institución). En este artículo se propone un modelo en dos etapas para la generación de horarios dentro de un programa de estudios especifico. Este modelo se aplico usando datos reales de la universidad y los resultados fueron comparados con los requisitos solicitados por la universidad. Se obtuvieron horarios en menos tiempo que el implicado en una asignación manual.

Palabras clave. Generación de horarios, investigación de operaciones, programación entera.

An Integer Programming Model for University Timetable Generation: A Case Study

Abstract. This article shows an integer programming model, specially a binary programming model for teacher-course-schedule assignment, in a case study of a mexican university for schedule planning (timetabling), according to organizational requeriments of the institution, because this activity demands a high investment of time for its development, since there are various factors (internal and external to the institution) that must be met. This article proposes a two-stage optimization model for the allocation of schedules for a specific educational program. This model is applied to a course and its results are compared with the requested requirements, obtaining the model results in less time than the obtained by manual assignment, in addition to complying with the restrictions established by the institution, as well as with the requirements of the teachers.

Keywords. Timetabling, operations research, MILP.

1. Introducción

Uno de los problemas más ampliamente estudiados en el área de la optimización combinatoria es el conocido como Timetabling (Planificación de horarios) [25, 3, 32] . En dicho problema se pretende hacer la planeación de los cursos que se imparten en una institución

doi: 10.13053/CyS-28-1-4610

educativa durante un periodo lectivo, incluyendo detalles como el horario de impartición de cada curso, el profesor que imparte el curso y el salón donde se llevara a cabo dicho curso [25, 3]. La planeación depende ampliamente de la organización que tiene cada institución educativa.

Dependiendo del nivel educativo, los problemas de organización pueden ser distintos, por ejemplo en [8, 24, 26] se muestran casos particulares de instituciones de educación media superior en Estados Unidos y Dinamarca respectivamente, mientras que en [10, 20, 4, 7, 17, 21, 1, 18, 5, 27] se muestran casos de instituciones de educación superior (IES).

Existen también distintos esquemas de organización de los cursos a impartir dependiendo de las decisiones que se toman por las instituciones educativas sobre como gestionar la demanda de los planes de estudio en dichas instituciones. La siguiente lista muestra algunas de las características consideradas en la planeación de los cursos:

1. Disponibilidad infraestructura física.
2. Disponibilidad del capital humano (profesores).
3. Preferencias y capacidad de los profesores para la participación de los cursos.
4. Subdivisión de los cursos en sesiones.
5. Horario laboral.
6. Gestión de la demanda.
7. Forma de impartir la clase.
8. Reglas laborales.
9. Preferencias de los alumnos.

Cabe destacar que los puntos del 7 al 9 pudiesen tener características de operación heterogéneas lo que complica su generalización, razón por la cuál se pueden observar múltiples trabajos en la literatura, donde cada uno aborda las características especificas de una institución en particular [3].

En el caso de la gestión de la demanda, esta depende ampliamente de las características especificas de los planes de estudio, la forma de inscribirse de los alumnos y si existen cursos comunes entre planes de estudio.

En este sentido, tanto en instituciones de educación media superior como en instituciones de educación superior estadounidenses o europeas se tienen modelos homogéneos de la gestión de la demanda, donde algunos cursos tienen claves comunes, no solo a nivel Institucional, si no incluso a nivel nacional, esto para facilitar el intercambio o la transferencia de créditos.

En cuanto a la forma de que se imparte la clase también existe heterogeneidad ya que mientras lo común es que un curso sea impartido por un solo profesor, existen instituciones donde algunos cursos como laboratorios o talleres son impartidos por hasta tres profesores de manera simultánea o diferida.

Por último, las reglas laborales son las que hacen la mayor diferencia entre casos de estudio, ya que pueden variar, entre instituciones, sistemas educativos, estados y países. Por lo que se tiene que modelar y resolver la situación particular.

Es claro que algunas universidades mexicanas tienen reglas de operación comunes o fácilmente generalizables [3], pero dados los tres aspectos heterogéneos antes mencionados es necesario adaptar los modelos a las condiciones de las IES de México.

En [10, 20, 4, 21, 1] se presentan casos de estudio en IES de México que incluyen reglas de operación específicas de Universidades Estatales con un sistema de Facultades, Universidades Privadas, Universidades Politécnicas e Institutos Tecnológicos.

El modelo que se presentará en este trabajo está basado en dichas referencias, tomando los aspectos que se tienen en común. La gestión de la demanda de los cursos es un elemento distintivo de una institución a otra. Es posible clasificar la forma en que se gestiona la demanda en IES de México en 4 diferentes formas:

- Demanda fija y no compartida.
- Demanda fija y compartida.
- Demanda flexible y no compartida.
- Demanda flexible y compartida.

La demanda fija se basa en lo que se conoce como los modelos basados en la estructura curricular como se puede observar en [5], donde se planean los cursos basándose en los distintos planes de estudio y asegurando que los alumnos regulares tengan garantizado su avance siempre que se cumplan los prerequisitos de los cursos.

En México se da el caso que las IES más grandes imparten todos los cursos obligatorios en los planes de estudio durante todos los periodos lectivos y se asegura que al menos una vez al año se impartan los cursos optativos.

Por otro lado, universidades de un tamaño más reducido, deben limitar su oferta debido a los limitados recursos físicos y de capital humano. A continuación describiremos cada uno de los mecanismos de gestión de demanda.

Demanda fija: La demanda está determinada por el plan de estudios, los alumnos regulares están obligados a llevar los cursos en el orden que se establece en el plan de estudios. Los alumnos irregulares se adaptan a la oferta generada por los alumnos regulares. En este esquema lo común es que la IES asigne al alumno su carga académica obligatoria sin tomar en cuenta sus preferencias por algún profesor u horario.

Demanda flexible: La demanda está determinada por el plan de estudios, los alumnos deciden el orden en que llevan las materias únicamente respetando los requisitos de cada curso. En este esquema lo común es que el alumno se inscriba en los cursos que desee, en el horario y con el profesor que prefiera, sólo evitando traslapes y respetando los prerequisitos de los cursos.

Demanda no compartida: Los cursos se imparten a los alumnos asociados a un plan de estudios en particular, es decir, si hay cursos en común con otro plan de estudios, estos no pueden tener alumnos de diferente plan de estudios.

Por ejemplo en donde los cursos podrían tener el mismo nombre pero el contenido y profundidad es diferente es necesario hacer dicha distinción como un Curso de Álgebra Lineal para Matemáticos y el mismo curso para Ingenieros, en apariencia son el mismo pero el contenido es diferente y podría provocar dificultades en los alumnos.

Demanda compartida: Los planes de estudio con cursos en común permiten que existan cursos con alumnos de diferentes planes de estudio. Por ejemplo, en una IES con sólo ingenierías se tiene como curso común Cálculo por lo que es una buena opcion que un curso de Cálculo pueda aceptar a alumnos de cualquier plan de estudios de ingeniería de dicha IES.

En el caso particular de México existen IES con los cuatro formas de gestión de demanda antes mencionadas. En este trabajo nos concentraremos en un caso de estudio que presenta el caso de demanda fija y no compartida.

Utilizaremos como base el modelo presentado en [1] que corresponde con el mismo esquema de gestión demanda y con el mismo sistema educativo, es decir, es una Universidad Politécnica de México (UPM).

Algunos elementos del modelo presentado en [1] serán retomados y otros particulares del caso de estudio se agregarán en nuestro modelo. Por otro lado, se debe destacar que en lo que se refiere a problemas de generación automática de horarios, existen dos tipos de requerimientos:

Requerimientos duros: Son aquellos que forzosamente se deben cumplir para que se considere un horario factible, por ejemplo, un curso debe tener asignado un profesor.

Requerimientos blandos: Son aquellos que se intentan cumplir en medida de lo posible, por ejemplo, las preferencias de cursos de los profesores.

Lo mas común para manejar los distintos requisitos de un horario, es que los requerimientos duros se modelen como restricciones, mientras que los requerimientos blandos, se modelen como parte de la función objetivo [3].

2. Características del problema

A continuación, se describen las características de la operación de las UPM:

- Los periodos lectivos, son cuatrimestrales.

- Cada plan de estudios está vinculado a un único departamento que se encarga de planear los horarios de dicho plan para cada periodo.

  Por lo tanto, la asignación de horarios para curso en una UPM, se puede realizar por plan de estudios de manera independiente (Demanda no comparttida).
- Cada curso es impartido por solo un profesor.
- Existen cursos especiales que se deben dar en laboratorios o talleres.
- Los salones estándar pueden tener distintas capacidades.
- Existen profesores de tiempo completo (PTC) que tienen un horario establecido y una carga académica mínima y una carga máxima.
- Existen profesores de Asignatura (PA) ellos establecen su disponibilidad y tienen una carga máxima y una carga mínima opcional.
- Los profesores dan preferencias sobre los cursos que pueden impartir basándose en su perfil.
- Un curso tiene un número de horas en las que se debe impartir en una semana.
- Un curso se divide en sesiones, las sesiones tienen una duración mínima y una duración máxima.
- La sesiones duran horas completas, no pueden ser interrumpidas e inician y finalizan en horas enteras. Por ejemplo de 7:00-9:00 es una sesión de 2 hrs valida.
- En un día puede haber a lo mas una sesión de un curso y las horas que conforman la sesión deben ser consecutivas.
- Un grupo es un conjunto de alumnos que acude a un conjunto de cursos establecidos por la coordinación de acuerdo al plan de estudios.

3. Caso de estudio

El presente trabajo se realizó con datos de la Universidad Politécnica Metropolitana de Hidalgo (UPMH), localizada en el municipio de Tolcayuca en el Estado de Hidalgo, México.

La UPMH pertenece al sistema UPM por lo que la demanda es fija y no compartida junto con las características previamente mencionadas parte del sistema de las UPM.

Adicional a las características mencionadas se tienen las siguientes situaciones especificas del caso de estudio que corresponde a la UPMH, ya que dicha universidad opera bajo el modelo Bilingüe Internacional y Sustentable (BIS), donde se priorizan las actividades relacionadas a un segundo idioma y el seguimiento de alumnos mediante actividades de tutoría.

Por tanto, las características de operación específicas son las siguientes:

- La coordinación de inglés (segundo idioma) reserva los horarios en los que sus profesores impartirán clase en cada grupo. Por lo que los horarios elegidos para los cursos de inglés de cada grupo no están disponibles para el resto de los cursos del grupo.
- Los PTC están en un horario laboral de 8 horas continuas.
- Tanto los PA como los PTC acuerdan con la coordinación el número mínimo y máximo de horas frente a grupo.
- Los PTC y PA seleccionados deben tener una hora para tutoría grupal y una hora para tutoría individual con al menos 1 grupo y no mas de dos grupos.
- Los cursos del turno matutino se imparten de las 7:00 hrs a las 18:00 hrs.
- Los cursos del turno vespertino se imparten de las 12:00 hrs a las 21:00 hrs.
- El turno de cada grupo es establecido por la coordinación de acuerdo con el avance general del grupo y las actividades adicionales a realizar.

4. Metodología

Realizar este tipo de planeaciones es un tema ampliamente estudiado. En [13] se puede encontrar un resumen reciente de los avances en metodologías metaheurísticas y como estas han ayudado en algunos casos de estudio en concreto.

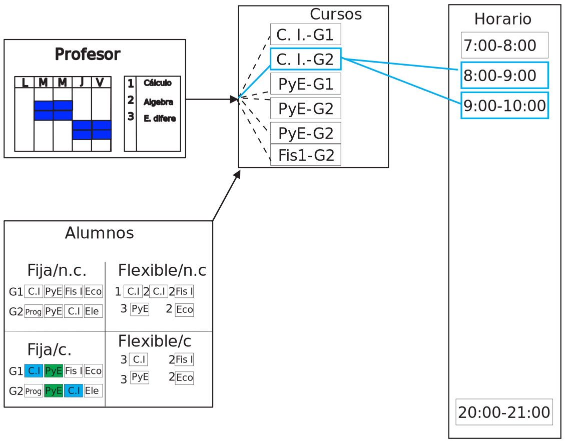


Fig. 1. Asignación de profesor-curso-horario. Se muestra cómo la demanda de cursos se genera de acuerdo a como se gestione la misma

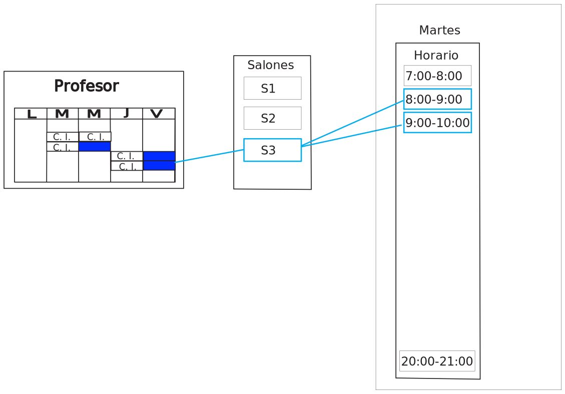


Fig. 2. Asignación de salón. En este caso el profesor ya tiene una carga horaria y se le asigna el salón de acuerdo a criterios de capacidad y preferencia

En [23, 31, 12, 19, 22], se pueden ver casos específicos, resueltos con técnicas metaheurísticas. Por último, en [29] se da un análisis profundo de como afecta en el desempeño de los modelos el dividir un problema de asignación multi-dimensional en asignaciones parciales tanto en la optimalidad como en el tiempo de ejecución.

En este trabajo se optó por la modelación programación entera y su resolución con un solver comercial, para el tamaño de las instancias, dicha opción fue suficiente. Se desarrollaron dos modelos de programación entera.

Dadas las características de la instancia se hicieron unas modificaciones al modelo propuesto en [1]. En este caso se dividió el problema en dos etapas. En la primera etapa se realizará una asignación de profesor-curso-horario como se muestra en la Figura 1. La asignación de salón se hará en una segunda etapa de manera similar a como se realiza en [28] como se observa en la Figura 2.

5. Modelo de programación binaria para la asignación profesor-curso-horario

Variables

$$x_{i,j}^{t,d} = \begin{cases} 1, & \text{Si el profesor } i \text{ es asignado al curso } j \text{ en la hora } t \text{ del día } d. \\ 0, & \text{En otro caso.} \end{cases}$$

$$y_{i,j} = \begin{cases} 1, & \text{Si el profesor } i \text{ es asignado al curso } j. \\ 0, & \text{En otro caso.} \end{cases}$$

$$z_{j,d} = \begin{cases} 1, & \text{Si el curso } j \text{ se imparte el día } d. \\ 0, & \text{En otro caso.} \end{cases}$$

$$s_{j,t,d} = \begin{cases} 1, & \text{Si el curso } j \text{ comienza a la hora } t \text{ en el día } d. \\ 0, & \text{En otro caso.} \end{cases}$$

Parámetros

P : Conjunto de profesores.

C : Conjunto de cursos.

T : Conjunto de carriles horarios.

D : Conjunto de días.

G : Conjunto de grupos.

$C(g)$: Cursos del grupo g, $g \in G$.

C_{fijo} : Cursos con horario fijo, $C_{\text{fijo}} \subset C$.

P_{TC} : Profesores de tiempo completo.

P_A : Profesores de asignatura.

P_{Tut} : Profesores que imparten tutorías.

P_{ing} : Profesores de inglés.

P_F : Profesores ficticios.

T_{am} : Carriles horarios de las 7:00 a las 18:00 hrs.

T_{pm} : Carriles horarios de las 12:00 a las 21:00 hrs.

$\tau(j)$: Horas a la semana del curso j.

P : $P_{\text{TC}} \cup P_A \cup P_{\text{ing}} \cup P_F$.

$P_{\text{Tut}} \subseteq$: $P_{\text{TC}} \cup P_A \cup P_F$.

$\gamma_{\text{mín}}(i)$: Horas mínimas que puede impartir el profesor i.

$\gamma_{\text{máx}}(i)$: Horas máximas que puede impartir el profesor i.

$\psi_{\text{mín}}(j)$: Duración mínima de la sesión del curso j.

$\psi_{\text{máx}}(j)$: Duración máxima de la sesión del curso j.

$$\beta\,_i^{t,d} = \begin{cases} 1, & \text{Si el profesor } i \text{ esta disponible} \\ & \text{en la hora } t \text{ el día } d. \\ 0, & \text{En otro caso.} \end{cases}$$

$$H_j^{t,d} = \begin{cases} 1, & \text{Si el curso fijo } j \text{ esta programado} \\ & \text{en la hora } t \text{ el día } d. \\ 0, & \text{En otro caso.} \end{cases}$$

$$\alpha : P \times C \to \mathbb{Z}^+, \tag{1}$$

donde:

$\alpha(i, j) =$ Preferencia del profesor i por el curso j.

Se tienen tres objetivos. El primero es asignar en medida de lo posible un horario al profesor dentro de su disponibilidad. Por lo tanto minimizamos la suma de las diferencias entre lo que solicitan los profesores y los horarios en los que se les asignan los cursos:

$$f_1 = \sum_{i \in P} \sum_{j \in C} \sum_{t \in T} \sum_{d \in D} (1 - \beta_i^{t,d}) \cdot x_{i,j}^{t,d}. \tag{2}$$

Por otro lado se busca asignarle al profesor el curso que prefiera de una lista de preferencias previamente otorgadas, donde la preferencia mas alta tiene el valor mas bajo:

$$f_2 = \sum_{i \in P} \sum_{j \in C} \alpha(i, j) \cdot y_{i,j}. \tag{3}$$

Para finalizar la función objetivo se agrega un conjunto de profesores ficticios que pueden impartir cualquier asignatura y que tienen disponibilidad completa, pero que su uso genera un costo alto. Esto último evita que el problema sea infactible, es claro que se deben agregar tantos profesores ficticios como sea necesario:

$$f_3 = \sum_{i \in P_F} \sum_{j \in C} y_{i,j}. \tag{4}$$

Con lo que la función objetivo global queda como sigue:

$$\text{mín } w = f_1 + f_2 + M f_3, \tag{5}$$

donde M es una constante con un valor muy grande. El profesor i puede atender a lo mas un curso durante la hora t en el día d:

$$\sum_{j \in C} x_{i,j}^{t,d} \leq 1, \;\; i \in P, \;\; t \in T, \;\; d \in D. \tag{6}$$

El profesor i debe impartir al menos $\gamma_{\text{min}}(i)$ horas a la semana y no mas de $\gamma_{\text{max}}(i)$ horas a la semana:

$$\gamma_{\text{mín}}(i) \leq \sum_{j \in C} \sum_{t \in T} \sum_{d \in D} x_{i,j}^{t,d} \leq \gamma_{\text{máx}}(i), \;\; i \in P. \tag{7}$$

El curso j debe tener asignado un profesor:

$$\sum_{i \in P} y_{i,j} = 1, \;\; j \in C. \tag{8}$$

El profesor i debe impartir todas las horas del curso j, si el es seleccionado para impartirlo:

$$\sum_{t \in T} \sum_{d \in D} x_{i,j}^{t,d} = \tau(j) \cdot y_{i,j} \;, \;\;\; i \in P, \;\; j \in C. \tag{9}$$

Las sesiones del curso j deben tener una duración de al menos $\psi_{\text{mín}}(j)$ hrs y no mas de $\psi_{\text{máx}}(j)$ hrs:

$$\begin{gathered}\psi_{\text{mín}}(j)\cdot z_{j,d} \le \sum_{i\in P}\sum_{t\in T} x^{t,d}_{i,j} \le \psi_{\text{máx}}(j)\cdot z_{j,d},\\ j\in C,\, d\in D.\end{gathered} \tag{10}$$

Para lograr que una sesión de un curso tenga horas consecutivas, sin tiempos muertos, se necesitan las restricciones (10) y (11) originalmente propuestas por [2] en otro contexto pero se pueden adaptar a nuestras necesidades. El curso i sólo puede tener un inicio el día d:

$$\sum_{t\in T} s_{j,t,d} \le z_{j,d},\ j\in C, d\in D, \tag{11}$$

$$\begin{gathered}s_{j,t,d} \ge x^{t,d}_{i,j} - x^{t-1,d}_{i,j},\\ i\in P,\ j\in C,\ t\in T\backslash 1,\ d\in D.\end{gathered} \tag{12}$$

Las restricciones (13–22) corresponden con las condiciones particulares del caso de estudio de la UPMH. Un grupo g puede tener a lo mas un curso con un profesor i, es decir, un profesor no puede impartir dos o mas cursos a un mismo grupo:

$$\sum_{q\in C(g)} y_{i,q} \le 1,\ i\in P,\ g\in G. \tag{13}$$

Los cursos de un grupo g no pueden programarse de manera simultánea:

$$\sum_{q\in C(g)} x^{t,d}_{i,q} \le 1,\ i\in P,\ g\in G,\ t\in T,\ d\in D. \tag{14}$$

Los grupos del turno matutino toman clases de 7:00 a 18:00 hrs:

$$x_{i,j,t,d} \le 0,\ i\in P,\ j\in C,\ t\in T_{\text{pm}},\ d\in D. \tag{15}$$

Los grupos del turno vespertino toman clases de 12:00 a 21:00 hrs:

$$x_{i,j,t,d} \le 0,\ i\in P,\ j\in C,\ t\in T_{\text{am}},\ d\in D. \tag{16}$$

Los cursos no fijos de un grupo g ($C(g)\backslash C_{\text{fijo}}(g)$) no pueden programarse en las horas reservadas para los cursos fijos de ese grupo.

En nuestro caso particular los únicos cursos fijos son los cursos de inglés que son determinados previamente por la coordinación correspondiente y enviados como restricción a la coordinación de cada carrera. Los profesores de los cursos fijos son provistos por otra coordinación, en nuestro caso los representamos mediante el conjunto P_{ing}:

$$\begin{gathered}x_{i,q,t,d} \le 0,\ i\in P\backslash P_{\text{ing}},\ q\in C(g)\backslash C_{\text{fijo}}(g),\\ (t,d)|H^{t,d}_j = 1, j\in P_{\text{ing}},\ t\in T,\ d\in D.\end{gathered} \tag{17}$$

Por otro lado, debemos asegurar que los profesores de los cursos fijos no sean asignados a cursos no fijos:

$$\begin{gathered}x_{i,q,t,d} \le 0,\ i\in P_{\text{ing}},\ q\in C\backslash C_{\text{fijo}},\\ (t,d)|H^{t,d}_j = 0, j\in P_{\text{ing}},\ t\in T,\ d\in D.\end{gathered} \tag{18}$$

Es necesario asegurar que los profesores de los cursos fijos (P_{ing}) se asignen a cursos fijos solamente. Esto se puede hacer dos maneras. Se puede colocar costos altos a la asignación de profesores de cursos fijos con cursos no fijos, o bien, se puede asignar directamente a los profesores de cursos fijos solamente a dichoscursos. Para asegurar la factibilidad se crean tantos profesores como cursos fijos existan.

Además de que se le da una disponibilidad total a estos profesores y que su única preferencia sean los cursos fijos dado que estos profesores son gestionados de manera externa. Un curso fijo j debe tener asignado un profesor del conjunto de profesores para cursos fijos (P_{ing}) en el horario $(t,d)|H^{t,d}_j = 1$:

$$\sum_{i\in P_{\text{ing}}} x_{i,j,t,d} = 1,\ j\in C_{\text{fijo}},\ (t,d)|H^{t,d}_j = 1. \tag{19}$$

Un profesor i de cursos fijos puede tener a lo mas un curso fijo asignado en el horario $(t,d)|H^{t,d}_j = 1$:

$$\sum_{j\in C_{\text{fijo}}} x_{i,j,t,d} \le 1,\ i\in P_{\text{ing}},\ (t,d)|H^{t,d}_j = 1. \tag{20}$$

El grupo de profesores para dar tutoría debe dar al menos una hora de tutoría grupal y un hora de tutoría individual con el mismo grupo y no se le pueden asignar más de dos horas de tutoría (grupal e individual) a estos profesores. Para este propósito las horas de tutoría se agregarán como un curso de tutoría grupal e individual de dos horas a la semana ($C_{\text{TGI}} \subset C \backslash C_{\text{fijo}}$) y con sesiones de duración mínima y máxima de una hora:

$$1 \leq \sum_{j \in C_{\text{TGI}}} y_{i,j} \leq 2, \; i \in P_{\text{Tut}}. \tag{21}$$

Por último, en una hora t no puede haber más de λ cursos simultáneos:

$$\sum_{i \in P} \sum_{j \in C} x_{i,j}^{t,d} \leq \lambda \;\; t \in T, \;\; d \in D. \tag{22}$$

Con el modelo de programación entera (2–22) se asegura una asignación de profesores a cursos basándose en su preferencia y disponibilidad de horario. Adicional a esto el modelo incluye las reglas de operación específicas de la universidad, por lo que su solución genera un horario factible o bien que requiere cambios mínimos.

Resultado de este modelo se podrán obtener, el horario global de todos los cursos, el horario de cada profesor y los horarios de los grupos. Dado que un grupo tiene al mismo conjunto de alumnos, bastaría con hacer la asignación por grupos y no por cursos a los salones. Por lo que el modelo para asignar grupo salón horario queda como sigue:

6. Modelo de programación binaria para la asignación de curso-salón-horario

Variables

$$x_{i,j}^{t,d} = \begin{cases} 1, & \text{Si el curso } i \text{ es asignado al salón } j \\ & \text{en la hora } t \text{ del día } d. \\ 0, & \text{En otro caso.} \end{cases}$$

$$x_{i,j}^{t,d} = \begin{cases} 1, & \text{Si el curso } i \text{ es asignado al salón } j \\ & \text{en el día } d. \\ 0, & \text{En otro caso.} \end{cases}$$

$$y_{i,j} \;\; = \begin{cases} 1, & \text{Si el salón } j \text{ es usado en el día } d. \\ 0, & \text{En otro caso.} \end{cases}$$

donde:

$$r_d = \text{Número de salones utilizados en el día } d.$$

Parámetros

C	: Conjunto de cursos.
S	: Conjunto de salones.
G	: Conjunto de grupos.
T	: Conjunto de carriles horarios.
D	: Conjunto de días.
$\mathrm{CT}(t,d)$	: Cursos que se traslapan en la hora t del día d.
$C(g)$	: Cursos del grupo g.
$\lvert\mathcal{H}_i^d\rvert$	: Número de horas que se imparten del curso i el día d.
$\rho(g,s)$	: Factor de utilización del grupo g en el salón s.
$\phi(c,s)$	: Preferencia del salón s para el curso c.

$$\mathcal{H}_i^{t,d} = \begin{cases} 1, & \text{Si el curso } i \text{ esta programado en la} \\ & \text{hora } t \text{ el día } d. \\ 0, & \text{En otro caso.} \end{cases}$$

En este caso se tienen cuatro objetivos que se mencionan a continuación:

1. Maximizar la permanencia de un grupo en un salón ($\mathcal{F}_1$).
2. Minimizar el número de salones utilizados por día ($\mathcal{F}_2$).
3. Asegurar que el salón asignado tenga el tamaño adecuado para el tamaño del grupo ($\mathcal{F}_3$).
4. Asegurar que se asignen los salones de acuerdo a las preferencias que da la coordinación ($\mathcal{F}_4$).

Tabla 1. Resultados de la asignación profesor-curso-horario

Profesor	Gurobi		CPLEX		MOSEK		HIGHS	
	P_H	P_C	P_H	P_C	P_H	P_C	P_H	P_C
1	1.00	1.00	1.00	1.00	Infactible		Infactible	
2	1.00	1.00	1.00	1.00				
3	1.00	1.00	1.00	1.00				
4	1.00	1.00	0.95	1.00				
5	1.00	1.00	1.00	1.00				
6	1.00	1.00	1.00	1.00				
7	1.00	1.00	0.96	1.00				
8	1.00	1.00	1.00	1.00				
9	1.00	1.00	1.00	1.00				
10	0.90	1.00	0.90	1.00				
11	1.00	1.00	1.00	1.00				
12	0.93	1.00	0.93	1.00				
13	1.00	1.00	1.00	1.00				
14	1.00	1.00	1.00	1.00				
15	0.92	1.00	1.00	1.00				
16	1.00	1.00	1.00	1.00				
17	1.00	1.00	1.00	1.00				
Total	**16.75**	**17.00**	**16.74**	**17.00**				
Proporción	**0.99**	**1.00**	**0.98**	**1.00**				
Tiempo (s)	**266**		**3600**		**3600**		**3600**	

donde $\mathcal{F}_1$ se define como a continuación:

$$\mathcal{F}_1 = \sum_{g\in G} \sum_{j\in S} \sum_{p,q\in C(g),p\neq q} y_{p,j}^d \cdot y_{q,j}^d. \tag{23}$$

En (23) se busca que los cursos del mismo grupo se lleven en el mismo salón en la medida de lo posible, por lo que equivalente a minimizar los cambios de salón, es maximizar el número de cursos en un mismo salón, es necesario linealizar este objetivo, el procedimiento es sencillo y se puede encontrar en [14] o bien se puede incluir directamente en Gurobi [9]. Para el objetivo (2), se define $\mathcal{F}_2$ como sigue:

$$\mathcal{F}_2 = \sum_{d\in D} r_d. \tag{24}$$

Para el caso del objetivo (3) es necesario definir el factor de utilización como:

$$\rho(g,s) = \frac{|g|}{|s|}, \tag{25}$$

donde $|g|$ es el número de alumnos inscritos en el grupo g, mientras que $|s|$ es el cupo máximo del salón s. Por lo que $\mathcal{F}_3$ queda como sigue:

$$\mathcal{F}_3 = \sum_{g\in G} \sum_{i\in C(g)} \sum_{j\in S} \sum_{d\in D} \rho(g,j) \cdot y_{i,j}^d. \tag{26}$$

Por último se debe asegurar que se da preferencia a los salones especificos para los cursos que asi lo necesiten, como por ejemplo los cursos que requieren un laboratorio o taller en especial:

$$\mathcal{F}_4 = \sum_{i\in C} \sum_{j\in S} \sum_{d\in D} \phi(i,j) \cdot y_{i,j}^d. \tag{27}$$

Por lo tanto la función objetivo global queda como sigue:

$$\text{mín}\, z = -A_1 \cdot \mathcal{F}_1 + A_2 \cdot \mathcal{F}_2 + A_3 \cdot \mathcal{F}_3 + A_4 \cdot \mathcal{F}_4, \tag{28}$$

donde A_1, A_2, A_3 y A_4 son constantes positivas. En cuanto a las restricciones se tienen las que se enuncian enseguida. Un curso i debe tener asignado un salón en una hora t el día d:

$$\sum_{j\in S} x_{i,j}^{t,d} = 1,\ i\in G,\ t\in T,\ d\in D. \tag{29}$$

Un salón j puede albergar a lo más un curso durante una hora t, en el día d:

$$\sum_{i\in G} x_{i,j}^{t,d} \leq 1,\ j\in S,\ t\in T,\ d\in D. \tag{30}$$

Cursos que se traslapan en la hora t del dia d, a lo más uno puede ocupar el salón j:

$$\sum_{i\in CT(t,d)} y_{i,j}^d \leq 1,\ j\in S. \tag{31}$$

Se deben impartir todas las horas de una sesión del día d en el mismo salón:

$$\sum_{t\in\mathcal{H}_i^{t,d}=1} x_{i,j}^{t,d} = |\mathcal{H}_i^d| \cdot y_{i,j}^d.\ i\in C,\ j\in S,\ d\in D. \tag{32}$$

Tabla 2. Resultados de la asignación de salones

Día	**Gurobi**		**CPLEX**		**MOSEK**		**HIGHS**	
	P_T	P_S	P_T	P_S	P_T	P_S	P_T	P_S
Lunes	0.83	0.85	0.83	0.87	0.83	0.85	0.83	0.85
Martes	0.90	0.98	0.90	0.98	0.90	0.98	0.90	0.98
Miércoles	0.85	0.87	0.85	0.87	0.85	0.87	0.85	0.87
Jueves	0.88	0.93	0.88	0.93	0.88	0.93	0.88	0.93
Viernes	0.87	0.95	0.87	0.95	0.87	0.95	0.87	0.95
Tiempo (s)	**0.85**		**1.33**		**18.73**		**2.61**	

En un día d no se pueden usar más de r_d salones:

$$\sum_{j\in S} w_{j,d} \leq r_d \ \ d \in D. \tag{33}$$

7. Resultados

Se resolvió una instancia con datos del cuatrimestre 2022-3 de septiembre a diciembre de 2022 usando Python [30], la biblioteca PULP [15] y los solvers Gurobi 10.0 [9]. CPLEX 22.1[6], MOSEK 10.1 [16] y HIGHS 1.6.0 [11] en un equipo Con CPU Intel Core i5 a 3.40GHz con 16 GB de RAM.

La instancia consta de la programación de 75 cursos de los cuales 10 tienen horario fijo (cursos de inglés), repartidos entre 10 grupos (cinco matutinos y cinco vespertinos), tomando en cuenta la disponibilidad y preferencias de de 17 profesores, lo cuál genera un modelo con 784,354 restricciones y 183,450 variables enteras.

Para medir los resultados de dicha instancia en la etapa de de asignación profesor-curso-horario, se utilizaron los indicadores P_H qué es la proporción de horario asignado respecto al horario solicitado por un profesor. El indicador P_C es la proporción de cursos asignados dentro las preferencias de un profesor.

En la Tabla 1 Se muestran los resultados de la instancia, también se muestra la suma de la proporciones para P_H y P_C, además de la proporción total de profesores y el tiempo de ejecución en segundos.

Como se puede observar se cubrió casi en su totalidad la demanda de los cursos dentro de la disponibilidad de los profesores salvo en cuatro casos donde se pudo llegar a un acuerdo con ellos o el cambio se hizo de manera manual.

En cuanto a las preferencias, a todos los profesores se les asignaron cursos dentro de su lista proporcionada y no se asigno a nadie alguna materia que estuviera fuera de sus aptitudes.

Por último, cabe destacar que en los resultados de la instancia se hizo necesario que se contratará a un profesor adicional para impartir dos cursos que no pudieron ser asignados a ningún profesor de la plantilla laboral actual.

Cabe destacar que los solvers MOSEK y HIGHS no pudieron encontrar una solución factible en un tiempo máximo de 3600 segundos. Por otro lado, en la segunda etapa.

La instancia cuenta con 75 cursos con horario y profesor establecido que debieron programarse en siete salones y dos laboratorios. El modelo resuelto de esta segunda etapa se resolvió con las mismas herramientas que en la primera etapa, el modelo resultante cuenta con 74,540 restricciones y 72,726 variables enteras.

En la Tabla 2. Se muestran por día los indicadores P_T que representa la proporción de cursos que fueron asignados al salón del tamaño apropiado, tomando en cuenta que cada curso está asociado a un grupo y dicho grupo tiene una cantidad de alumnos establecida. En este caso se dejaron fuera los cursos asignados a los laboratorios ya que se tiene que asignar el curso a dicho laboratorio a pesar de que el tamaño no sea el adecuado.

Por otro lado P_S representa la proporción de cursos que fueron asignados a salones dentro de su lista de preferencias. En este caso los valores de las A_k, $k = 1, 2, 3, 4$ se ajustaron de manera experimental siguiendo las necesidades del tomador de decisiones. Los archivos con las instancias se pueden encontrar en la página [1] en formato csv. Cabe destacar que todos los solvers pudieron resolver a optimalidad la instancia pero en tiempo distintos.

[1] github.com/lurbanrivero/tt_upmh

8. Conclusiones y trabajo futuro

Se logró construir dos modelos de programación lineal entera que son capaces de generar horarios escolares en el caso particular de la UPMH. Cabe destacar que aunque se resolvió un caso particular, algunas características son generalizables a todo el sistema de UPM y sólo se requieren ajustes específicos para el caso en particular. El proceso de manera normal se llevaba a cabo de manera manual por la coordinación de cada carrera. Dicho proceso es susceptible a errores humanos e inconformidades, además de que requiere de dos a tres semanas de trabajo para consolidar el horario.

Con los horarios generados por está metodología se tiene una propuesta que requiere ajustes mínimos y permite consolidar una propuesta de horario en menos de una semana.

Esto último ya sin la generación manual. Cabe resaltar que existe una limitación para esta metodología que es que no permite cambios puntuales y mantener toda la demás asignación, de hacer esto último se podría generar un horario completamente diferente.

La metodología generada en este trabajo está limitada a un esquema de demanda. Una dirección de investigación sería explorar otro modelo de demanda, generar un modelo y aplicarlo en un caso de estudio donde se puedan mostrar las ventajas de generar horarios de manera automatizada y adecuados a cada caso.

Referencias

1. **Arratia-Martinez, N. M., Maya-Padron, C., Avila-Torres, P. A. (2021).** University course timetabling problem with professor assignment. Mathematical Problems in Engineering, Vol. 2021, pp. 1–9. DOI: 10.1155/2021/6617177.

2. **Bixby, E. R., Fenelon, M., Gu, Z., Rothberg, E., Wunderling, R. (2000).** MIP: Theory and practice — closing the gap. International Federation of Information Processing. Advances in Information and Communication Technology, pp. 19–49. DOI: 10.1007/978-0-387-35514-6_2.

3. **Chen, M. C., Sze, S. N., Goh, S. L., Sabar, N. R., Kendall, G. (2021).** A survey of university course timetabling problem: Perspectives, trends and opportunities. Vol. 9, pp. 106515–106529. DOI: 10.1109/access.2021.3100613.

4. **Chávez-Bosquez, O., Hernández-Torruco, J., Hernández-Ocaña, B., Canul-Reich, J. (2020).** Modeling and solving a Latin American university course timetabling problem instance. Mathematics, Vol. 8, No. 10, pp. 1833. DOI: 10.3390/math8101833.

5. **Colajanni, G., Daniele, P. (2020).** A new model for curriculum-based university course timetabling. Optimization Letters, Vol. 15, No. 5, pp. 1601–1616. DOI: 10.1007/s11590-020-01588-x.

6. **CPLEX, IBM ILOG (2022).** V22. 1: User's manual for CPLEX. International Business Machines Corporation, Vol. 46, No. 53, pp. 157.

7. **Cruz-Chávez, M. A., Flores-Pichardo, M., Martínez-Oropeza, A., Moreno-Bernal, P., Cruz-Rosales, M. H. (2016).** Solving a real constraint satisfaction model for the university course timetabling problem: A case study. Mathematical Problems in Engineering, Vol. 2016, pp. 1–14. DOI: 10.1155/2016/7194864.

8. **Dorneles, A. P., de Araújo, O. C. B., Buriol, L. S. (2014).** A fix-and-optimize heuristic for the high school timetabling problem. Computers and Operations Research, Vol. 52, pp. 29–38. DOI: 10.1016/j.cor.2014.06.023.

9. **Gurobi Optimization, LLC (2023).** Gurobi optimizer reference manual.

10. **Hernández-Vázquez, J. I., Hernández-González, S., Baltazar-Flores, M. R., Jiménez-García, J. A., Hernández-Vázquez, J. O. (2020).** Programación matemática binaria por etapas en la elaboración de un horario universitario. Entreciencias: Diálogos en la Sociedad del Conocimiento, Vol. 8, No. 22. DOI: 10.22201/enesl.20078064e.2020.22.70018.

11. **Huangfu, Q., Hall, J. A. J. (2017).** Parallelizing the dual revised simplex method. Mathematical Programming Computation, Vol. 10, No. 1, pp. 119–142. DOI: 10.1007/s12532-017-0130-5.

12. **Iqbal, Z., Ilyas, R., Chan, H. Y., Ahmed, N. (2021).** Effective solution of university course timetabling using particle swarm optimizer based hyper heuristic approach. Baghdad Science Journal, Vol. 18, No. 4, pp. 1465. DOI: 10.21123/bsj.2021.18.4(suppl.).1465.

13. **Kaur, M., Saini, S. (2020).** A review of metaheuristic techniques for solving university course timetabling problem. Lecture Notes in Networks and Systems, pp. 19–25. DOI: 10.1007/978-981-15-5421-6_3.

14. **MirHassani, S. A., Hooshmand, F. (2019).** Methods and models in mathematical programming. Springer. DOI: 10.1007/978-3-030-27045-2.

15. **Mitchell, S., O'Sullivan, M., Dunning, I. (2011).** PuLP: A linear programming toolkit for python. The University of Auckland, Auckland, New Zealand, Vol. 65.

16. **MOSEK ApS (2023).** The mosek optimization toolbox for MATLAB manual. Version 10.1.

17. **Prabodanie, R. A. R. (2017).** An integer programming model for a complex university timetabling problem: A case study. Industrial Engineering and Management Systems, Vol. 16, No. 1, pp. 141–153. DOI: 10.7232/iems.2017.16.1.141.

18. **Pratiwi, M., Rosyidi, C. N., Yuniaristanto (2021).** An optimization model for course scheduling in undergraduate industrial engineering program of universitas sebelas maret. Institute of Physics Conference Series: Materials Science and Engineering, Vol. 1072, No. 1, pp. 012008. DOI: 10.1088/1757-899x/1072/1/012008.

19. **Sakaliuk, O., Trishyn, F. (2021).** Using a genetic algorithm to solve the courses timetabling creation problem. Automation of Technological and Business Processes, Vol. 13, No. 2, pp. 22–28. DOI: 10.15673/atbp.v13i2.2053.

20. **Sánchez-Partida, D., Baquela, E. G., Mora-Vargas, J., Smith, N. R. (2016).** Case study: Simulated annealing for improving the educational timetable. Nova Scientia, Vol. 8, No. 17, pp. 340–360.

21. **Sánchez-Partida, D., Martínez-Flores, J. L., Olivares-Benítez, E. (2014).** An integer linear programming model for a university timetabling problem considering time windows and consecutive periods. Journal of Applied Operational Research, Vol. 6, No. 3, pp. 158–173.

22. **Song, T., Liu, S., Tang, X., Peng, X., Chen, M. (2018).** An iterated local search algorithm for the university course timetabling problem. Applied Soft Computing, Vol. 68, pp. 597–608. DOI: 10.1016/j.asoc.2018.04.034.

23. **Sylejmani, K., Gashi, E., Ymeri, A. (2022).** Simulated annealing with penalization for university course timetabling. Journal of Scheduling, Vol. 26, No. 5, pp. 497–517. DOI: 10.1007/s10951-022-00747-5.

24. **Sørensen, M., Dahms, F. H. W. (2014).** A two-stage decomposition of high school timetabling applied to cases in Denmark. Computers and Operations Research, Vol. 43, pp. 36–49. DOI: 10.1016/j.cor.2013.08.025.

25. **Tan, J. S., Goh, S. L., Kendall, G., Sabar, N. R. (2021).** A survey of the state-of-the-art of optimisation methodologies in school timetabling problems. Expert Systems with Applications, Vol. 165, pp. 113943. DOI: 10.1016/j.eswa.2020.113943.

26. **Tassopoulos, I. X., Iliopoulou, C. A., Beligiannis, G. N. (2019).** Solving the greek school timetabling problem by a mixed integer programming model. Journal of the Operational Research Society, Vol. 71, No. 1, pp. 117–132. DOI: 10.1080/01605682.2018.1557022.

27. **Tavakoli, M. M., Shirouyehzad, H., Lotfi, F. H., Najafi, S. E. (2020).** Proposing

a novel heuristic algorithm for university course timetabling problem with the quality of courses rendered approach: A case study. Alexandria Engineering Journal, Vol. 59, No. 5, pp. 3355–3367. DOI: 10.1016/j.aej.2020.05.004.

28. **Urbán-Rivero, L. E., Benítez-Escárcega, M. R., Alvarez, J. V. (2020).** An integer linear programming model for a case study in classroom assignment problem. Computación y Sistemas, Vol. 24, No. 1, pp. 97–104. DOI: 10.13053/cys-24-1-3191.

29. **Valencia, C. E., Alfaro, C. A., Zaragoza-Martinez, F. J., Vargas-Magaña, M. C., Perez-Perezt, S. L. (2018).** Outperforming several heuristics for the multidimensional assignment problem. 15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), pp. 1–6. DOI: 10.1109/ICEEE.2018.8533978.

30. **Van-Rossum, G., Drake, F. L. (2009).** Python 3 reference manual.

31. **Yusoff, M., Roslan, N. (2019).** Evaluation of genetic algorithm and hybrid genetic algorithm-hill climbing with elitist for lecturer university timetabling problem. Advances in Swarm Intelligence, pp. 363–373. DOI: 10.1007/978-3-030-26369-0_34.

32. **Zhu, K., Li, L. D., Li, M. (2021).** A survey of computational intelligence in educational timetabling. International Journal of Machine Learning and Computing, Vol. 11, No. 1, pp. 40–47. DOI: 10.18178/ijmlc.2021.11.1.1012.

Article received on 11/05/2023; accepted on 06/11/2023.
** Corresponding author is Luis E. Urbán-Rivero.*

ISSN 2007-9737

Adaptation of Transformer-Based Models for Depression Detection

Olaronke O. Adebanji, Olumide E. Ojo,
Hiram Calvo*, Irina Gelbukh, Grigori Sidorov

Instituto Politécnico Nacional, Centro de Investigación en Computación,
Mexico

{olaronke.oluwayemisi, olumideoea, ir.gelbukh}@gmail.com,
{hcalvo, sidorov}@cic.ipn.mx

Abstract. Pre-trained language models are able to capture a broad range of knowledge and language patterns in text and can be fine-tuned for specific tasks. In this paper, we focus on evaluating the effectiveness of various traditional machine learning and pre-trained language models in identifying depression through the analysis of text from social media. We examined different feature representations with the traditional machine learning models and explored the impact of pre-training on the transformer models and compared their performance. Using BoW, Word2Vec, and GloVe representations, the machine learning models with which we experimented achieved impressive accuracies in the task of detecting depression. However, pre-trained language models exhibited outstanding performance, consistently achieving high accuracy, precision, recall, and F1 scores of approximately 0.98 or higher.

Keywords. Depression, bag-of-words, word2vec, GloVe, machine learning, deep learning, transformers, sentiment analysis.

1 Introduction

Depression is a profound psychological disorder that affects people in different regions of the world, regardless of their gender, age, or social status [1].

It is a psychiatric condition characterized by persistent feelings of boredom, negativity, and sadness during daily activities. People experiencing depression often face challenges in interpersonal relationships, occupational performance, and maintaining healthy bonds, which ultimately affect their overall well-being. It is estimated that more than 280 million people worldwide struggle with depression, making it the main cause of disability on a global scale [6].

Despite its prevalence and impact, depression frequently evades detection and remains untreated. Early detection of depression presents challenges due to people who do not seek professional help or are unaware of their symptoms.

The presence of social media (SM) and online forums provides researchers with a unique opportunity to analyze online expressions of thoughts and emotions, potentially uncovering indications of depression.

Therefore, it is essential to develop accessible and efficient techniques that can help identify people at risk of depression, allowing them to receive the necessary support. SM platforms have become popular for communication and self-expression [8, 7, 47].

Sentiment analysis (SA) identifies and extracts subjective information from text, such as SM posts, reviews, and news articles. By analyzing the language used in SM texts [41, 42, 4, 31, 17, 2, 28], SA algorithms can determine the general sentiment of the text.

SM serves as a valuable tool to connect with people who might be susceptible to depression or who face challenges related to mental well-being. Considering the extensive use of SM platforms such as Facebook, Twitter, and Instagram by billions of people around the world to communicate and share information, it has evolved into an integral aspect of contemporary communication

doi: 10.13053/CyS-28-1-4691

methods. Studies have shown that people with depression use SM as a coping mechanism, seeking support and validation from others through online interactions [3, 9, 36].

Environmental and social factors, among others, have a great influence on the development of depression. The process of analyzing large volumes of text derived from SM using natural language processing (NLP) techniques allows the identification of language patterns that can indicate depression [24, 48, 52, 49].

Machine Learning (ML) techniques are revolutionizing the field of NLP [19, 26, 43, 30, 35, 10, 31, 32, 27]. These techniques have enabled researchers to build sophisticated models that can analyze and understand complex human language, including sentiment, syntax, and semantics. These algorithms use statistical rules to discover patterns in the data and use them to inform decisions as a result of this learning.

Deep learning (DL), a subset of ML, involves training artificial neural networks with many layers to recognize patterns and make decisions. Transformers are a type of DL architecture that has become popular for its ability to process sequential data, such as text.

Fine-tuned pre-trained language models are a specific application of these DL techniques. These models are pre-trained on massive volumes of text data before being fine-tuned for specific tasks such as named entity recognition or sentiment analysis.

By fine-tuning these models on a specific task, researchers can leverage the pre-existing knowledge encoded in the model and achieve state-of-the-art performance on the task at hand.

In this paper, we performed various experiments to test the efficiency of traditional ML and DL techniques, including fine-tuned pre-trained transformer models, for the detection of depression in social media texts.

We conducted an in-depth investigation of these models and examined their performance. Our goal is to provide information on how well these models can detect depression and highlight areas for future research and development.

The findings of our research improved our understanding of the potential use of these sentiment analysis techniques to detect depression and inform the development of targeted interventions that can reduce the burden of depression on society as a whole. Our research contributes to the existing literature as follows:

1. A thorough analysis of depression was carried out, as well as the exploration of the possible use of social media as a tool to express depression traits, and how machine learning can help detect depression on social media data.

2. We applied different feature representations with machine learning and deep learning algorithms for depression detection and evaluated the performance of the models using accuracy, recall, precision, and F1 scores.

3. We evaluated pre-trained language models and show that they exhibit outstanding performance by consistently achieving high accuracy, precision, recall, and F1 scores.

4. Context, feature extraction, and pre-training all had a significant impact on the models' performance as far as depression detection is concerned.

2 Literature Review

SA approaches have gained interest as a promising method of identifying patterns in text that can serve as indicators of depression. These approaches involve classifying the sentiment expressed in a given text to identify potential signs of depression symptoms.

In a study by Haque et al. [18], machine learning algorithms were employed to develop models capable of effectively identifying depression in children. The findings revealed that the Random Forest Classifier exhibited the highest efficiency in detecting depression.

Furthermore, the study identified 11 specific questions that can be used to detect depression in children and adolescents, helping to early diagnosis and treatment of this condition while understanding the contributing factors.

Another study by Reece et al. [38] used machine learning techniques to analyze Instagram data to identify possible indicators

Table 1. Related Studies on Detecting Depression

Model	Reference	F1 Score	Accuracy	Year
MNB	S.G. Burdisso et al.	0.96	0.96	2019
MLP	I. Fatima et al.	0.92	0.92	2019
CNN	J. Kim	0.79	0.75	2020
RFC	A Priya et al.	0.77	0.80	2020
Char CNN	K. Cornn	0.94	0.93	2020
SVM	H.S. AlSagri et al.	0.79	0.83	2020
Sense Mood	C. Lin et al.	0.94	0.88	2020
3D-CNN	H. Wang et al.	0.64	0.77	2021
RFC	EM de Souza Filho et al.	0.89	0.89	2021
LSTM	M. Muzammel et al.	0.95	0.95	2021
SBERT CNN	Z. Chen	0.86	0.86	2023

of depression. The study involved evaluating more than 43,000 Instagram photos and extracting statistical features such as color analysis, metadata components, and face identification.

Interestingly, their algorithm outperformed general practitioners in diagnosing depression, highlighting the potential of computational analysis of visual social media data as a scalable approach to detecting mental illnesses. In the study conducted by Cornn K. [14], a combination of various machine learning algorithms and neural networks was used to classify depression within social media text.

The most successful model was a CNN model, achieving an impressive accuracy of 92.5%. The one-dimensional convolutional layer played a vital role in noise reduction and was regarded as the most crucial component of the model.

Interestingly, the use of Word embedding proved to be ineffective in representing the text used in this particular study. In another work by Ziwei et al. [54], an application was developed to differentiate between depressive and non-depressive tweets using a classification function.

The application also provided a visualization of the user's depression status through a web interface. The research emphasized the importance of early detection of depression and highlighted the potential of social media platforms in predicting mental and physical illnesses.

However, the application faced limitations imposed by Twitter's API, such as the constraint of analyzing only a limited number of tweets. In a study conducted by De Choudhury et al. [15], sentiment analysis techniques were used to analyze Facebook data to detect symptoms of depression.

The findings revealed that individuals with depression symptoms tended to use a higher frequency of first-person pronouns, express negative emotions through their choice of words, and display a reduced use of terms associated with happiness in their Facebook posts, compared to individuals without symptoms.

Chen et al. [11] conducted a data analysis on Reddit data to identify people with depression. They proposed a hybrid deep learning model that combined a pre-trained sentence BERT (sBERT) with a convolutional neural network (CNN) to effectively identify individuals with depression based on their Reddit posts.

Interestingly, the model exceeded previously reported state-of-the-art results in the literature, achieving an accuracy of 0.86 and an F1 score of 0.86. The improved hybrid model was also applied to other text analysis tasks, showcasing its versatility and efficacy.

The research carried out by Wen et al. [51] used social media data to detect depression among users. Through the development of a classification model specifically designed to identify depression in tweets, the authors achieved remarkable results, with a high test accuracy of 98.94% and an F1 score of 99.04%.

The study highlights the effectiveness of analyzing the language used on social media platforms as a valuable approach for the early detection of depression among individuals. In a related study, Hosseini et al. [21] explored the integration of psychological and psychoanalytical insights to improve the identification of individuals with depression.

By combining traits observed in both depressed and non-depressed groups, the researchers created a bipolar feature vector. They successfully improved their models and achieved an impressive F1 score of 82.75% using a modified Bayesian classifier to classify social media users into depressed and non-depressed groups. In the research conducted by Wang et al. [50], a method to improve the features was introduced as input to a 3D CNN speech emotion recognition

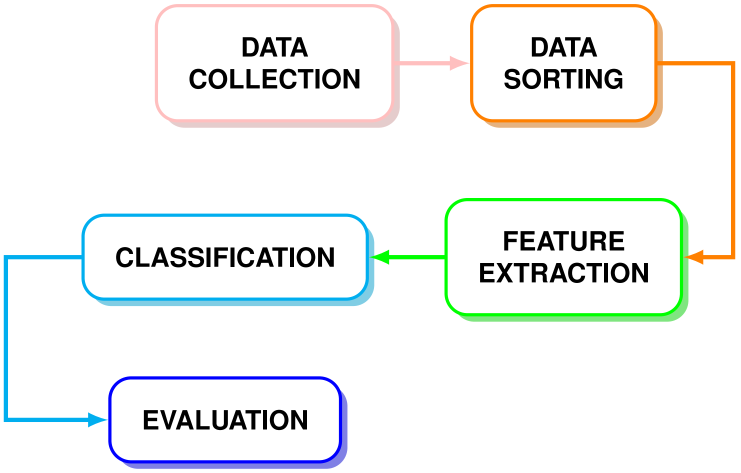


Fig. 1. Depression detection process flow chart

model, with the aim of identifying depression in its earliest stages.

The experiments carried out demonstrated that the combination of the enhanced feature and the model significantly improved the ability to detect and recognize depression.

Additionally, their study emphasized the necessity for future investigations to incorporate more detailed levels of analysis and extract additional features from speech signals to enhance detection accuracy.

Muzammel et al. [25] conducted experiments on depression detection by integrating multimodal features and selecting the optimal fusion strategy. The authors proposed two unimodal representations based on RNN and CNN networks.

These networks were utilized to acquire dynamic temporal representations of multimodal data, allowing for a comprehensive understanding of depression.

These investigations indicate that supervised learning techniques can be effective in identifying depression through the analysis of social media data. The summarized research findings related to depression detection are presented in Table 1.

However, there are some limitations to some of these methods that highlight the need to continue developing and fine-tuning these techniques to improve their accuracy and effectiveness. Figure 1 illustrates the steps involved in our classification method.

3 Methodology

3.1 Data

The dataset used in this experiment was sourced from Kaggle [5], a widely used platform known for hosting diverse datasets and machine learning competitions for individuals and organizations. It consists of depression-related text, acquired from Reddit, a highly popular social media platform worldwide, using web scraping techniques.

The datasets includes a total of 7,731 posts, which we divided into train and test sets to ensure accuracy and consistency in the analysis. The sentiment classes are ('1') or non-depression ('0'), which indicate whether the text contained expressions of depression or not.

Table 2 presents an example of text labeled with sentiment classes denoting depression and non depression. The dataset was divided into two parts: the training set and the testing set, consisting of 6539 and 1192 text inputs, respectively.

Table 3 presents the statistics of the text indicating depression and non-depression in both the training and the testing sets.

Table 2. Sample Text with Sentiment Classes

Text	Label
i ve lost everything i lost my best friend a community of people who were my only social outlet i m a failure i m i ve never been in a relationship i couldn t graduate college i m stuck working at a job which doesn t pay enough for me to afford rent so i have to live with my retirement age parent i can t find a job anywhere else i started cutting myself today never did it a a teenager but i did it now and it feel great i don t want to die but i don t see any other solution i can not afford help to me being in debt is worse than death i ve lost so much i can t go on	1
I ve been feeling really depressed lately and find myself with no one to talk I have these cry spell whenever i m alone and convinced that i m worthless and not worth anyone s time it s getting harder to pick myself up from the floor bed and be productive or practice self care my friend live far away and emotionally at arm length my family understands that i m depressed but not how much it debilitates me with no one to talk to i feel trapped i m hoping finding online support can help me understand how to go on so i m kinda new to this how does this thread help you	1
am i really just that awful no one want to be my friend my old friend abuse me i hate everything but especially myself when will it get better	1
Our membership had expired and to renew them, we have to do a new induction which can't happen until next Tuesday	0
bored of sims for today and still thinking of a name for me and like youtube account to post our awesome new video on idea people	0
hetty christ heh yeah i shakily conquered the ladder pointless job though we are too far away to receive digital signal with antenna	0

To comprehensively evaluate the effectiveness and reliability of our depression detection models, we conducted extensive experiments by combining intelligent pre-trained transformer models with traditional machine learning techniques.

By integrating diverse feature representations and transformer architectures, we obtained valuable insights into the performance and suitability of various approaches for depression classification.

The availability of this dataset on Kaggle makes it easier for other researchers to replicate this experiment and build on the work done in this research.

3.2 Models

The traditional machine learning algorithms included Multinomial Naive Bayes (MNB) [37], Stochastic Gradient Descent (SGD) [53], Logistic Regression Classifier (LRC) [40], Decision Tree Classifier (DTC) [45], Random Forest Classifier (RFC) [33], K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Multi-Layer Perceptron (MLP) [46].

These algorithms are commonly used in text classification tasks and are well established in the field of machine learning [34, 43, 29, 44]. Furthermore, we also used fine-tuned pre-trained language models for depression detection. The models used in the study included BERT [16], RoBERTa [23], XLM-RoBERTa [13],

Table 3. Statistics of depression and non-depression in the train and test datasets

Data	Instances	Label
Train	3,239	1
	3,300	0
Test	592	1
	600	0
Total	3,831	1
	3,900	0

DistilBERT [39], ALBERT [22], DistilRoBERTa [23] and ELECTRA [12]. These models are capable of capturing semantic and syntactic relationships between words, and the efficiency and effectiveness of these techniques make them often used for a wide range of applications, including language generation, machine translation and text classification.

4 Results

For this study, we evaluated different machine learning and pre-trained language models to detect and evaluate signs of depression. We extracted meaningful features from the text of social media to represent language patterns associated with depression.

The accuracy, precision, recall, and F1 evaluation metrics were used to assess the performance of the depression detection models. The features used in our experiments include bag-of-words (BoW), Word2Vec, and GloVe embeddings. By analyzing these results, we shed light on the profound influence of these distinct features on the overall performance of the models.

4.1 Experiment with Traditional Machine Learning Models and BoW

The BoW model represents text data as a collection of individual words and converts them into numerical representations that can be used by various machine learning algorithms. Machine learning models are trained on labeled datasets, where each text sample is associated with labels indicating the presence or absence of depression. The models learn to identify patterns and associations between the extracted BoW features and the corresponding labels.

The trained models are then evaluated using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score. Our findings are presented in Tables 4 - 7, which provide a comprehensive overview of the results of our experiments. In the experiment, several models were evaluated using the Bag-of-Words (BoW) feature representation, and their performance scores were recorded.

According to the results, the LRC model achieved the highest performance in all metrics, with an average accuracy, precision, recall, and F1 score of 0.96. This indicates that the model excelled at accurately classifying depression. The SGD and SVM also demonstrated strong performance with average scores of 0.94 and 0.95 respectively.

These models showed excellent overall performance in terms of accuracy, precision, recall, and F1 score. However, MNB, DTC, RFC, and MLP achieved good performance, with average scores ranging from 0.83 to 0.91. Although these models did not achieve as high scores as the top performers, they still exhibited reasonably good results.

The KNN model had the lowest performance among the evaluated models, with an average score of 0.74. This suggests that the model faced challenges in accurately classifying instances related to depression compared to the other models.

Table 4. Result of machine learning models using the BoW feature representation

Model	Average	Accuracy	Precision	Recall	F1
MNB	macro avg	0.83	0.87	0.83	0.83
	weighted avg		0.87	0.83	0.83
SGD	macro avg	0.94	0.94	0.94	0.94
	weighted avg		0.94	0.94	0.94
LRC	macro avg	0.96	0.96	0.96	0.96
	weighted avg		0.96	0.96	0.96
DTC	macro avg	0.86	0.87	0.86	0.86
	weighted avg		0.87	0.86	0.86
RFC	macro avg	0.93	0.93	0.93	0.93
	weighted avg		0.93	0.93	0.93
KNN	macro avg	0.74	0.78	0.74	0.73
	weighted avg		0.78	0.74	0.73
SVM	macro avg	0.95	0.96	0.95	0.95
	weighted avg		0.96	0.95	0.95
MLP	macro avg	0.91	0.91	0.91	0.91
	weighted avg		0.91	0.91	0.91

4.2 Experiment with Traditional Machine Learning Models and Word2Vec

Unlike the BoW model, Word2Vec captures not only the frequency of words, but also their semantic meaning and contextual relationships. The Word2Vec model learns dense vector representations by analyzing large corpora of text data.

It represents each word in a high-dimensional vector space, where words with similar meanings or contextual usage are located closer to each other. The text data were preprocessed by tokenizing the text into words and removing any stop words or irrelevant characters.

Each word is then replaced by its corresponding Word2Vec vector representation obtained from the pre-trained model. This transforms the text data into numerical vectors, where each word is represented by a dense vector of fixed length.

The Word2Vec vectors are subsequently used as input features for machine learning models to detect depression. The models learn to identify patterns and associations between Word2Vec embeddings and the corresponding labels and are evaluated using accuracy, precision, recall, and F1 score.

Using Word2Vec word embeddings, the models effectively capture semantic and contextual information within the text data, resulting in improved accuracy and more meaningful predictions. The findings of the analysis, using Word2Vec as feature representations, are presented in Table 5.

Table 5 presents notable insights into the performance of different machine learning models using Word2Vec features. Among these models, the MLP model stands out with an impressive accuracy of 0.94. Both the RFC and SVM models consistently demonstrated moderate performance

Table 5. Result of machine learning models using the Word2Vec feature representation

Model	Average	Accuracy	Precision	Recall	F1
MNB	macro avg	0.52	0.53	0.52	0.48
	weighted avg		0.53	0.52	0.48
SGD	macro avg	0.81	0.83	0.81	0.80
	weighted avg		0.84	0.81	0.80
LRC	macro avg	0.87	0.87	0.87	0.87
	weighted avg		0.87	0.87	0.87
DTC	macro avg	0.82	0.82	0.82	0.82
	weighted avg		0.82	0.82	0.82
RFC	macro avg	0.91	0.92	0.91	0.91
	weighted avg		0.92	0.91	0.91
KNN	macro avg	0.80	0.83	0.80	0.80
	weighted avg		0.83	0.80	0.80
SVM	macro avg	0.91	0.91	0.91	0.91
	weighted avg		0.91	0.91	0.91
MLP	macro avg	0.94	0.94	0.94	0.94
	weighted avg		0.94	0.94	0.94

with accuracy, precision, recall, and F1 scores hovering around 0.91. The SGD, KNN, LRC, and DTC models performed adequately, albeit at a slightly lower level. The MNB model exhibited poor performance, as indicated by lower accuracy, precision, recall, and F1 scores.

4.3 Experiment with Traditional Machine Learning Models and GloVe

In order to conduct further analysis, we employed the use of GloVe embedding representation to capture the semantic relationships between words. These vector representations are derived from the co-occurrence statistics of words in a corpus.

By encoding information about word meaning and context, these embeddings enable machine learning models to benefit from this knowledge. Using pre-trained GloVe embeddings, each word in the text is mapped to its corresponding vector representation.

These word vectors are then concatenated to create document-level representations, which are subsequently used to train the machine learning models. The results of our experiments using GloVe embeddings are presented in Table 6. The results of the experiment using the GloVe feature representation for machine learning models are summarized in Table 6.

The SGD, LRC, and SVM models consistently outperformed all other models, achieving high accuracy, precision, recall, and F1 scores of approximately 0.94, 0.96, and 0.95, respectively. The RFC model also exhibited strong performance, with accuracy, precision, recall, and an F1 score of around 0.93.

The KNN, DTC and MLP models yielded good performance, yielding an accuracy, precision, recall, and F1 score of approximately 0.74, 0.86, and 0.91, respectively. On the other hand, the MNB model showed relatively lower performance compared to the other models, with

Table 6. Results of machine learning models using the GloVe feature representation

Model	Average	Accuracy	Precision	Recall	F1
MNB	macro avg	0.60	0.64	0.60	0.58
	weighted avg	-	0.64	0.60	0.58
SGD	macro avg	0.94	0.94	0.94	0.94
	weighted avg	-	0.94	0.94	0.94
LRC	macro avg	0.96	0.96	0.96	0.96
	weighted avg	-	0.96	0.96	0.96
DTC	macro avg	0.86	0.87	0.86	0.86
	weighted avg	-	0.87	0.86	0.86
RFC	macro avg	0.93	0.93	0.93	0.93
	weighted avg	-	0.93	0.93	0.93
KNN	macro avg	0.74	0.78	0.74	0.73
	weighted avg	-	0.78	0.74	0.73
SVM	macro avg	0.95	0.96	0.95	0.95
	weighted avg	-	0.96	0.95	0.95
MLP	macro avg	0.91	0.91	0.91	0.91
	weighted avg	-	0.91	0.91	0.91

an accuracy, precision, recall, and F1 score of approximately 0.60. These findings indicate that when combined with these specific models, the GloVe feature representation can be highly valuable for the analysis and classification of depression in textual data.

4.4 Experiment with Transformer Architectures

Pre-trained language models have demonstrated remarkable success in various NLP tasks [31, 20]. Initially trained on vast amounts of text data from the Internet, these models acquire a contextual understanding of words and sentences.

To apply pre-trained language models for depression detection, we fine-tuned them by training them on labeled data. Labeled data consist of text samples annotated with depression-related labels.

Through the fine-tuning process, the pre-trained language models learn to capture significant linguistic patterns and contextual cues associated with depression.

Using the fine-tuned models, we classify new text samples as either indicating depression or not. We evaluated the models' performance using various metrics and found that the ELECTRA and Roberta-large models outperformed others, achieving the highest cumulative scores across all metrics.

Notably, these models achieved an F1 score of 0.99 each after only 10 epochs. Table 7 presents the performance of the transformer models. Table 7 displays the results of an extensive evaluation of various pre-trained language models in the depression detection task.

Transformer models showcased strong performance across all metrics evaluated. BERT achieved an accuracy, precision, recall, and F1 score of 0.97, demonstrating its effectiveness in detecting depression.

Furthermore, models such as RoBERTa, XLM-RoBERTa, DistilBERT, ALBERT and ELECTRA consistently achieved high scores, with accuracy, precision, recall, and F1 scores around or above 0.98.

Table 7. Results of the Transformer models in the experiment

Model	Feature	Accuracy	Precision	Recall	F1
BERT	Transformer	0.97	0.97	0.97	0.97
	Embedding				
RoBERTa	Transformer	0.99	0.99	0.99	0.99
	Embedding				
XLM-RoBERTa	Transformer	0.98	0.98	0.98	0.98
	Embedding				
DistilBERT	Transformer	0.98	0.98	0.98	0.98
	Embedding				
ALBERT	Transformer	0.98	0.98	0.98	0.98
	Embedding				
DistilRoBERTa	Transformer	0.96	0.97	0.96	0.96
	Embedding				
ELECTRA	Transformer	0.99	0.99	0.99	0.99
	Embedding				

These models exhibited robustness and reliability in the capture and comprehension of complex language patterns. While DistilRoBERTa slightly underperformed compared to other models, it still achieved an accuracy of 0.96. Both the ELECTRA and Roberta large models achieved the highest F1 scores of 0.99 each.

This underscores its exceptional potential for accurately detecting depression in this experiment. Incorporating these models could potentially revolutionize the identification and treatment of depression, leading to early detection and treatment.

5 Discussion

Pre-trained language models can continuously improve and adapt as they encounter new data, enhancing their diagnostic accuracy and generalization capabilities.

In this research, an investigation was conducted into the effectiveness of a variety of machine learning models in detecting depression in social media data, including pre-trained language models such as BERT, RoBERTa, XLM-RoBERTa, DistilBERT, ALBERT, DistilRoBERTa, and ELECTRA.

These models were assessed based on their accuracy, precision, recall, and F1 scores. Throughout the test, all transformer models achieved high accuracy and F1 scores, with RoBERTa and ELECTRA as the best performers.

This high performance of pre-trained transformer models suggests that they can effectively identify depression in text data.

Furthermore, these models can provide institutions responsible for the prevention of depression with a cost-effective alternative to their traditional methods of recognizing depression.

With the use of pre-trained language models and social media data for depression detection, significant advancement has been made in this study, emphasizing the potential of pre-trained language models and social media analysis for depression treatment and prevention.

6 Conclusions

As pre-trained language models continue to evolve, they hold the potential to revolutionize the field of depression prevention and treatment. The key strength of pre-trained language models lies in their ability to learn from vast amounts of diverse textual data, enabling them to discern subtle linguistic cues indicative of depression across different languages. Our study demonstrates the high effectiveness of pre-trained language models in detecting depression in English text from social media sources.

In our experiments, the pre-trained language models with which we experimented obtained very good accuracy, precision, recall, and F1 values. However, more research is needed to determine whether they are generalizable to larger, more diverse datasets and a different language. Real-world application challenges such as model biases, interpretability, and scalability still need to be addressed.

Our findings still underscore the need to leverage these pretrained language models to detect and address depression at scale. Through continued development, these models can contribute significantly to early detection and improved well-being for individuals suffering from depression.

Acknowledgments

The work was carried out with partial support from the Mexican government through grant A1-S-47854 of CONACYT, Mexico, and by the Secretaria de Investigacion y Posgrado of the Instituto Politécnico Nacional, Mexico, under grants 20211884, 20220859 and 20230140, EDI; and COFAA-IPN.

The authors thank CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercomputo of the INAOE, Mexico, and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

References

1. **Abdul-Razzak, H., Harbi, A., Ahli, S. (2019).** Depression: Prevalence and associated risk factors in the United Arab Emirates. Oman Medical Journal, Vol. 34, No. 4, pp. 274–282. DOI: 10.5001/omj.2019.56.

2. **Adebanji, O. O., Gelbukh, I., Calvo, H., Ojo, O. E. (2022).** Sequential models for sentiment analysis: A comparative study. Proceedings of the 21st Mexican International Conference on Artificial Intelligence. Advances in Computational Intelligence, pp. 227–235. DOI: 10.1007/978-3-031-19496-2_17.

3. **Ali, F., Tauni, M. Z., Ashfaq, M., Zhang, Q., Ahsan, T. (2023).** Depressive mood and compulsive social media usage: The mediating roles of contingent self-esteem and social interaction fears. Information Technology and People. DOI: 10.1108/itp-01-2021-0057.

4. **Armenta-Segura, J., Núñez-Prado, C. J., Sidorov, G. O., Gelbukh, A., Román-Godínez, R. F. (2023).** Ometeotl@Multimodal hate speech event detection: Hate speech and text-image correlation detection in real life memes using pre-trained BERT models over text. Proceedings of the 6th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text, pp. 53–59.

5. **Banachewicz, K., Massaron, L., Goldbloom, A. (2023).** The Kaggle book: Data analysis and machine learning for competitive data science. Pack Publishing, pp. 530.

6. **Bhatt, S., Devadoss, T., Jha, N. K., Baidya, M., Gupta, G., Chellappan, D. K., Singh, S. K., Dua, K. (2023).** Targeting inflammation: A potential approach for the treatment of depression. Metabolic Brain Disease, Vol. 38, No. 1, pp. 45–59. DOI: 10.1007/s11011-022-0 1095-1.

7. **Braddock, J., Heide, S., Spaniardi, A. (2023).** Introduction to the virtual world: Pros and cons of social media. Teens, Screens, and Social Connection: An Evidence-Based Guide to Key Problems and Solutions, pp. 31–48. DOI: 10.1007/978-3-031-24804-7_3.

8. **Bui, H. Q., Tran, T. T. (2023).** CMC users' positive and negative emotions: Features of social media platforms and

users' strategies. In Multidisciplinary Applications of Computer-Mediated Communication. pp. 188–210. DOI: 10.4018/978-1-6684-7034-3.ch010.

9. **Buodo, G., Moretta, T., Santucci, V. G., Chen, S., Potenza, M. N. (2023).** Using social media for social motives moderates the relationship between post-traumatic symptoms during a COVID-19-related lockdown and improvement of distress after lockdown. Behavioral Sciences, Vol. 13, No. 1, pp. 53. DOI: 10.3390/bs13010053.

10. **Calvo, H., Carrillo-Mendoza, P., Gelbukh, A. (2018).** On redundancy in multi-document summarization. Journal of Intelligent and Fuzzy Systems, Vol. 34, No. 5, pp. 3245–3255. DOI: 10.3233/jifs-169507.

11. **Chen, Z., Yang, R., Fu, S., Zong, N., Liu, H., Huang, M. (2023).** Detecting reddit users with depression using a hybrid neural network. Proceedings of the 11th IEEE International Conference on Healthcare Informatics, pp. 610–617. DOI: 10.1109/ICTACS56270.2022.9988489.

12. **Clark, K., Luong, M. T., Le, Q. V., Manning, C. D. (2020).** ELECTRA: pre-training text encoders as discriminators rather than generators. International Conference on Learning Representations, pp. 1–18. DOI: 10.48550/ARXIV.2003.10555.

13. **Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V. (2019).** Unsupervised cross-lingual representation learning at scale. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451.

14. **Cornn, K. (2020).** Identifying depression on social media. Stanford University Stanford.

15. **De Choudhury, M., Gamon, M., Counts, S., Horvitz, E. (2021).** Predicting depression via social media. Proceedings of the International AAAI Conference on Web and Social Media, Vol. 7, No. 1, pp. 128–137. DOI: 10.1609/icwsm.v7i1.14432.

16. **Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019).** BERT: Pre-training of deep bidirectional transformers for language understanding. North American Chapter of the Association for Computational Linguistics, Vol. 1, pp. 4171–4186.

17. **García-Mendoza, C. V., Gambino, O. J., Villarreal-Cervantes, M. G., Calvo, H. (2020).** Evolutionary optimization of ensemble learning to determine sentiment polarity in an unbalanced multiclass corpus. Entropy, Vol. 22, No. 9, pp. 1020. DOI: 10.3390/e22091020.

18. **Haque, U. M., Kabir, E., Khanam, R. (2021).** Detection of child depression using machine learning methods. Public Library of Science One, Vol. 16, No. 12, pp. e0261131. DOI: 10.1371/journal.pone.0261131.

19. **Hernández-Castañeda, A., Calvo, H., Gelbukh, A., García-Flores, J. J. (2016).** Cross-domain deception detection using support vector networks. Soft Computing, Vol. 21, No. 3, pp. 585–595. DOI: 10.1007/s00500-016-2409-2.

20. **Hoang, T. T., Ojo, O. E., Adebanji, O. O., Calvo, H., Gelbukh, A. (2022).** The combination of BERT and data oversampling for answer type prediction. Proceedings of the Central Europe Workshop, Vol. 3119.

21. **Hosseini-Saravani, S. H., Besharati, S., Calvo, H., Gelbukh, A. (2020).** Depression detection in social media using a psychoanalytical technique for feature extraction and a cognitive based classifier. Proceedings of the 19th Mexican International Conference on Artificial Intelligence. Advances in Computational Intelligence, Vol. 12469, pp. 282–292. DOI: 10.1007/978-3-030-60887-3_25.

22. **Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. (2019).** ALBERT: A lite BERT for self-supervised learning of language representations.

Proceedings of the International Conference on Learning Representations. Conference Blind Submission. DOI: 10.48550/arXiv.1909.11942.

23. **Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019).** RoBERTa: A robustly optimized BERT pretraining approach. Proceedings of the International Conference on Learning Representations. Conference Blind Submission. DOI: 10.48550/ARXIV.190 7.11692.

24. **Mustafa, R. U., Ashraf, N., Ahmed, F. S., Ferzund, J., Shahzad, B., Gelbukh, A. (2020).** A multiclass depression detection in social media based on sentiment analysis. Proceedings of the 17th International Conference on Information Technology New Generations, pp. 659–662. DOI: 10.1007/978-3-030-43020-7_89.

25. **Muzammel, M., Salam, H., Othmani, A. (2021).** End-to-end multimodal clinical depression recognition using deep neural networks: A comparative analysis. Computer Methods and Programs in Biomedicine, Vol. 211, pp. 106433. DOI: 10.1016/j.cmpb.2021.106433.

26. **Ojo, O., Adebanji, O., Calvo, H., Dieke, D., Ojo, O., Akinsanya, S., Abiola, T., Feldman, A. (2023).** Legend at ArAIEval shared task: Persuasion technique detection using a language-agnostic text representation model. Proceedings of ArabicNLP, pp. 594–599.

27. **Ojo, O. E., Adebanji, O. O., Gelbukh, A., Calvo, H., Feldman, A. (2023).** MedAI dialog corpus (MEDIC): Zero-shot classification of doctor and AI responses in health consultations.

28. **Ojo, O. E., Gelbukh, A., Calvo, H., Adebanji, O. O. (2021).** Performance study of n-grams in the analysis of sentiments. Journal of the Nigerian Society of Physical Sciences, Vol. 3, No. 4, pp. 477–483.

29. **Ojo, O. E., Gelbukh, A., Calvo, H., Adebanji, O. O., Sidorov, G. (2020).** Sentiment detection in economics texts. Proceedings of the 20th Mexican International Conference on Artificial Intelligence. Advances in Computational Intelligence, pp. 271–281. DOI: 10.1007/978-3-030-60887-3_24.

30. **Ojo, O. E., Gelbukh, A., Calvo, H., Feldman, A., Adebanji, O. O., Armenta-Segura, J. (2022).** Language identification at the word level in code-mixed texts using character sequence and word embedding. Proceedings of the 19th International Conference on Natural Language Processing: Shared Task on Word Level Language Identification in Code-mixed Kannada-English Texts, pp. 1–6.

31. **Ojo, O. E., Ta, H. T., Gelbukh, A., Calvo, H., Adebanji, O. O., Sidorov, G. (2023).** Transformer-based approaches to sentiment detection. Vol. 2, pp. 101–110. DOI: 10.1007/978-3-031-23476-7_10.

32. **Ojo, O. E., Ta, T. H., Gelbukh, A., Calvo, H., Sidorov, G., Adebanji, O. O. (2022).** Automatic hate speech detection using deep neural networks and word embedding. Computación y Sistemas, Vol. 26, No. 2, pp. 1007–1013. DOI: 10.13053/cys-26-2-410 7.

33. **Parmar, A., Katariya, R., Patel, V. (2019).** A review on random forest: An ensemble classifier. International Conference on Intelligent Data Communication Technologies and Internet of Things, Vol. 26, pp. 758–763. DOI: 10.1007/978-3-030-03146-6_86.

34. **Peng, J., Feldman, A., Jazmati, H. (2015).** Classifying idiomatic and literal expressions using vector space representations. Proceedings of the International Conference Recent Advances in Natural Language Processing, pp. 507–511.

35. **Peng, J., Feldman, A., Vylomova, E. (2014).** Classifying idiomatic and literal expressions using topic models and intensity of emotions. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 2019–2027. DOI: 10.3115/v1/D14-1216.

36. **Popat, A., Tarrant, C. (2022).** Exploring adolescents' perspectives on social media and mental health and well-being – A qualitative literature review. Clinical Child Psychology and Psychiatry, Vol. 28, No. 1, pp. 323–337. DOI: 10.1177/13591045221092884.

37. **Raschka, S. (2014).** Naive Bayes and text classification I - Introduction and theory. arXiv. DOI: 10.48550/arXiv.1410.5329.

38. **Reece, A. G., Danforth, C. M. (2017).** Instagram photos reveal predictive markers of depression. European Physical Journal of Data Science, Vol. 6, pp. 15. DOI: 10.1140/epjds/s13688-017-0110-z.

39. **Sanh, V., Debut, L., Chaumond, J., Wolf, T. (2019).** DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. Proceedings of the 5th Edition Co-located with Neural Information Processing Systems, pp. 1–5. DOI: 10.48550/arXiv.1910.01108.

40. **Schein, A. I., Ungar, L. H. (2007).** Active learning for logistic regression: an evaluation. Machine Learning, Vol. 68, No. 3, pp. 235–265. DOI: 10.1007/s10994-007-5019-5.

41. **Shahiki-Tash, M., Armenta-Segura, J., Ahani, Z., Kolesnikova, O., Sidorov, G., Gelbukh, A. (2023).** LIDOMA at HOMO-MEX2023@IberLEF: Hate speech detection towards the mexican spanish-speaking LGBT+ population. The importance of preprocessing before using BERT-based models. Proceedings of the Central Europe Workshop and Iberian Languages Evaluation Forum, Vol. 3496.

42. **Shahiki-Tash, M., Armenta-Segura, J., Ahani, Z., Kolesnikova, O., Sidorov, G., Gelbukh, A. (2023).** LIDOMA@ DravidianLangTech: Convolutional neural networks for studying correlation between lexical features and sentiment polarity in Tamil and Tulu languages. Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages, pp. 180–185.

43. **Sidorov, G., Miranda-Jiménez, S., Viveros-Jiménez, F., Gelbukh, A., Castro-Sánchez, N., Velásquez, F., Díaz-Rangel, I., Suárez-Guerra, S., Treviño, A., Gordon, J. (2012).** Empirical study of machine learning based approach for opinion mining in tweets. Proceedings of the 11th Mexican International Conference on Advances in Artificial Intelligence, Vol. 7629, pp. 1–14. DOI: 10.1007/978-3-642-37807-2_1.

44. **Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernández, L. (2014).** Syntactic n-grams as machine learning features for natural language processing. Expert Systems with Applications, Vol. 41, No. 3, pp. 853–860. DOI: 10.1016/j.eswa.2013.08.015.

45. **Swain, P. H., Hauska, H. (1977).** The decision tree classifier: Design and potential. IEEE Transactions on Geoscience Electronics, Vol. 15, No. 3, pp. 142–147. DOI: 10.1109/TGE.1977.6498972.

46. **Taud, H., Mas, J. F. (2018).** Multilayer perceptron (MLP). Geomatic approaches for modeling land change scenarios, pp. 451–455. DOI: 10.1007/978-3-319-60801-3_27.

47. **Tovar, M., Rosillo, M., Spaniardi, A. (2023).** Social media's influence on identity formation and self expression. Teens, Screens, and Social Connection: An Evidence-Based Guide to Key Problems and Solutions, pp. 49–61. DOI: 10.1007/978-3-031-24804-7_4.

48. **Trotzek, M., Koitka, S., Friedrich, C. M. (2018).** Utilizing neural networks and linguistic metadata for early detection of depression indications in text sequences. IEEE Transactions on Knowledge and Data Engineering, Vol. 32, No. 3, pp. 588–601. DOI: 10.1109/tkde.2018.2885515.

49. **Uddin, M. Z., Dysthe, K. K., Følstad, A., Brandtzaeg, P. B. (2021).** Deep learning for prediction of depressive symptoms in a large textual dataset. Neural Computing and Applications, Vol. 34, No. 1, pp. 721–744. DOI: 10.1007/s00521-021-06426-4.

50. **Wang, H., Liu, Y., Zhen, X., Tu, X. (2021).** Depression speech recognition with a three-dimensional convolutional network. Frontiers in Human Neuroscience, Vol. 15. DOI: 10.3389/fnhum.2021.713823.

51. **Wen, S. (2021).** Detecting depression from tweets with neural language processing. Journal of Physics: Conference Series, Vol. 1792, No. 1, pp. 12058. DOI: 10.1088/1742-6596/1792/1/012058.

52. **Wolohan, J. T., Hiraga, M., Mukherjee, A., Sayyed, Z. A., Millard, M. (2018).** Detecting linguistic traces of depression in topic-restricted text: Attending to self-stigmatized depression with NLP. Proceedings of the 1st International Workshop on Language Cognition and Computational Models, pp. 11–21.

53. **Zhang, T. (2004).** Solving large scale linear prediction problems using stochastic gradient descent algorithms. Proceedings of the 21st International Conference on Machine Learning, pp. 116. DOI: 10.1145/1015330.1015332.

54. **Ziwei, B. Y., Chua, H. N. (2019).** An application for classifying depression in tweets. Proceedings of the 2nd International Conference on Computing and Big Data, pp. 37–41. DOI: 10.1145/3366650.3366653.

Article received on 30/08/2023; accepted on 16/11/2023.
** Corresponding author is Hiram Calvo.*

ISSN 2007-9737

Framework for Heterogeneous Data Management: An Application Case in a NoSQL Environment from a Climatological Center

Alicia Margarita Jiménez-Galina*, Aide Aracely Maldonado-Macías,
Karla Miroslava Olmos-Sanchez, Israel Hernández,
Fernando Estrada-Saldaña, Felipe Adrián Vázquez-Gálvez

Universidad Autónoma de Ciudad Juárez,
Chihuahua,
Mexico

al206560@alumnos.uacj.mx, {amaldona, kolmos,
israel.hernandez, festrada, fvazquez}@uacj.mx

Abstract. Processing, visualizing and understanding data from meteorological networks can present several challenges due to the variety and complexity of the data and must be accessible in real time and in different formats, protocols and standards. This paper presents the development of an innovative technological framework for handling heterogeneous climatological data in a NoSQL environment. The framework was developed following the Action Research methodology and enables the extraction of heterogeneous data, their homogenization, and the creation of a dataset. Its real-case application took place in data repositories used for climatological data management in a specialized regional center in Ciudad Juarez, México. The main repository use MongoDB and contain 631,202 documents with data from several meteorological stations. A 70% reduction in data processing time is evidence that the methodology and framework developed were effective in the case of the application. In addition, the generated data sets are homogenized and in formats compatible with advanced analysis tools.

Keywords. Heterogeneous data, homogenization, action research.

1 Introduction

Nowadays, organizations around the world generate information resulting from diverse activities; such information is composed of different kinds of data. This data can be structured, unstructured, or semi-structured, but it is mostly heterogeneous and comes from several sources.

Proper data processing and understanding are crucial as they lead to better predictions and decision making. Therefore, companies need to handle data efficiently, effectively, and reliably, seeking to use low-cost and optimization alternatives to safeguard them [17, 24].

1.1 Big Data Solutions for Data Management and Migration

One alternative to solve these organizational needs can be found in relational database systems, which have been used for more than 40 years for similar purposes [3].

Although currently storing large amounts of information in relational database systems is inexpensive, such systems have limitations in instances that require handling unstructured data and horizontal scaling, which makes it impossible to partition data on different computers.

That is why new technologies have emerged to help manage data effectively, for example MapReduce / Hadoop, and NoSQL, among others [4]. Likewise, cloud computing and new big data system-related applications have appeared, both of which can handle massive data, thus allowing organizations to improve their understanding of stored information.

Big data systems have the infrastructure, technology, and service capacities to manage large amounts of data.

doi: 10.13053/CyS-28-1-4474

Manage implies entry, storage, analysis, search, exchange, and transfer of data. Furthermore, data visualization, consultation, and actualization are important in maintaining their privacy, origin, veracity, and value of data [4].

Among these technologies, NoSQL databases provide flexible structures and enable the horizontal scaling of large amounts of data and users. Unstructured databases can be displayed in several forms, for example documents, key-values, column-widths, and graphs.

The best-known NoSQL database of the document type is MongoDB, which, in addition to other features, stores data such as objects in a JSON (JavaScript Object Notation) format and allows each record to become a document with an independent structure from the others [3].

The tendency in organizations is to take advantage of these new information management technologies to enhance their decision-making processes and promote data-based innovative solutions [7]. However, for those cases where data is scattered though different repositories and relational databases, information management process can be difficult and laborious.

Currently, it is necessary to migrate information to unstructured schemes in an optimal way while preserving the basic principles of integrity, confidentiality, and availability. Thus, migration from a structured database like SQL to an unstructured NoSQL like MongoDB is increasingly frequent and necessary [21, 31, 1].

However, the process must be careful and efficient since it is time-consuming and, most of the time, underestimated [27]. In addition, it requires exhaustive data origin analysis [8]. Some cases have been found where from data contained in the SQL database, a record migrated efficiently as a document to MongoDB [3, 10].

Therefore, effective data migration requires the use of data mining techniques such as ETL (Extract-Transform-Load) processes, which are useful and convenient for data source analysis and which also make the cleaning, transforming, or reformatting processes possible [12, 13, 27].

Other advantages of ETL processes are the establishment of a central repository, as well as decision-making processes based on the analysis of data concentrated in a new database. They also aid in various processes such as data migration between different applications, as well as their synchronization and consolidation [22]. These processes extract data from one or more sources, transform them or clean them if necessary, and load them into another database, called Data Warehouse (DW), for later analysis.

However, migration processes also present difficulties; among them are the migration of structured SQL databases to unstructured NoSQL in MongoDB [1], the homogenization of heterogeneous data [17], and the creation of a dataset from MongoDB databases [5].

1.2 Solutions for Migrating Data Between Databases

Regarding the difficulty of migration between databases, some authors have proposed some solutions to improve the implementation of ETL processes [13]. These authors have also carried out in-depth studies on aspects such as elasticity, dynamism, and the cost of resources.

Additionally, they have analyzed ETL solutions for the domain of big data in the cloud through task or programming parallelization [12] and have proposed an alternative solution based on a new architecture that eliminates the “buffer zone” to cut storage space in half, in addition to reducing data-processing time.

Further solutions have been proposed where semantic technology techniques were used based on data in the cloud and big data characteristics such as speed, variety, and volume [13]. Finally, some solutions proposed improvements to the ETL process by combining the Query Cache and Scripting methods [27].

1.3 Miscellaneous Data-Management Studies

This section presents several studies on heterogeneous data management, SQL queries, the use of metadata, and other frameworks developed to migrate and homogenize data. Regarding solutions for managing heterogeneous data, the literature has shown efficient use of framework development.

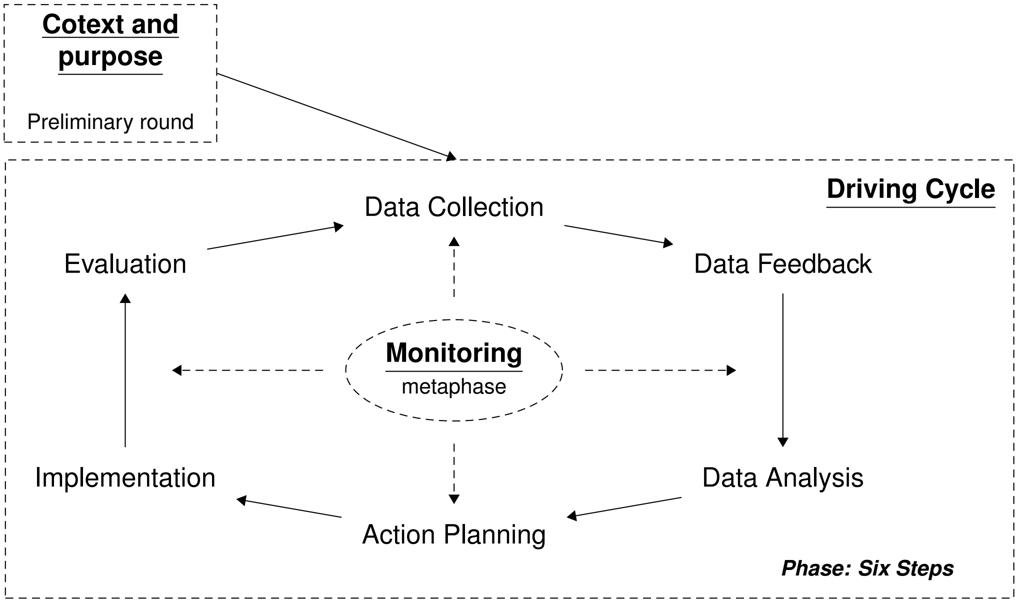


Fig. 1. Action research cycle [14]

For example, the HDS Analytics framework created a heterogeneous dataset to feed an analysis model that located the shortest route in a public transport domain [18].

Another framework was developed to detect medical events and create trends by correlating physical sensors such as temperature, air pressure, wind, and rain with suspended particles and a social sensor [11].

One further study [32] referred to a semi-structured query engine through which SQL queries were optimized according to the model. In addition, several proposals were found for the handling of non-relational data, and the use of metadata and their integrity.

For example, [30] managed data effectively through the use of an R-tree structure for operations in MongoDB. Another solution proposed by [23] improved the dataset homogenization process by incorporating metadata to optimize queries and data integration.

Finally, [20] developed a solution in order to take care of information integrity, which created a framework that, together with the metadata, enabled the extraction of the information to be analyzed.

Finally, some authors [19] presented a project created in a NoSQL environment, whose process of quality evaluation, homogenization, and visualization of climatological data precedes the development of this framework.

As can be seen, the implementation of new technologies in the areas of dataset migration, homogenization, and generation contributes to a better understanding of data. However, special care must be taken during data source analysis to identify valuable content and convert it to a JSON document for effective migration to MongoDB.

1.4 Action Research Methodology

The Action Research (AR) methodology aims to address a problem in an organization, whether it relates to a research topic or an organizational challenge, and solve it in a cooperative and participatory way [15, 26, 6].

Another research study added the participatory and simultaneous elements to the characteristics of the action looking for innovative solutions [9]. As shown in Figure 1, the AR methodology consists of a preliminary round that includes the driving cycle and the monitoring metaphase [14].

In the preliminary round, the objectives and the context are established and understood. Then, the Driving Cycle takes place; it involves a six-step phase that focuses initially on data and then on the action.

Thus, it first collects, gives feedback on, and analyzes data, and then plans, implements, and evaluates the action. The monitoring metaphase is the follow-up phase, in which the results of each of the steps are verified.

1.5 Paper Contribution and Organization

The literature review shows that, thus far, only partial solutions for migration processes have been offered. Thus, the development of a complete and comprehensive solution can be considered an open problem or an opportunity for innovation.

Because innovations in these processes are necessary for a better understanding of data, this paper presents a development of an innovative technological framework applied to an environmental data and information management case with the following characteristics:

Handling of heterogeneous data in a NoSQL environment, an initial storage procedure, data extraction and transformation methods, and dataset creation in three different formats for subsequent analysis.

The framework was developed using the Action Research methodology, which has the advantages of providing effective solutions for improving processes, practices, and strategies [15, 26, 6, 9, 14]. This paper is organized as follows.

The introduction includes the problem statement and the literature review. Section 2 describes the methodology used for the creation of the framework, as well as the use of other studies.

Section 3 explains the case of application in the climatological center, including the context and purpose, and an explanation of the driving cycle with its five processes: metadata, integral solution, uploading, development, and evaluation. Finally, Section 4 discusses the conclusions as well as some future research initiatives.

2 Methodology

This section describes how the AR methodology was used, as well as complementary studies in the real application case for the framework developed.

2.1 Application of the AR Methodology

The implementation of the AR methodology in the development of the framework took place in two main parts. Part one consisted of the preliminary round, which included the context and the purpose. Part two consisted of the driving cycle, composed of the six steps in the monitoring metaphase.

The monitoring metaphase supervised and verified each of the steps in collaboration with the experts. The application and results of the steps in the application case in the climatological center are also described.

2.2 Miscellaneous Proposals for Data Management

In addition to the AR methodology used, this section presents some proposals for managing heterogeneous data that were considered for the framework development: Investigations related to ETL processes were used at different stages of the framework for cleaning, loading, and transforming data; during the phase of initial loading of SQL to MongoDB; and later in the homogenization and dataset creation phase [12, 13].

The investigations by [18, 11, 32] were taken into consideration to improve heterogeneous data understanding and management, while the studies by [30, 23] influenced the development of the structure and use of metadata to support the management framework.

3 Application Case: Climatological Center

This section explains the implementation of the two main parts of the AR methodology in the application case.

3.1 Part One. Context and Purpose

The importance of having historical climate bases has been highlighted by several authors. Some authors, promoted their use in order to improve climate predictions [16]; others proposed them to support agriculture [29]; some others used them to analyze energy, health, and insurance [28]; and others analyzed the impact of climatic variability on natural gas [25].

The case chosen for the application of this framework was the Centro de Estudios Atmosféricos y Tecnologías Verdes, CECATEV (Center for Atmosphere Studies and Green Technologies, for its Spanish acronym), which is located at the Universidad Autónoma de Ciudad Juárez (Autonomous University of Ciudad Juarez, UACJ for its Spanish acronym) as part of a collaboration agreement between the UACJ and the Instituto Nacional de Ecología y Cambio Climático (National Institute for Ecology and Climate Change).

CECATEV was created as a scientific reference laboratory for the Ciudad Juarez atmospheric basin air quality program and oversees the maintenance of the climatological network as well as the study of air pollution. CECATEV has worked on different projects to increase the meteorological stations in the city. To carry out its work, the center must create big data systems to concentrate the climate variables in the region's climatological network in databases.

These meteorological data are temperature, direction, wind speed, relative humidity, evaporation, rainfall, and solar radiation. Once the information is gathered, it must be shared with experts, different users, and several universities inside and outside the country.

This application case was carried out using a central repository in MongoDB containing 631,202 documents with data from five meteorological stations and one station for gases and suspended particles. To fully understand the impact of this application case, the following sections will describe its manual process as well as the problem studied.

3.1.1 Manual Process Description

The following manual process was carried out in the meteorological station: Every day, users downloaded files in csv (Comma Separated Values) format from a repository on the web.

Then, they conducted a data cleaning, or pre processing procedure, to eliminate hyphens, invalid characters, and non corresponding columns; this process was carried out in an Excel file. Once data was pre-processed, it was loaded onto a tool called “R” used to create graphs and analyze data.

If users found any human error at any step of data processing, they had to restart, which delayed the process. Another way to develop the dataset was to run a query directly on the SQL server, yet this put the integrity of data at risk due to direct manipulation.

In addition, the staff lacked the knowledge to generate SQL queries and troubleshoot any kind of errors. After the query was successfully generated, it was exported to a csv file and users could go through the cleanup process described above.

The previous cleaning and loading processes also applied to the gas and suspended particles station, except for the generation of the file since the laboratory staff would have had to enter the CECATEV site to access the server and download the files, and that would have represented a risk to physical security and data integrity.

Thus, the creation of a dataset of non homogenized data from a station considering one month of data took about 40 to 60 minutes.

3.1.2 Problem Description and Purpose

There were different problems in the manual process described, such as the time used, the risk of integrity, the management of heterogeneous data and the number of stations.

With respect to time and integrity, by including information from specific sensors and multiple stations, it involved several complex manual processes that required a lot of time and represented a significant risk factor to data integrity.

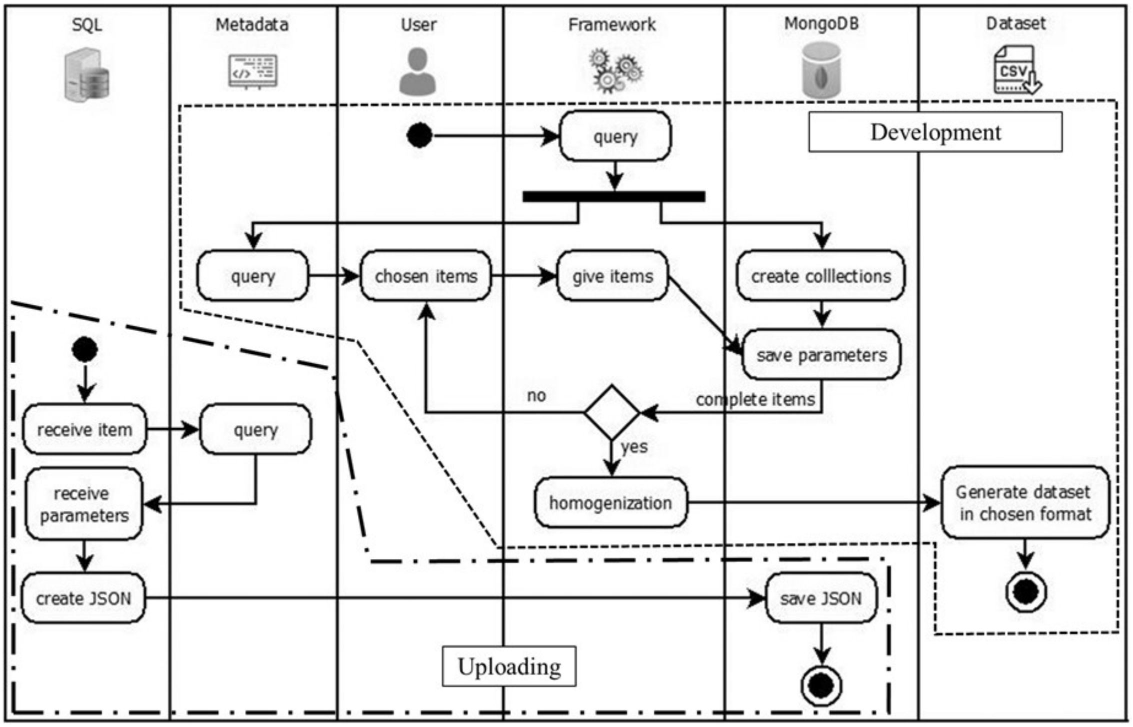


Fig. 2. Diagram of the solution

Regarding the heterogeneity of the data, it is due to various factors, such as the weather network is made up of weather stations of different brands and models. Another factor is that the stations for gases and suspended particles are of different types.

In addition, each of the stations can contain a different number of sensors and of a different brand; finally, the readings collected may be in different units of measure. Due to these factors, the resulting data set was not homogenized. Finally, this process was carried out individually by weather station.

Meteorological data analysis is used to support decision making, product development and a better understanding of radioactive and contaminating processes in our region, but data visualization has been a constant challenge.

Therefore, it was essential to design a tool that would allow data processing, time minimization, and data homogenization so that they could be assimilated into predictive atmosphere models.

3.2 Part two. Driving Cycle

This section will describe the driving cycle, which includes the 6 steps of the monitoring metaphase that are embedded in the processes. The use of italics emphasizes these steps in the text. The first process describes the definition and construction of the metadata, followed by the development of a comprehensive solution.

The initial data loading is described later, and the development of the framework is explained at the end, along with the evaluation of the framework in the application case.

3.2.1 Metadata

The steps of the monitoring metaphase, data collection, data feedback and data analysis, involved a review of the origin of data, hence it was necessary to migrate it efficiently. The information that is migrated from a relational database to a NoSQL can only include the valuable data instead of the entire record; that is, in NoSQL databases

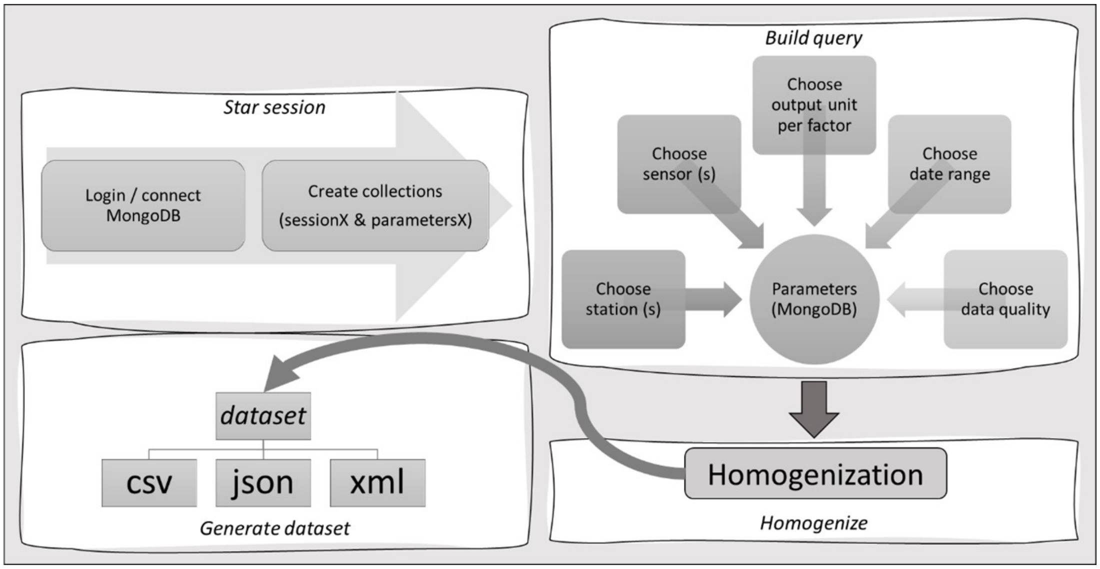


Fig. 3. Framework components diagram

it is possible to have documents with independent structures. That is the reason why the metadata was defined and constructed in XML format. The metadata was made up of four groups.

The first group contained the information related to the server; the second group was formed by the conversion factors, which made it possible to choose the output unit to homogenize the sensor values; the third group included the stations' profile; and the fourth group contained the sensors belonging to each station.

The use of metadata had two purposes: the first was to provide the elements that would make up the structure of the document in JSON, which would be built from the fields with SQL values, inserted into MongoDB. The second purpose was to provide the elements to be included in the dataset, as well as the information needed to homogenize and generate dataset in the output.

In this application case, data collected by the weather stations was stored in a SQL server with a database of 65 fields. However, the stations had an average of 19 sensors taking readings, therefore there was storage waste. To solve this, the metadata provided by the elements was used to build the JSON document for migration.

3.2.2 Integral Solution

In the action planning and implementation steps, the integral solution was designed. It was made up of the process of uploading SQL to MongoDB (see section 3.2.3) and the development of the framework (see section 3.2.4). Figure 2 shows the activity diagram of the solution.

3.2.3 Uploading

This process was carried out collaboratively. The name of the station to migrate was provided. Then the metadata was checked to identify the fields with values. Finally, the query was built to generate the JSON file to load to MongoDB.

3.2.4 Development

This process led to the development of the described framework as a solution to the process of homogenization and generation of datasets in the csv, JSON, and XML formats. The central repository used for this project was a NoSQL database on MongoDB, and the framework was designed using the Python language because of its advantages in the use of mathematical functions and its compatibility with MongoDB [2].

Table 1. Homogenized data processing times per station using the framework

Process of Framework													
Line	Station	# Sensors	Start Date	End Date	Days	Documents MongoDB	Time in seconds						Total time in minutes
							Generation Dataset	Generation File:				Total time all files	
								readme	json	csv	xml		
1	Estacion 05	14	2019-06-27 18:10	2019-10-04 18:05	98.9965278	17,358	75.3255	0.0034	1.9741	0.7102	1.1697	79.1830	1.3197
2	Estacion 05	14	2019-08-04 18:05	2020-01-15 01:05	163.291667	31,936	132.5774	0.0041	3.9222	1.3321	2.6404	140.4762	2.3413
3	Estacion 05	14	2019-06-27 18:10	2020-01-15 01:05	201.288194	39,780	169.1441	0.0030	4.9403	1.6928	3.2616	179.0418	2.9840
4	Estacion 09	19	2019-03-08 19:50	2019-06-08 19:50	92	18,980	79.3514	0.0045	2.5711	0.9402	1.7722	84.6394	1.4107
5	Estacion 09	19	2019-03-08 19:50	2019-09-04 18:05	179.927083	36,914	156.2165	0.0029	5.1794	1.8674	3.2699	166.5360	2.7756
6	Estacion 09	19	2019-03-08 19:50	2019-11-04 18:05	240.927083	54,467	233.9262	0.0029	7.4542	2.6587	4.8130	248.8551	4.1476
7	Estacion 25	19	2019-10-01 00:00	2019-11-01 00:00	31	8,929	40.2444	0.0038	1.3998	0.4805	0.8378	42.9664	0.7161
8	Estacion 25	19	2019-01-01 00:00	2020-01-01 00:00	365	80,353	358.0659	0.0030	11.9192	4.3548	7.7943	382.1372	6.3690
9	**Estacion 25**	**19**	**2017-04-03 20:55**	**2020-01-15 00:40**	**1016.15625**	**194,377**	**861.0886**	**0.0041**	**29.6684**	**10.6237**	**18.9564**	**920.3411**	**15.3390**
10	Estacion 26	14	2019-03-28 19:40	2019-06-28 19:40	92	26,460	114.6024	0.0036	3.1081	1.0885	1.9389	120.7416	2.0124
11	Estacion 26	14	2019-03-28 19:40	2019-10-15 00:40	200.208333	54,119	232.4990	0.0032	6.5583	2.3220	4.1233	245.5058	4.0918
12	Estacion 26	14	2019-03-28 19:40	2020-01-15 00:40	292.208333	79,223	344.7402	0.0034	9.7701	3.3503	6.2492	364.1132	6.0686
13	Estacion 101	19	2018-09-25 20:10	2018-12-31 20:10	97	25,682	113.9750	0.0047	3.7560	1.3825	2.4682	121.5864	2.0264
14	Estacion 101	19	2018-09-25 20:10	2019-05-28 18:40	244.9375	64,921	282.4692	0.0030	9.7583	3.4286	6.3382	301.9973	5.0333
15	Estacion 101	19	2018-09-25 20:10	2019-08-28 18:40	336.9375	90,384	399.1689	0.0039	13.3276	4.8884	8.8727	426.2614	7.1044
16	Teledyne	18	2018-09-01 00:00	2019-03-01 00:00	181	88,379	391.0676	0.0042	8.3926	3.4936	5.2528	408.2109	6.8035
17	Teledyne	18	2018-09-01 00:00	2019-08-15 01:00	348.041667	132,427	589.4341	0.0039	12.3581	5.1125	7.1101	614.0187	10.2336
18	Teledyne	18	2018-09-01 00:00	2020-01-15 01:00	501.041667	172,971	757.7355	0.0044	15.9425	6.4051	9.4482	789.5356	13.1589

The Python license was certified as Open Source[1] and was compatible with the GPL[2]. Figure 3 shows a diagram of the framework's component operation.

The framework was built as a set of libraries, which performed specific functions; thus, when combined, they generated the homogenized output dataset with user selections. The operation was divided into four processes: start session, build query, homogenize, and generate dataset. These processes are detailed below.

Start Session. Initially, on the MongoDB server, the user collection was created to manage users and their working collections, which would be used during their session in the framework.

Later in the start session process, the user was registered and validated; the session was created and closed, and the working collections (sessionX and parametersX) were generated.

These working collections were maintained during the user's session. The parametersX collection stored the user selections in each of the levels represented in the XML metadata; thus, it saved the parameters needed to build the query with which the MongoDB information would be extracted.

The sessionX collection, on the other hand, contained the resulting homogenized dataset to be exported to csv, JSON, or XML. Note: The X at the end of the collections is a random number between 1 and 1000. That number was verified in MongoDB before creating the collections to avoid collisions.

Build Query. Once the session started, the elements to be included in the dataset had to be chosen. The build query process contained the set of libraries that made up data extraction query. Each library displayed each of the groups and subgroups of metadata items to choose from.

This way, the user first selected the stations, then the sensors to include per station, and finally the output's unit of measurement for each group of factors (each sensor belonged to a group of

[1] News from the blog — Open Source Initiative. opensource.org

[2] The GNU General Public License v3.0 - GNU Project - Free Software Foundation. www.gnu.org/licenses/gpl-3.0.html

Table 2. Frame homogenization process times for 6 stations and 103 sensors

Process of Framework												
						Time in Feconds						
							Generation File:					
Line	**Start date**	**End date**	**Days**	**Documents MongoDB**	**Embedded Documents**	**Generation Dataset**	**readme**	**json**	**csv**	**xml**	**Total time all files**	**Total time in minutes**
1	2019-10-01 00:00	2019-11-01 00:00	31.0000	43,295	8,929	41.1790	0.0039	4.8485	1.9920	3.3504	51.3739	0.8562
2	2018-09-01 00:00	2019-03-05 00:00	185.0000	144,188	90,143	391.2329	0.0037	20.6551	15.2575	14.4693	441.6185	7.3603
3	2019-02-08 02:05	2019-11-04 00:00	268.9132	327,738	77,445	361.3077	0.0040	39.0183	16.4658	27.0040	443.7999	7.3967
4	2018-09-25 00:00	2019-11-04 00:00	405.0000	444,441	146,663	664.7569	0.0034	57.4077	28.5741	39.5156	790.2577	13.1710
5	2018-01-27 00:00	2020-01-15 01:00	718.0417	571,358	219,505	991.5286	0.0046	73.1795	39.9118	49.6220	1,154.2465	19.2374
6	**2017-04-03 20:55**	**2020-01-15 01:05**	**1,016.1736**	**631,202**	**279,349**	**1,260.0749**	**0.0039**	**81.1401**	**48.3309**	**55.0631**	**1,444.6129**	**24.0769**
Average			437.3547	360,370	137,006	618.3467	0.0039	46.0415	25.0887	31.5041	720.9849	12.0164

conversion factors). The date range and the quality of data were part of data requested. This is how the selections and parameters were stored in the parametersX collection.

At this point, it was possible to change the chosen elements as many times as the user wished. Finally, data was extracted through the query to go on to the homogenization process.

Homogenization. During this phase, each of the documents extracted was analyzed. The value of the item by chosen quality was used as the input unit to be transformed into the chosen output unit. The homogenized result was stored in the sessionX collection.

Generate Dataset. For the generate dataset process, the user had already selected the output format for the dataset, which could be csv, JSON or XML, and the sessionX information had undergone a dataset construction process into the desired format.

In the generated dataset, the stations were embedded by datetime. Finally, another file was generated along with the dataset. It was a Readme.txt file which contained detailed information on the dataset content.

3.2.5 Evaluation

In the evaluation step, the framework execution times for dataset generation were shown, including their homogenization. As can be seen in line 9 of Table 1, the framework took 15,339 minutes to generate the queries, homogenize data, create the Readme.txt file, and generate the dataset in csv, JSON and XML. All data from Station 25 were included in this process: a total of 19 sensors and 194,377 documents, which corresponded to 1,016.15 days. Table 2 shows the frame run times including all 6 stations, all sensors for each station, and all time periods.

As can be seen in line 6, the total time in minutes used by the framework for the query-making and homogenized dataset generation processes was less than 24.0769 minutes. Although a total of 631,202 documents MongoDB were analyzed, when generating the dataset, only 279,349 embedded documents were created; this is because the documents were aligned by a timestamp.

In order to compare the manual process with the one carried out using the proposed framework, the following aspects were considered: the dataset generation time in minutes, the number of stations, the analysis time in days, and the homogenized data.

For the manual process, it can take 60 minutes to generate a single station dataset including 30 days of non-homogenized data. In contrast, the proposed framework used half the time to generate a dataset for six stations including up to 1,200 days of homogenized data. This represents a significant increase in data throughput.

Once the results were tested, it was observed that the framework is indeed an efficient solution since it decreases the dataset generation times considerably in comparison to the manual process.

In addition, it is a tool that can homogenize data, generate datasets in different formats that can be adapted to other advanced analysis tools, provide a profile of the generated dataset content, maintain data integrity by eliminating direct contact with them, and be implemented in a user-friendly environment.

4 Conclusions and Feature Research

It can be concluded that the methodology and the framework developed were effective in the case of application as they enabled efficient data loading and showed a considerable reduction in processing times while including the homogenization and generation of datasets in formats that are compatible with advanced analysis tools.

Finally, the proposed methodology developed a framework that contributes to several technological aspects, which will be explained in the following paragraphs. The framework provides a methodology for data management, including efficient extraction and loading, as well as for data conversion factors using metadata from an unstructured database.

In this case, MongoDB was used since it takes advantage of a dynamic structure to align the records by timestamp. Additionally, the framework achieves a considerable reduction in dataset generation times, including the homogenization process for ensuing analyses.

Furthermore, it creates datasets in different formats such as csv, XML and JSON, which were validated by experts. In addition, it ensures data integrity by avoiding their direct manipulation. The framework also contributes in terms of physical security by eliminating having to enter restricted spaces to obtain the required information.

Additionally, the solution was developed in the open-source Python language applying the AR methodology. Furthermore, it was developed using a standardized coding pattern, so new libraries can be easily added.

Likewise, it was proven portable since, due to the language used in its development, it was implemented in two environments such as command line and Django.

One advantage in using this framework is that it can be extended to other domains since this architecture design allows for its adaptation through the definition of metadata.

For example, it can migrate from structured to unstructured data and be implemented as a template in scenarios that require handling and transforming heterogeneous data, as well as providing files in different formats for further advanced analyses.

Regarding further research, several opportunities were identified. In defining metadata, a tool can be developed to simplify maintenance and to easily include additional information for any group.

As for the framework's areas of opportunity, libraries could be added for the creation of datasets in other output formats to support different time zones. Additional libraries can also feature other functions to meet the needs of the CECATEV meteorological center.

References

1. **Arora, R., Aggarwal, R. R. (2013).** An algorithm for transformation of data from MySQL to NoSQL (MongoDB). International Journal of Advanced Studies in Computer Science and Engineering, Vol. 2, No. 1, pp. 6–12.

2. **Brink, H., Richards, J. W., Mark, F. (2017).** Real-world machine learning. Manning Publications.

3. **BĂZĂR, C., IOSIF, C. S. (2014).** The transition from RDBMS to NoSQL. A comparative analysis of three popular non-relational solutions cassandra, MongoDB and couchbase. Database Systems Journal, Vol. 5, pp. 49–59.

4. **Camargo-Vega, J. J., Camargo-Ortega, J. F., Joyanes-Aguilar, L. (2015).** Conociendo big data. Facultad de Ingeniería, Vol. 24, No. 38.

5. **Chauhan, D., Bansal, K. L. (2017).** Using the advantages of NOSQL: A case study on MongoDB. International Journal on Recent and Innovation Trends in Computing and Communication, Vol. 5, No. 2, pp. 90–93.

6. **Checkland, P., Holwell, S. (1998).** Action research: Its nature and validity. Systemic Practice and Action Research, Vol. 11, No. 1, pp. 9–21. DOI: 10.1023/a:1022908820784.

7. **Chen, M., Mao, S., Liu, Y. (2014).** Big data: A survey. Mobile Networks and Applications, Vol. 19, No. 2, pp. 171–209. DOI: 10.1007/s11036-013-0489-0.

8. **Chicco, G. (2021).** Data consistency for data-driven smart energy assessment. Frontiers in Big Data, Vol. 4, pp. 1–19. DOI: 10.3389/fdata.2021.683682.

9. **Coughlan, P., Coghlan, D. (2002).** Action research for operations management. International Journal of Operations and Production Management, Vol. 22, No. 2, pp. 220–240. DOI: 10.1108/01443570210417515.

10. **Cruz, A., Antaño, M., Mario, J., Martínez-Castro, J. M., Cuevas-Valencia, R. (2014).** Migración de bases de datos SQL a NoSQL. Revista Tlamati Sabiduria, Vol. 5, pp. 144–148.

11. **Dao, M. S., Zettsu, K. (2015).** Discovering environmental impacts on public health using heterogeneous big sensory data. IEEE International Congress on Big Data, BigData Congress, pp. 741–744. DOI: 10.1109/BigDataCongress.2015.122.

12. **Diouf, P. S., Boly, A., Ndiaye, S. (2018).** Performance of the ETL processes in terms of volume and velocity in the cloud: State of the art. 4th IEEE International Conference on Engineering Technologies and Applied Sciences, ICETAS 2017, pp. 1–5. DOI: 10.1109/ICETAS.2017.8277875.

13. **Diouf, P. S., Boly, A., Ndiaye, S. (2018).** Variety of data in the ETL processes in the cloud: State of the art. IEEE International Conference on Innovative Research and Development, pp. 1–5. DOI: 10.1109/ICIRD.2018.8376308.

14. **Dresch, A., Pacheco-Lacerda, D., Cauchick-Miguel, P. A. (2015).** A distinctive analysis of case study, action research and design science research. Revista Brasileira de Gestao de Negocios, Vol. 17, No. 56, pp. 1116–1133. DOI: 10.7819/rbgn.v17i56.2069.

15. **Eden, C., Huxham, C. (1996).** Action research for management research. British Journal of Management, Vol. 7, No. 1, pp. 75–86. DOI: 10.1111/j.1467-8551.1996.tb00107.x.

16. **Giffard-Roisin, S., Yang, M., Charpiat, G., Kumler-Bonfanti, C., Kégl, B., Monteleoni, C. (2020).** Tropical cyclone track forecasting using fused deep learning from aligned reanalysis data. Frontiers in Big Data, Vol. 3, pp. 1–13. DOI: 10.3389/fdata.2020.00001.

17. **Jäger, S., Allhorn, A., Bießmann, F. (2021).** A Benchmark for data imputation methods. Frontiers in Big Data, Vol. 4, pp. 1–16. DOI: 10.3389/fdata.2021.693674.

18. **Jaybal, Y., Ramanathan, C., Rajagopalan, S. (2018).** HDSanalytics: A data analytics framework for heterogeneous data sources. Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, pp. 11–19. DOI: 10.1145/3152494.3152516.

19. **Jiménez, A., Nieve, J., Estrada, F., Vázquez-Gálvez, F. A., Hernández, I. (2019).** Management of heterogeneous data in the red climatológica UACJ in a NoSQL environment. IEEE International Autumn Meeting on Power, Electronics and Computing, pp. 1–6. DOI: 10.1109/ROPEC48299.2019.9057068.

20. **Liu, Q., Guo, X., Fan, H., Zhu, H. (2018).** A novel data visualization approach and scheme for supporting heterogeneous data. Proceedings of the 2nd IEEE Information Technology, Networking, Electronic and Automation

Control Conference, pp. 1259–1263. DOI: 10.1109/ITNEC.2017.8284978.

21. **Patil, M. M., Hanni, A., Tejeshwar, C. H., Patil, P. (2017).** A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retriewal operations using a web/android application to explore load balancing — Sharding in MongoDB and its advantages. International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), pp. 325–330. DOI: 10.1109/i-smac.2017.8058365.

22. **PowerData (2013).** Procesos ETL: Definición, características, beneficios y retos.

23. **Steinacker, A., Ghavam, A., Steinmetz, R. (2001).** Metadata standards for web-based resources. IEEE MultiMedia, Vol. 8, No. 1, pp. 70–76. DOI: 10.1109/93.923956.

24. **Stevenson, R. D., Suomela, T., Kim, H., He, Y. (2021).** Seven primary data types in citizen science determine data quality requirements and methods. Frontiers in Climate, Vol. 3. DOI: 10.3389/fclim.2021.645120.

25. **Stuivenvolt-Allen, J., Wang, S. S. Y. (2019).** Data mining climate variability as an indicator of US natural gas. Frontiers in Big Data, Vol. 2, pp. 1–6. DOI: 10.3389/fdata.2019.00020.

26. **Thiollent, M., Colette, M. (2020).** Pesquisa-ação, universidade e sociedade. Revista Mbote, Vol. 1, No. 1, pp. 042–066. DOI: 10.47551/mbote.v1i1.9382.

27. **Tiwari, P. (2017).** Improvement of ETL through integration of query cache and scripting method. Proceedings of the International Conference on Data Science and Engineering. DOI: 10.1109/ICDSE.2016.7823935.

28. **Volpi, D., Meccia, V. L., Guemas, V., Ortega, P., Bilbao, R., Doblas-Reyes, F. J., Amaral, A., Echevarria, P., Mahmood, R., Corti, S. (2021).** A novel initialization technique for decadal climate predictions. Frontiers in Climate, Vol. 3, pp. 1–14. DOI: 10.3389/fclim.2021.681127.

29. **Wurster, P. M., Maneta, M., Kimball, J. S., Endsley, K. A., Beguería, S. (2021).** Monitoring crop status in the continental United States using the SMAP level-4 carbon product. Frontiers in Big Data, Vol. 3, pp. 1–17. DOI: 10.3389/fdata.2020.597720.

30. **Xiang, L., Huang, J., Shao, X., Wang, D. (2016).** A MongoDB-based management of planar spatial data with a flattened R-tree. ISPRS International Journal of Geo-Information, Vol. 5, No. 7, pp. 119. DOI: 10.3390/ijgi5070119.

31. **Zeng, N., Zhang, G. Q., Li, X., Cui, L. (2017).** Evaluation of relational and NoSQL approaches for patient cohort identification from heterogeneous data sources. Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine, pp. 1135–1140. DOI: 10.1109/BIBM.2017.8217817.

32. **Zhong, M., Liu, M. (2009).** 3Se: A semi-structured search engine for heterogeneous data in graph model. Proceedings of the 18th ACM conference on Information and knowledge management. DOI: 10.1145/1645953.1646131.

Article received on 2023/01/09; accepted on 20/01/2024.
** Corresponding author is Alicia Margarita Jiménez-Galina.*

ISSN 2007-9737

PIONEER: An Interest-Aware POI Recommendation Engine

Sanjeev K. Cowlessur*,[1], Annappa Basava[2],
Bibudhendu Pati[3]

[1] Université des Mascareignes,
Pamplemousses,
Mauritius

[2] National Institute of Technology Karnataka,
Surathkal,
India

[3] Rama Devi Women's University,
Bhubaneswar,
India

scowlessur@adm.ac.mu

Abstract. Over the past decades, tourism has become a key economic industry for many countries. In today's global economy, it is an essential source of employment and revenue. Tourism as a leisure activity is a very popular form of recreation which involves the movement of people to foreign cities to visit new and unfamiliar places of interest (POIs). The task of recommending personalised tours for tourists is very demanding and time-consuming. The recommended tours must satisfy the tourist's interests and must at the same time be completed within a limited time span and within some budget. In existing itinerary recommender systems, if there is no past visit history about a particular POI, then that POI is not included in the recommended itinerary. To address this challenge, we have devised an algorithm called PIONEER which is based on a genetic algorithm for suggesting an itinerary based on tourist interests, POI popularity, and travel costs. Our algorithm recommends itineraries for tourists who want to visit locations which are unfamiliar to them. We have used the publicly available Flickr dataset in our work. The results demonstrate the superiority of our PIONEER algorithm compared to the baseline algorithms with regards to metrics like precision, recall and F1-Score.

Keywords. POI, tour recommendation, NSGA-II, multi-objective optimisation.

1 Introduction

Planning a visit to a foreign city can be a very daunting task [1]. The tourist needs to identify interesting POIs and then plan his/her visits as a connected itinerary while taking into account various spatial and temporal constraints. There are a number of factors which affect a tourist's decision and choice of visiting a particular POI [24].

Some of the factors are internal, that is, personal to the tourist, for example, age, education, occupation, income or his prior travel experiences and some are external meaning that they do not depend on the tourist, for example, climate and reviews from other travellers [9, 19].

In this work, we propose a recommendation engine for tourists which provides the most relevant suggestions for POI visits keeping in mind tourists' interests, popularity of POIs and travelling cost [22]. The remainder of this paper is organised as follows. Section 2 reviews and discusses a few relevant research work undertaken in the area. Then, section 3 provides definitions of necessary concepts.

doi: 10.13053/CyS-28-1-4454

In the following section 4, the problem is defined and the proposed new algorithm is explained in section 5. Then follows section 6 which contains a discussion on the various experiments conducted and the results obtained. Conclusions, ideas and suggestions for future research in the area under study is given in section 7.

2 Review of Related Work

Recently, tour recommendation has become a popular subject of interest among researchers [2]. Several applications [4, 18, 29, 30] have been built to deliver personalised tours.

2.1 The Orienteering Problem

Many tour recommender systems have as their starting point the Orienteering Problem [10, 21]. The idea orienteering problem came from a sports game which consisted of a number of checkpoints each having an associated score. Each player had to start at a given checkpoint, with a view to visit as many checkpoints as possible to accummulate scores.

The player who obtained the largest score in the smallest possible time was declared the winner. One constraint imposed was that each checkpoint had to be visited at most once. However, it was not mandatory for the player to start and end at the same point. In the past years, many researchers have been using the orienteering problem [11, 28] in their tour recommendation works.

2.2 Tour Recommendations based on the Orienteering Problem

In their paper, Choudhury et al. [7] proposed a tour itinerary based on the orienteering approach, in which the tourist begins the tour at some POI and finishes the at some other POI, where the goal was to recommend an itinerary comprising the most popular POIs, all within a given budget. Lim et al. [15] brought modifications to the orienteering problem by ensuring that the tourist visits one POI catergory he/she is interested in. Vansteenwegen et al. [27] proposed an approach for adapting the tour schedule so that it would improve the overall balance between the defined degree of involvement from the starting and end, such as expenditure and all POIs. Lim et al. [17] have identified places to visit which require minimum queuing time. Algorithms have also been developed to recommend tours for groups of tourists which satisfy the different levels of interest of each tourist within the group [1, 17].

2.3 Other Tourism Related Work

The wealth of information available in geo-tagged photos can be used to understand tourists' behaviour and to find out how popular a given POI is. Ji et al. [12] use a graphical model to evaluate the popularity of a POI by using photos which have been uploaded to websites.

The amount of time spent by a tourist at a given POI and in which order he/she visits the POIs can be extracted from the images data [20]. The authors in [6] have used geo-tagged photos to find out the location of clusters where popular and interesting activities are taking place.

In their paper Li et al. [14] were able determine the approximate location of photos [13]. A time aware measurement technique which considers a tourist's current location to recommend the next POI to visit was proposed by Ying et al. [31].

The $\mathrm{PIONEER}$ algorithm suggested in this paper differs significantly from the current POI and tour recommendation schemes in that this algorithm uses geo-tagged images to categorise the interests of visitors dynamically depending on time spent at a POI and its popularity.

3 Background

A tourist travelling to a any city across the globe will certainly be looking to visit Points Of Interest (POIs). We can think of a POI as a place which a person finds useful or interesting. Suppose there are n POIs in a given city denoted by: $p_1, p_2, p_3, \ldots, p_n$. Suppose each POI p_i belongs to a category $c_{p_i} \in \mathbb{C}$ associated with it, where $\mathbb{C}$ is the set of all categories of POIs (some examples of POI categories are: parks, museums, shops, restaurants).

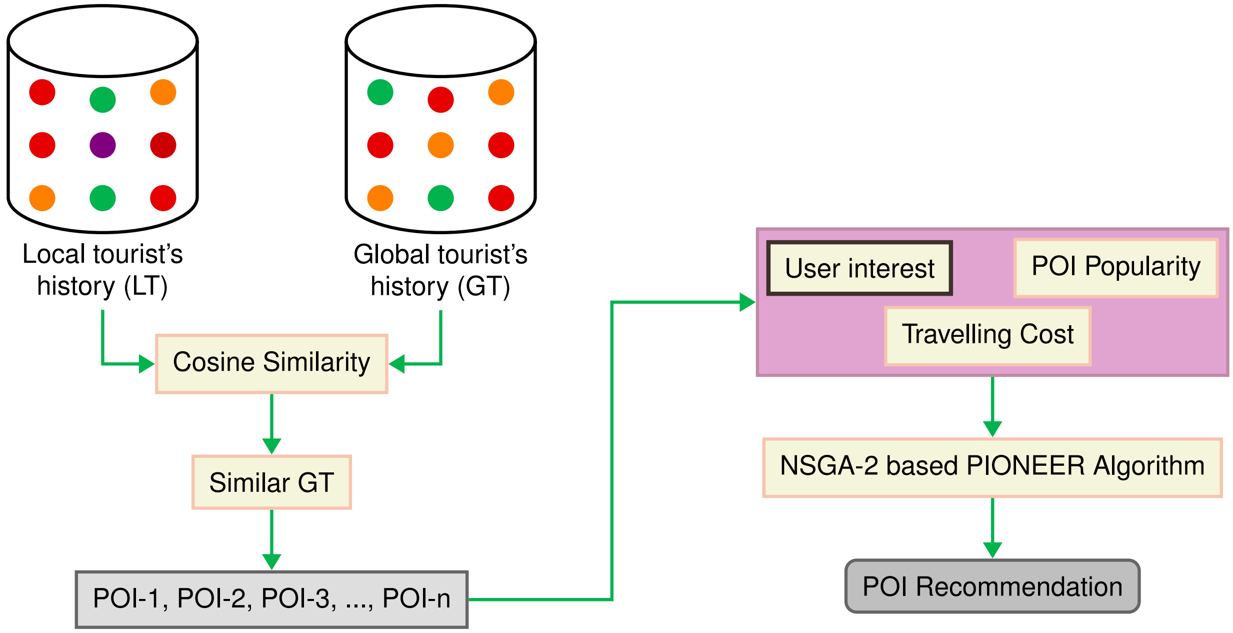

Local tourist's history (LT)
Global tourist's history (GT)
Cosine Similarity
Similar GT
POI-1, POI-2, POI-3, ..., POI-n
User interest
POI Popularity
Travelling Cost
NSGA-2 based PIONEER Algorithm
POI Recommendation

Fig. 1. System framework

3.1 Local and Global Tourists

In this study, tourists are classified into two different categories - local tourists and global tourists. A tourist who has already visited a certain city is referred to as a local tourist with respect to that city. If this tourist now travels to a different city which he has not previously visited, the local tourist of that city becomes a global tourist with respect to the travelling tourist.

To make the concepts clearer, consider two tourists T_1 and T_2 where T_1 has visited Mauritius but not Bangalore and T_2 who has visited Bangalore but not Mauritius. Then T_1 is a local tourist for Mauritius and T_2 is a local tourist for Bangalore. Suppose now that tourist T_1 wants to visit Bangalore. Then T_2 becomes the global tourist for T_1. Similarly, if T_2 flies to Mauritius then T_1 becomes the global tourist for T_2.

3.2 Travelling History of a Tourist

Let $\mathbb{U}$ be a set of tourists. Suppose there is some tourist $u \in \mathbb{U}$ who has so far visited k POIs. Then, the travelling history of u is given by a sequence of triples $\mathbb{H}_u = \big((p_1, t^a_{p_1}, t^d_{p_1}), \ldots, (p_k, t^a_{p_k}, t^d_{p_k})\big)$.

In the triple $(p_\iota, t^a_{p_\iota}, t^d_{p_\iota})$, p_ι is the POI visited by the tourist, $t^a_{p_\iota}$ is the time of arrival at POI p_ι and $t^d_{p_\iota}$ is the time of departure from p_ι. The difference between $t^a_{p_\iota}$ and $t^d_{p_\iota}$ gives the amount of time spent at p_ι. To make the notation simpler, we will write $\mathbb{H}_u = (p_1, \ldots, p_k)$ instead of $\mathbb{H}_u = \big((p_1, t^a_{p_1}, t^d_{p_1}), \ldots, (p_k, t^a_{p_k}, t^d_{p_k})\big)$.

3.3 Travelling Sequences of a Tourist

Given some tourist, his/her travel history is broken down into several distinct travel sequences if the time difference between two consecutive POI visits is t_{seq} hours or more. In our work we use $t_{\text{seq}} = 8$ as proposed by Lim in [16]. So, the travel history $\mathbb{H}^1_u$ can be written as $\mathbb{H}^1_u, \mathbb{H}^2_u, \ldots, \mathbb{H}^k_u$, where k is the number of travel sequences.

3.4 Average Time Spent at $a = \mathrm{POI}$

For every tourist, the history of his/her past travels is known. Given this information, the equation 1 can be used to calculate the mean time spent by all tourists who have visited a specific POI p [3, 5]. This value is denoted by $A(p)$:

Algorithm 1: $\mathrm{PIONEER}$ Algorithm

```
Data: $u$
Result: $(p_1,\, p_2,\, \ldots,\, p_n)$
$\mathcal{I} \leftarrow \emptyset$
for $u$ in $\mathrm{matching\_global\_tourist\_list}$ do
    $\mathcal{I} \leftarrow \mathcal{I} \cup \mathcal{I}_u$
$\mathcal{P}^{\text{initial}} \leftarrow \mathcal{I}$
$\mathcal{F}_1 \leftarrow \mathrm{Objective\ \_1}(\mathcal{P}^{\text{initial}})$
$\mathcal{F}_2 \leftarrow \mathrm{Objective\ \_2}(\mathcal{P}^{\text{initial}})$
$\mathcal{R} \leftarrow \mathrm{non\_dominated\_sorting}(\mathcal{F}_1,\, \mathcal{F}_2,\, \mathcal{P}^{\text{initial}})$
$\mathcal{D}_{\text{crowd}} \leftarrow \mathrm{find\_crowding\_distance}()$
$\mathcal{P}^{\text{sol}} \leftarrow \mathrm{select\_initial\_pop}(\mathcal{R},\, \mathcal{D}_{\text{crowd}})$
while termination_condition_not_reached do
    $\mathcal{C}^{\text{sol}} \leftarrow \mathrm{gen\_child\_pop}(\mathcal{P}^{\text{sol}})$
    $\mathcal{F}_1 \leftarrow \mathrm{Objective\_1}(\mathcal{C}^{\text{sol}})$
    $\mathcal{F}_2 \leftarrow \mathrm{Objective\_2}(\mathcal{C}^{\text{sol}})$
    $\mathcal{R} \leftarrow \mathrm{non\_dominated\_sorting}(\mathcal{F}_1,\, \mathcal{F}_2,\, \mathcal{P}^{\text{sol}} \cup \mathcal{C}^{\text{sol}})$
    $\mathcal{D}_{\text{crowd}} \leftarrow \mathrm{find\_crowding\_distance}()$
    $\mathcal{P}^{\text{sol}} \leftarrow \mathrm{select\_next\_gen}(\mathcal{R},\, \mathcal{D}_{\text{crowd}})$
return $\mathcal{P}^{\text{sol}}$
```

$$A(p) = \frac{\displaystyle\sum_{u=1}^{q} \sum_{\imath=1}^{r} (t^d_{p_\imath} - t^a_{p_\imath})\delta(p_\imath = p)}{\displaystyle\sum_{u=1}^{q} V_u\ \delta(p_\imath = p)}, \quad \forall p \in \mathcal{P}, \quad (1)$$

where $u = \{1, 2, \ldots, q\}$, $\jmath = \{1, 2, \ldots, r\}$, V_u is the frequency of tourist's u visit to POI p and $\delta(p_\imath = p) = 1$ if $p_\imath = p$ and 0 otherwise.

3.5 Tourist Interest for POI Category

Recall that the symbol $\mathbb{C}$ has been used to denote the set of categories of POIs and $c_p \in \mathbb{C}$ to denote the category of a POI p. Then, the interest a particular tourist u has for a particular category c of POI can be calculated using equation 2:

$$\mathrm{Int}_u(c) = \sum_{\jmath=1}^{n} \frac{(t^d_{p_\jmath} - t^a_{p_j})}{A(p_\jmath)} \delta(c_{p_\jmath} = c), \quad \forall c_p \in \mathbb{C}, \quad (2)$$

where $\delta(c_{p_\jmath}) = 1$ if $c_{p_\jmath} = c$ and 0 otherwise. The tourist interest for the POI category c is obtained from equation 1 by calculating the time spent by a tourist u at POI category c relative to the total time spent by all the tourists. It makes sense that a tourist will stay for a longer period at a POI category in which he/she is most interested in.

3.6 Local and Global Tourist Similarity

The degree of similarity between local and global tourists can be determined based on their interests for a given destination. For two distinct tourists u_x and u_y, we can compute their similarity using the cosine similarity measure as shown in equation 3:

$$\mathbb{S}(u_x,\ u_y) = \frac{\vec{\mathrm{Int}}_{u_x} \cdot \vec{\mathrm{Int}}_{u_y}}{||\vec{\mathrm{Int}}_{u_x} \cdot \vec{\mathrm{Int}}_{u_y}||}. \quad (3)$$

3.7 Tourist Interest for a POI

The interest a particular tourist u has for a particular POI p can be determined using the equation 4:

$$\mathrm{Int}_u(p) = \sum_{i=1}^{n} \frac{(t^a_{p_i} - t^d_{p_i})\delta(p_i = p)}{A(p_i)\delta(p_i = p)}, \quad (4)$$

where $A(p)$ (see equation 2) is the average time spent by all tourists at POI p and $\delta(p_i = p) = 1$ if $p_i = p$ and 0 otherwise.

3.8 Popularity of a POI

Every POI p has a certain popularity associated with it which is denoted by $\mathrm{Pop}(p)$. The popularity of a POI is taken to be the number of times the POI has been visited by all tourists. More formally, $\mathrm{Pop}(p)$ is defined by:

$$\mathrm{Pop}(p) = \sum_{u \in U} \Phi_{u,\ p}, \quad (5)$$

where U is the set of all tourists and $\Phi_{u,\ p}$ is the number of times tourist u has visited POI p.

3.9 Travelling Cost

There is a cost involved while travelling from one POI to another. In previous studies the cost of travel from one POI p_i to another POI p_j was a measure of the time taken by the tourist to complete the trip from p_i to p_j.

The total cost was considered to be the total time taken for an entire tour. The problem with using time as a measure of travel cost is that travel time depends on the means of transport used.

Table 1. Comparison of $\mathrm{Precision}$ between our proposed method and other baseline algorithms

Algorithms	PIONEER	TRIC	GREEPOP	GREENEAR	RAND
Delhi- Edinburgh	**0.589**±**0.027**	0.512±0.018	0.462±0.029	0.432±0.014	0.392±0.019
Osaka-Edinburgh	**0.613**±**0.025**	0.562±0.011	0.521±0.023	0.483±0.029	0.452±0.043
Vienna-Edinburgh	**0.692**±**0.013**	0.610±0.042	0.582±0.031	0.554±0.008	0.535±0.024
Delhi-Osaka	**0.593**±**0.029**	0.546±0.021	0.416±0.015	0.396±0.027	0.371±0.007
Glasgow-Edinburgh	**0.406**±**0.013**	0.336±0.029	0.307±0.006	0.281±0.035	0.263±0.014

For example if two POIs are very far apart and the tourist decides to take a flight, then the travel time will be much lower. In this paper, the total distance travelled by the tourist has been used as the travel cost and the aim is to minimise the distance travelled in an itinerary. The total cost of an itinerary $I = (p_1, p_2, \ \dots \ , p_N)$ with N POIs is given by:

$$\mathrm{Cost}(I) = \sum_{\iota=1}^{N-1} \mathrm{Dist}(p_\iota, \ p_{\iota+1}), \tag{6}$$

where $\mathrm{Dist}(p_\iota, \ p_j)$ is the distance between POIs p_ι and p_j which can be calculated using the Haversine formula [25].

4 Problem Definition

The main objective of this work is to suggest an itinerary $I_u = (p_1, \dots, p_n)$ for a tourist u such that the interests of the tourists and popularity of POIs visited are maximized but at the same time the cost of travel is minimised. This leads to the following optimisation problem [16]:

$$\mathbb{P}(I) = \sum_{i=1}^{n} \alpha \, \mathrm{Pop}(p_i) + (1-\alpha) \, \mathrm{Int}_u(c_{p_i}), \tag{7}$$

$$\mathbb{Q}(I) = \mathrm{Cost}(I), \tag{8}$$

where $\mathrm{Pop}(p_i)$ is the popularity of POI p_i, $\mathrm{Int}_u(c)$ refers to the interest tourist u has for POI category c, $\mathrm{Cost}(I)$ is the total distance between p_1 and p_n and α is a weight parameter which can be adjusted as required. The overall problem is thus:

$$\mathrm{Max}\left(\frac{\mathbb{P}(I)}{\mathbb{Q}(I)}\right). \tag{9}$$

Let $T_{p_i, p_j} = 1$, if the tourist travels directly from POI p_i to p_j and 0 otherwise [15]. The aim is to optimise equation 9 taking into consideration the constraints below:

$$\sum_{\jmath=2}^{N} T_{p_1, \ p_j} = \sum_{\iota=1}^{N-1} T_{p_\iota, \ p_N} = 1, \tag{10}$$

$$\sum_{\iota=1}^{N-1} T_{p_\iota, p_m} = \sum_{\jmath=2}^{N} T_{p_m, p_\jmath} \leq 1, \ \forall m = 2, \dots, N-1, \tag{11}$$

$$2 \leq p_\iota \leq N, \ \forall \iota = 2, \dots, N, \tag{12}$$

$$p_\iota - p_j + 1 \leq (N-1)(1 - T_{p_\iota, \ p_j}), \ \forall \iota, j = 2, \dots, N, \tag{13}$$

$$|\,\mathrm{cost}(I)| \leq B. \tag{14}$$

The limitation set out in eqn. 10 is to ascertain that the recommended itinerary starts at the first POI p_1 and finishes at the last POI p_n. The limitation in eqn. 11 ensures that no POI is visited more than once and that each POI in the itinerary is connected to the other. The restrictions imposed by eqns. 12 and 13 ensure that the proposed itinerary does not include a sub-itinerary 14. Equation 14 ensures that the cost of the itinerary does not exceed some budget B.

5 Proposed PIONEER Algorithm

The PIONEER algorithm proposed in this paper is based on the genetic algorithm called NSGA-II [8] which is a multi objective optimisation algorithm. The algorithm works as follows: Suppose a tourist u is travelling to a city c. A cosine similarity testis conducted between u and global tourists to obtain the top 10 matching global tourists.

Table 2. Comparison of Recall between our proposed method and other baseline algorithms

Algorithms	PIONEER	TRIC	GREEPOP	GREENEAR	RAND
Delhi- Edinburgh	**0.478±0.019**	0.386±0.015	0.359±0.009	0.342±0.038	0.316±0.023
Osaka-Edinburgh	**0.462±0.019**	0.372±0.029	0.346±0.022	0.319±0.036	0.291±0.011
Vienna-Edinburgh	**0.512±0.013**	0.414±0.020	0.388±0.019	0.358±0.031	0.343±0.041
Delhi-Osaka	**0.546±0.015**	0.463±0.013	0.431±0.032	0.407±0.007	0.378±0.025
Glasgow-Edinburgh	**0.372±0.003**	0.287±0.017	0.257±0.029	0.235±0.052	0.206±0.021

The list of POIs visited by the matching tourists and hence their itineraries are obtained from their travel histories. This list becomes the initial population $\mathcal{P}^{\text{initial}}$ (line 5) and is the input to the NSGA-II algorithm.

As defined in equations 7 and 8, popularity and interest of proposed itinerary must be maximised while at the same time minimising the cost. This leads to two objective functions:

$$\mathcal{F}_1 = \mathbb{P}(I), \tag{15}$$

$$\mathcal{F}_2 = \mathbb{Q}(I). \tag{16}$$

The two objective functions for every individual in the initial population $\mathcal{P}^{\text{initial}}$ are evaluated (lines 6-7) and each is assigned a rank and sorted into several fronts using a fast non-domination sorting method as described in [8] (line 8). Individuals belonging to the same front have the same rank. The crowding distance of each individual is determined from their objective values.

The parent population is selected from the initial population based on the rank and crowding distance. The genetic operations of selection, crossover and mutation are applied to the parent population to generate the child population $\mathcal{C}^{\text{sol}}$ (line 12). Fitness values of each itinerary in the child population are calculated (lines 13-14).

The child and parent lists are then combined (line 15) and a sorting algorithm is used to compare each itinerary with other itineraries using the criteria of nondominance and crowding distance. A natural selection is made by selecting all solutions belonging to the first fronts and discarding the others.

The algorithm stops when the maximum number of generations is reached.

6 Experiments

6.1 Dataset

This paper uses the YFCC100M (Yahoo! Flickr Creative Commons 100M) dataset [26]. It is a huge dataset comprising 100 million photos and videos obtained from Flickr.

From the metadata about the dataset information such as the date and time and the latitude and longitude values when the photos were taken and the ids of users who took the photos can be extracted.

6.2 Baseline Algorithms

- **Greedy Nearest (GREENEAR)**: The next POI to be visited is chosen at random from those POIs which are nearest, but which have not yet been visited.
- **Greedy Most Popular (GREEPOP)**: The next POI to be visited is chosen at random from these POIs which are the most popular, but which have not yet been visited.
- **Random Choice (RAND)**: The next POI to be visited is chosen at random from the set of POIs which have not yet been visited.
- **Tour Recommendation With Interest Category (TRIC)**: The recommended tour must include a compulsory category, which is the most frequently visited POI category in that city 18.

Table 3. Comparison of $\mathrm{F1\text{-}Score}$ between our proposed method and other baseline algorithms

Algorithms	PIONEER	TRIC	GREEPOP	GREENEAR	RAND
Delhi- Edinburgh	**0.528**±**0.033**	0.440±0.029	0.404±0.081	0.382±0.009	0.350±0.046
Osaka-Edinburgh	**0.527**±**0.010**	0.448±0.013	0.416±0.018	0.384±0.038	0.354±0.041
Vienna-Edinburgh	**0.589**±**0.046**	0.493±0.021	0.466±0.006	0.435±0.030	0.418±0.032
Delhi-Osaka	**0.569**±**0.016**	0.501±0.038	0.423±0.005	0.401±0.018	0.374±0.034
Glasgow-Edinburgh	**0.388**±**0.007**	0.310±0.026	0.280±0.041	0.256±0.027	0.231±0.018

6.3 Real-Life Evaluation

Only those tourists who have completed at least two travel sequences and visited at least two categories of POIs are used to evaluate the proposed algorithm.

The method is applied to both local and global datasets [23], as well as visitors who are comparable. We compare similar visitors in this study by looking at the top 10 associated visitors from global data sets.

For our experiments, categories of real travelling series are chosen based on the history of associated visitors in a given area. The standard evaluation metrics, that is, Precision, Recall and F1-Score have been used to test our algorithm.

– **Tour Recall (TourRec(I))**: Let C_{rec} be the list of POI categories suggested by our algorithm and let C_{real} be the list of all categories of POI which a tourist has visited in reality. Eqn. 17 defines $\mathrm{TourRecall}$, which returns the proportion of POI categories visited by a tourist which were also recommended by the algorithm:

$$\mathrm{TourRec}(I) = \frac{|C_{\text{rec}} \cap C_{\text{real}}|}{|C_{\text{real}}|}. \tag{17}$$

– **Tour Precision (TourPre(I))**: Let C_{rec} be the list of POI categories suggested by the algorithm and let C_{real} be the set of POI categories visited by a tourist in reality. $\mathrm{TourPrecision}$ is defined as the ratio of proposed POI categories which are also found in the tourist's actual travel history. $\mathrm{TourPrecision}$ is defined as follows:

$$\mathrm{TourPre}(I) = \frac{|C_{\text{rec}} \cap C_{\text{real}}|}{|C_{\text{rec}}|}. \tag{18}$$

– **Tour F1-Score (TourF1-score(I))**: The mean harmonic value of $\mathrm{Precision}$ and Recall for the proposed itinerary I is referred to as $\mathrm{Tour\ F1\text{-}Score}$ (Eqn. 19):

$$\mathrm{Tour\ F1\text{-}score}(I) = \frac{2 \times \mathrm{TourPre}(I) \times \mathrm{TourRec}(I)}{\mathrm{TourPre}(I) + \mathrm{TourRec}(I)}. \tag{19}$$

6.4 Comparison of Precision, Recall and F1-Score

The proposed $\mathrm{PIONEER}$ algorithm performs better when compared to other baseline algorithms such as GREEPOP, TRIC, GREENEAR and RAND. Tables 1, 2 and 3 show how the $\mathrm{PIONEER}$ algorithm compares with other baseline approaches in terms of $\mathrm{Precision}$, Recall and $\mathrm{F1 - Score}$ values. The results show that $\mathrm{PIONEER}$ fares better than the baseline approaches as far as $\mathrm{Precision}$, Recall and $\mathrm{F1\text{-}Score}$ metrics are concerned.

Recall measurements depending upon $|C_{\text{rec}}|$ and $|C_{\text{rec}} \cap C_{\text{real}}|$ as per in Eqn. 17. Here the values of $|C_{\text{rec}} \cap C_{\text{real}}|$ is better compared to the various baseline approaches which can be computed utilizing the $\mathrm{PIONEER}$ algorithm. Typically, the suggested $\mathrm{PIONEER}$ algorithm is based on local as well as global datasets, and ultimately suggests many POIs, resulting in better Recall scores for various baseline approaches.

For the $\mathrm{PIONEER}$ algorithm, the $\mathrm{Precision}$ scores are more because they are dependent on $|C_{\text{rec}}|$ and $|C_{\text{rec}} \cap C_{\text{real}}|$ as per in Eqn. 18. We found that C_{rec} scores vary for various baseline approaches during the analysis. The values of $|C_{\text{rec}} \cap C_{\text{real}}|$ are higher for the suggested $\mathrm{PIONEER}$ algorithm.

The $\mathrm{F1\text{-}Score}$ increase in the suggested $\mathrm{PIONEER}$ algorithm based on $\mathrm{Precision}$ and Recall compared to other baseline approaches.

7 Conclusion and Future Research

This research has presented a new method called $\mathrm{PIONEER}$ which recommends tourist itineraries which maximise tourist interest, POI popularity while at the same time reducing cost. The algorithm uses the actual travel patterns of tourists which are obtained from geo-tagged photos.

From the dataset, tourists' interests, tour popularity and travelling costs are calculated for training the $\mathrm{PIONEER}$ algorithm. The suggested method is dependent on the selection of many POIs by taking into account the POI time visiting factor. $\mathrm{PIONEER}$ will not depend on the travelling history of a certain individual in new locations.

The case in which a visitor wants to visit new places is therefore taken into consideration. $\mathrm{PIONEER}$ is compared with various baselines using multiple criteria such as $\mathrm{Precision}$, Recall, and $\mathrm{F1\text{-}Score}$. The findings of the study demonstrate that the suggested algorithm surpasses baseline approaches.

This research will be extended in the future to cater for tourists who travel in groups (e.g., with family and friends) where the challenge is to cater for the individual interests and preferences of each group member.

References

1. **Anagnostopoulos, A., Atassi, R., Becchetti, L., Fazzone, A., Silvestri, F. (2017).** Tour recommendation for groups. Data Mining and Knowledge Discovery, Vol. 31, No. 5, pp. 1157–1188. DOI: 10.1007/s10618-016-0477-7.

2. **Borràs, J., Moreno, A., Valls, A. (2014).** Intelligent tourism recommender systems: A survey. Expert Systems with Applications, Vol. 41, No. 16, pp. 7370–7389. DOI: 10.1016/j.eswa.2014.06.007.

3. **Brilhante, I., Macedo, J. A., Nardini, F. M., Perego, R., Renso, C. (2014).** Tripbuilder: A tool for recommending sightseeing tours. Proceedings of the European Conference on Information Retrieval, Advances in Information Retrieval, Vol. 8416, pp. 771–774. DOI: 10.1007/978-3-319-06028-6_93.

4. **Castillo, L., Armengol, E., Onaindia, E., Sebastia, L., Gonzalez-Boticario, J., Rodriguez, A., Fernandez, S., Arias, J., Borrajo, D. (2008).** Samap: An user-oriented adaptive system for planning tourist visits. Expert Systems with Applications, Vol. 34, No. 2, pp. 1318–1332. DOI: 10.1016/j.eswa.2006.12.029.

5. **Chen, C., Zhang, D., Guo, B., Ma, X., Pan, G., Wu, Z. (2015).** TripPlanner: Personalized trip planning leveraging heterogeneous crowdsourced digital footprints. IEEE Transactions on Intelligent Transportation Systems, Vol. 16, No. 3, pp. 1259–1273. DOI: 10.1109/tits.2014.2357835.

6. **Crandall, D. J., Backstrom, L., Cosley, D., Suri, S., Huttenlocher, D., Kleinberg, J. (2010).** Inferring social ties from geographic coincidences. Proceedings of the National Academy of Sciences, Vol. 107, No. 52, pp. 22436–22441. DOI: 10.1073/pnas.1006155107.

7. **De-Choudhury, M., Feldman, M., Amer-Yahia, S., Golbandi, N., Lempel, R., Yu, C. (2010).** Automatic construction of travel itineraries using social breadcrumbs. Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, pp. 35–44. DOI: 10.1145/1810617.1810626.

8. **Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002).** A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, Vol. 6, No. 2, pp. 182–197. DOI: 10.1109/4235.996017.

9. **Dumitrescu, L., Fuciu, M. (2015).** Consumer behaviour in the tourist segmentation process – a marketing research. Studies in Business

and Economics, Vol. 10, No. 1, pp. 66–76. DOI: 10.1515/sbe-2015-0005.

10. **Golden, B. L., Levy, L., Vohra, R. (1987).** The orienteering problem. Naval Research Logistics, Vol. 34, No. 3, pp. 307–318. DOI: 10.1002/1520-6750(198706)34:3⟨307::aid-nav3220340302⟩3.0.co;2-d.

11. **Gunawan, A., Lau, H. C., Vansteenwegen, P. (2016).** Orienteering problem: A survey of recent variants, solution approaches and applications. European Journal of Operational Research, Vol. 255, No. 2, pp. 315–332. DOI: 10.1016/j.ejor.2016.04.059.

12. **Ji, R., Xie, X., Yao, H., Ma, W. Y. (2009).** Mining city landmarks from blogs by graph modeling. Proceedings of the 17th Association for Computing Machinery International Conference on Multimedia, pp. 105–114. DOI: 10.1145/1631272.1631289.

13. **Kisilevich, S., Mansmann, F., Keim, D. (2010).** P-DBSCAN: A density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos. Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research and Application, pp. 1–4. DOI: 10.1145/1823854.1823897.

14. **Li, J., Qian, X., Tang, Y. Y., Yang, L., Mei, T. (2013).** GPS estimation for places of interest from social users' uploaded photos. IEEE Transactions on Multimedia, Vol. 15, No. 8, pp. 2058–2071. DOI: 10.1109/tmm.2013.2280127.

15. **Lim, K. H. (2015).** Recommending tours and places-of-interest based on user interests from geo-tagged photos. Proceedings of the Association for Computing Machinery Special Interest Group on Management of Data on PhD Symposium, pp. 33–38. DOI: 10.1145/2744680.2744693.

16. **Lim, K. H., Chan, J., Karunasekera, S., Leckie, C. (2017).** Personalized itinerary recommendation with queuing time awareness. Proceedings of the 40th International ACM Special Interest Group on Information Retrieval Conference on Research and Development in Information Retrieval, pp. 325–334. DOI: 10.1145/3077136.3080778.

17. **Lim, K. H., Chan, J., Leckie, C., Karunasekera, S. (2016).** Towards next generation touring: Personalized group tours. Proceedings of the International Conference on Automated Planning and Scheduling, Vol. 26, pp. 412–420. DOI: 10.1609/icaps.v26i1.13775.

18. **Lim, K. H., Chan, J., Leckie, C., Karunasekera, S. (2017).** Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. Knowledge and Information Systems, Vol. 54, No. 2, pp. 375–406. DOI: 10.1007/s10115-017-1056-y.

19. **Majumder, A., Sarkar, J. L., Pati, B., Panigrahi, C. R., Ramasamy, V., Roy, S., Kumar, V. (2022).** MERIT: Multi-itinerary tourist recommendation engine for industrial internet of things. Proceedings of the IEEE INFOCOM 2022 IEEE Conference on Computer Communications Workshops, pp. 1–6. DOI: 10.1109/INFOCOMWKSHPS54753.2022.9798002.

20. **Popescu, A., Grefenstette, G., Moëllic, P. A. (2009).** Mining tourist information from user-supplied collections. Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 1713–1716. DOI: 10.1145/1645953.1646211.

21. **Sarkar, J. L., Majumder, A. (2021).** A new point-of-interest approach based on multi-itinerary recommendation engine. Expert Systems with Applications, Vol. 181, pp. 115026. DOI: 10.1016/j.eswa.2021.115026.

22. **Sarkar, J. L., Majumder, A. (2022).** gTour: Multiple itinerary recommendation engine for group of tourists. Expert Systems with Applications, Vol. 191, pp. 116190. DOI: 10.1016/j.eswa.2021.116190.

23. **Sarkar, J. L., Majumder, A., Panigrahi, C. R., Roy, S. (2020).** MULTITOUR: A multiple itinerary tourists recommendation engine. Electronic Commerce Research and Applications, Vol. 40, pp. 100943. DOI: 10.1016/j.elerap.2020.100943.

24. **Sarkar, J. L., Majumder, A., Panigrahi, C. R., Roy, S., Pati, B. (2022).** Tourism recommendation system: A survey and future research directions. Multimedia Tools and Applications, Vol. 82, No. 6, pp. 8983–9027. DOI: 10.1007/s11042-022-12167-w.

25. **Sinnot, R. W. (1984).** Virtues of the haversine. Sky and Telescope, Vol. 68, No. 2, pp. 159.

26. **Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L. J. (2016).** YFCC100M: The new data in multimedia research. Communications of the ACM, Vol. 59, No. 2, pp. 64–73. DOI: 10.1145/2812802.

27. **Vansteenwegen, P., Souffriau, W., Berghe, G. V., Oudheusden, D. V. (2011).** The city trip planner: An expert system for tourists. Expert Systems with Applications, Vol. 38, No. 6, pp. 6540–6546. DOI: 10.1016/j.eswa.2010.11.085.

28. **Vansteenwegen, P., Souffriau, W., Oudheusden, D. V. (2011).** The orienteering problem: A survey. European Journal of Operational Research, Vol. 209, No. 1, pp. 1–10. DOI: 10.1016/j.ejor.2010.03.045.

29. **Vansteenwegen, P., Van Oudheusden, D. (2007).** The mobile tourist guide: An OR opportunity. OR Insight, Vol. 20, No. 3, pp. 21–27. DOI: 10.1057/ori.2007.17.

30. **Wörndl, W., Hefele, A. (2016).** Generating paths through discovered places-of-interests for city trip planning. Information and Communication Technologies in Tourism, pp. 441–453. DOI: 10.1007/978-3-319-28231-2_32.

31. **Ying, H., Wu, J., Xu, G., Liu, Y., Liang, T., Zhang, X., Xiong, H. (2018).** Time-aware metric embedding with asymmetric projection for successive POI recommendation. World Wide Web, Vol. 22, No. 5, pp. 2209–2224. DOI: 10.1007/s11280-018-0596-8.

Article received on 02/01/2023; accepted on 12/11/2023.

** Corresponding author is Sanjeev K. Cowlessur.*

ISSN 2007-9737

Different Applications of the Gyroscope Sensors Data Fusion in Distinctive Systems: An Extended Kalman Filter Approach

Amir Naderolasli*, Hamidreza Shirzadfar

Shahid Ashrafi Esfahani University,
Department of Electrical and Biomedical Engineering,
Faculty of Engineering and Technology, Isfahan,
Iran

amir.naderolasli@gmail.com

Abstract. Data fusion systems are applied greatly in the militaries industry, medical equipments and other multi-sensors systems. Here, the practical approaches of data fusion like Kalman filter (KF), support vector machine (SVM) and data fusion are surveyed for the noisy multi-sensors systems.The angular velocity quantum is one of the practical parameters in the different systems in which the data fusion problem is suggested for the measuring of them. For this purpose, two gyroscopes with a same structure of dynamic model and different parameters are utilized that the Gussian noises with zero-mean and different variances are applied to both of them to assessment the gyroscope sensors data fusion problem. The gyroscope outputs are estimated through the Kalman filter approach. This suggested structure of the sensors data fusion is evaluated for the systems' outputs. The convergence rate of Kalman filter coefficients and the covariance error are compared among three suggested structures of sensors data fusion. The simulation results survey the effectiveness of gyroscope sensors data fusion such that the obtained data by using multi-sensors is more applicable than a single-sensor.

Keywords. Data fusion, Kalman filter (KF), support vector machine (SVM), angular velocities, gyroscope sensors.

1 Introduction

Data fusion systems are applied widely in the sensor networks, robot systems, video-processing, images-processing and intelligent design systems. The emergence of new sensors, advanced processing methods and improved processing hardware allow researchers to develop the data fusion.

The data fusion is proposed for the military purposes, target trajectory tracking and medical engineering [8]. This data fusion is a process that the obtained information from various sensors systems and states observers are combined to provide the drawing accurate decision precision.

The problem of data fusion for multi-sensor systems is based on the speed of the samples with equal sampling rates resulting in relatively simple data integration with limited application. The input variables for the data fusion systems include sensor information, command signals and previous data. The data fusion problem provides a powerful tool for information and draws decision [24].

Therefore, a combination of data analysis from multiple sensors to increase knowledge of the system is challenging task [16]. The data fusion of multi-sensor measurements is a challenging task as providing an estimation of the states vector over a sensor. Among the many techniques for multi-sensor data fusion, artificial intelligence, pattern recognition and statistical estimations are of the most importance of these attitudes [20].

There exist particular problems in the data fusion like the presence of non-proportional sensors, the emission of signals and noise environments [12].

The most fundamental fusion characteristics are the transfer functions of the dynamic model among the observed states and the multi-sensor parameters and the decision or inference. A quantitative assessment of the data fusion

doi: 10.13053/CyS-28-1-4308

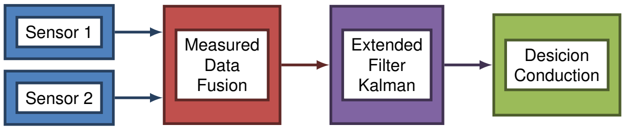


Fig. 1. The suggested block-diagram of data fusion through an extended Kalman Filter for two gyroscopes

systems is obtained usually through Monte Carlo simulations or analysis techniques of covariance error [11]. The combination of multiple sensors is a process of combining information that is gathered from different sensors and takes place at three levels of data, features, and decision making.

In the composition of data levels, all unfeasible data measured by multiple sensors is combined directly from a factor to generate useful information. Consequently, the selection of the described model depends on the function of the sensor combination and there exist no predefined model for data fusion.

In this article, Kalman filter is utilized for the continuous linear models of sensors in categorizing the state sequence and constitute with two dynamic models of sensors in the data fusion problem. This article is organized as follows. Different methods of data fusion are introduced in section 2. Examples of data fusion application in the different systems are surveyed in section 3. Experimental example of data fusion method with using an extended Kalman filter approach is utilized in section 4 and the article is concluded in section 5.

2 Different Methods of Data Fusion

2.1 Data Fusion by Applying Kalman Filter

Kalman filtering is widely applied in applications like optimization, estimation methods, filtering ability, measurement uncertainties, and mathematical models in the modern techniques of pursuing multi-sensory objectives [5]. The Kalman filter is a data recursive algorithm that estimates the uncertain system dynamics.

This algorithm is implemented subject to two processes of prediction of states based on the mathematical model and state correction based on the measured data from the sensors.

In practice, the Kalman filter is implemented with the assumptions that the systems is linearized by a linear dynamic model and is usually incompatible with the modeling error caused by the linearization. All the system dynamics and noise processes are specified exactly in the Kalman filter.

The combination of measured values is preferable to the state-vector combination when data fusion multi-sensor is based on the Kalman filter. The state-vector data fusion methods will be effective when the Kalman filter is stable, which limits the practical applications of data fusion methods; consequently, in many algorithms, the information is optimized through the Kalman filter.

The extended Kalman filter combines the measured data from multiple sensors to provide optimal results in the accumulated error terms, functionality, and frequency response. If this model is fully adjusted, the residue tends to be zero between the predicted and actual bode and converge within a bounded range. The extended Kalman filter equations are presented as follows:

$$x_{k+1} = f(x_k, k) + w_k, \tag{1}$$

$$P_k = (I - K_k\, H)\bar{P}_k + Q_k, \tag{2}$$

where K_k is obtained as follows:

$$K_k = P_k\, H_k^T [H_k\, P_k\, H_k^T + R_k]^{-1}, \tag{3}$$

$$\hat{x_k} = \hat{\bar{x}}_k + K_k [z_k - H_k\, \hat{\bar{x}}_k], \tag{4}$$

where, P_k is the error covariance matrix, K_k is an extended Kalman filter coefficient, Q_k is the discrete noise matrix, H_k is the attitude matrix and R_k is the attitude noise matrix. There exist two approaches for the combining multivariate data fusion based on the Kalman filter: in the first, the data taken from multiple sensors are simply merged with the Kalman filter estimator vector and in the second, the provided data by the sensors are

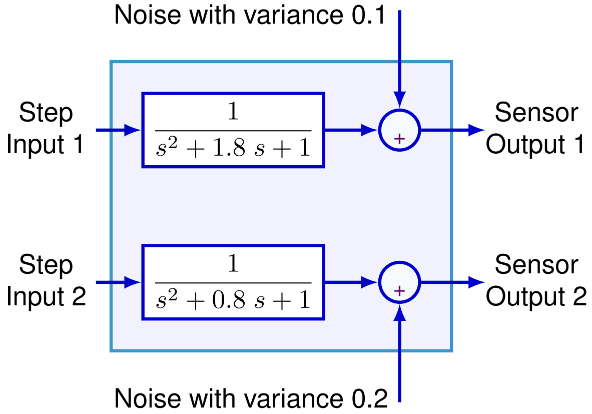


Fig. 2. The suggested block-diagram for the data fusion between two gyroscope sensors in the measuring angular velocities

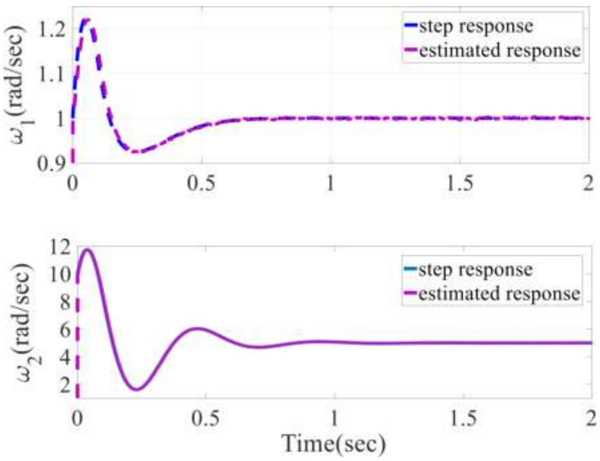


Fig. 3. The step response of data fusion between the two gyroscope sensors

combined according to the LSE criterion [10]. To estimate the online parameters, an extended Kalman filter provides the best estimations for the linear transfer function model.

The general block-diagram of the data fusion is proposed through an extended Kalman filter method as Figure 1. In this article, the values of angular velocities are calculated through an extended Kalman filter for two gyroscope sensors with different transfer function models.

2.2 Data Fusion Using the Support Vector Machine (SVM)

The divergence of the modeling errors is very important in the Kalman filter transfer functions. All the system state variables or estimation errors matrices are unrealistically dimmed and the Kalman filter loses its efficiency when the measured values do not provide enough information to be estimated. Moreover, the estimates can be applied to solve the modeling errors due to the divergence.

This issue increases the complexity of the Kalman filter, and can not guarantee that all the unstable states of the transfer function are the real models. The data fusion problem is addressed by the SVM algorithm that formulates the decisions to separate the different regions [25]. The support vector machine (SVM)-based multi-sensor data-processing system extracts features from the measured data [1].

A non-linear SVM provided through the kernel function when the obtaining data from the sensors does not change linearly becomes necessary [4]. A sensory combination method for assessing the physical activity of human beings based on SVM, which is an effective in the reducing system changes and, in particular, when the obtained data is augmented from the sensors to the combination transfer function model is proposed.

A hybrid method for categorizing the error signal based on the combination of sensor data through a SVM is proposed and a short-term Fourier transform transformation (STFT) technique is utilized. One of the advantages of SVM is the low number of parameters necessary for computation by the user where most of the parameters are determined by the internal algorithm. Therefore, the computational complexity of the SVM is subject to the number of data points relative to the system dimension.

2.3 Data Fusion for the Noise Systems

The data fusion problem is a challenging issue due to the existence of inaccurate and misleading data, contradictory data, data correlation, operation time, data dimensions and noise data [26].

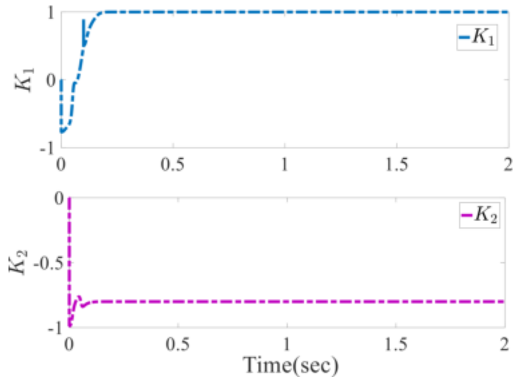


Fig. 4. The estimated transfer function coefficients through the extended Kalman filter

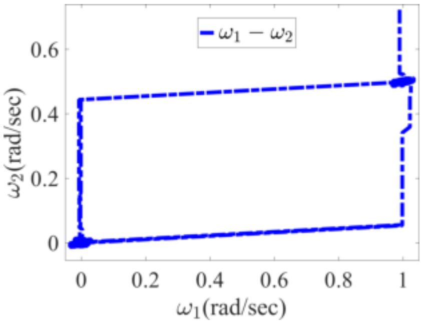


Fig. 5. The variation of the estimated transfer function for obtaining angular velocities between two gyroscope sensors

The combination of inaccurate dates like noise can have an adverse effects on the parameters estimation of transfer function. The common filtering techniques, based on the Kalman filters, rely on Gaussian noise and linear models that are not appropriate for the noise environments, accordingly, decisions are made on the basis of the hypotheses given where a decision test is made to ignore unreliable sensors [9].

3 Examples of Data Fusion Application in the Distinctive Systems

Data fusion is usually run in the form of a matrix method that separates the data sets from the higher-order matrices [27]. To assess the efficiency of the data combination in a random manner, some coefficients are changed to determine the effectiveness of the proposed method in the determining measurement parameters from inaccurate data. A new method for thecoefficients estimation is presented by applying a combination of two primary acceleration and pressure sensors in [23].

A high-level sensor combination is applied to get information from an unreachable sensors [7]. An interpolation data combination operator weighed by a sample of specific coefficients is suggested such that this method is not limited to clustering, classification, pattern identification, group decision methods, and data combinations [2].

An integrated algorithm is devised to combine the data fusion to produce parameters estimates some of the qualitative parameters in [6]. For many applications, information is provided by special sensors that are incomplete, inaccurate, and invalid [3].

The problem of data combinations for system instability, multi-speed, and multi-sensor linear systems are assessed in [22], where, the transfer function model is only specified for the best sampling rate, and is developed to combine the data from multi-sensor systems through the Kalman filter. A multi-sensor combination algorithm is proposed for the surgical surveillance with a human body that combines devices and explanatory data at a single moment [21].

4 An Experimental Example of the Data Fusion Method Using an Extended Kalman Filter Method

In this article, the data fusion is applied on some multiple gyroscope sensors to estimate the state variables. One of the most common factors in the experimental systems is the control and evaluation of angular velocities, which is measured by the gyroscope sensors [17]. To measure the angular velocities, the gyroscope sensors are modeled with a suggested second order transfer function, that is expressed as follows [18, 14]:

$$G_{\text{gyro}}(s) = \frac{\omega_n^2}{s^2 + 2\,\eta\,\omega_n\,s + \omega_n^2}, \tag{5}$$

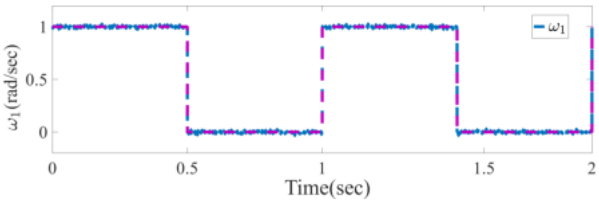


Fig. 6. The consecutive pulse response of gyroscope sensors through extended Kalman filter in spite of the applied noise

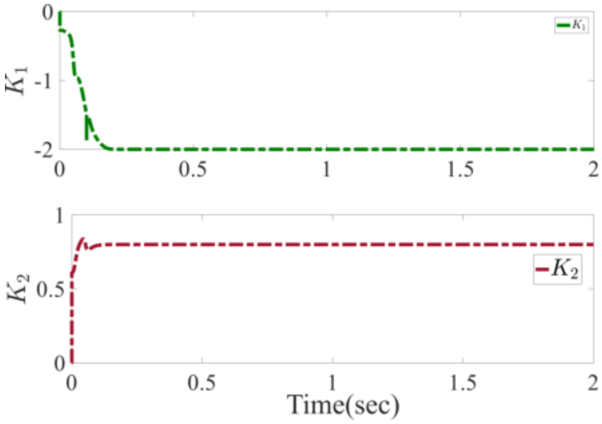


Fig. 7. The calculated coefficients by an extended Kalman filter approach for the obtaining angular velocities

where, parameters η and ω_n denote the damping rate and nature frequency, respectively. The parameters of gyroscope transfer function model G_{gyro1} and G_{gyro2} are specified with respect to the times and are converted into a discrete form through mapping $z = e^{ST}$ with a sampling time of $t = 0.01$ [15].

In order to evaluate the problem as best as possible, two different variances are determined based on the gyroscope sensors. As a result, the equations of the proposed model are presented for the gyroscope simulation with different systems damping coefficients as follows:

$$G_{\text{gyro1}} = \frac{1}{s^2 + 1.8s + 1}, \tag{6}$$

$$G_{\text{gyro2}} = \frac{1}{s^2 + 0.8s + 1}. \tag{7}$$

The Gaussian noise with zero mean and different variances 0.1 and 0.2 are augmented to evaluate the effects of the proposed gyroscope transfer function model, the measured values and the applied noise [13, 19].

The suggested block-diagram of data fusion between the two gyroscope sensors in the measuring angular velocities is drawn as Figure 2. In this section, three different data fusion structures are proposed for the measuring angular velocities between the two gyroscope sensors and their results are compared:

$$f(x) = \begin{cases} \mathrm{DF}_1 & = \dfrac{\omega_1 + \omega_2}{2}, \\ \mathrm{DF}_2 & = \dfrac{2\omega_1 + \omega_2}{3}, \\ \mathrm{DF}_3 & = \dfrac{\omega_1 + 2\omega_2}{3}. \end{cases} \tag{8}$$

The suggested extended Kalman filter algorithm is applied to combine the obtained data from two gyroscope sensors according to the proposed data fusion structures.

The step response of the gyroscope sensors and the estimated transfer function coefficients through an extended Kalman filter are shown in Figures 3 and 4, respectively.

The covariances in a steady state of the data-fusion methods are evaluated by the changing process covariance to provide the estimated coefficients for drawing accurate decision precision.

The estimated transfer functions are calculated from an extended filter Kalman for the measuring angular velocities between the two gyroscope sensors as follows:

$$\hat{G}_{\text{gyro1}} = \frac{0.6}{1 - 1.3z^{-1} + 0.9z^{-2}}, \tag{9}$$

$$\hat{G}_{\text{gyro2}} = \frac{1}{1 - 0.6z^{-1} + 0.9z^{-2}}. \tag{10}$$

The results of the data combination for the measured values in the various transfer function models are tabulated in Table 1, where, the terms

Table 1. The comparison scenario between different modes of the gyroscope sensors data fusion

States	C1	K_k
S1	0.071	0.65
S2	0.056	0.51
DF1	0.073	0.25
DF2	0.022	0.38
DF3	0.021	0.29

Table 2. The comparison between Kalman filter coefficients and errors variance for the different noise covariance

Q	var	K_k
e^{-5}	0.07	0.01580
e^{-4}	0.23	0.04370
e^{-3}	0.65	0.01317
e^{-2}	0.17	0.35820
e^{-1}	0.03	0.73220

$S1$ and $S2$ are the estimated data through an extended Kalman filter approach for the data fusion of two gyroscope sensors 1 and 2, K_k is an extended Kalman filter coefficient and $C1$ is the covariance error.

The error variance (var) for the different noise covariance (Q) is tabulated in Table 2. The variation of the estimated transfer function for obtaining angular velocities of data fusion between two gyroscope sensors is illustrated in Figure 5.

The numerical results indicate that the errors variance and extended Kalman filter coefficients increase for the convergence rate with increasing in the noise covariance of different states.

By increasing the number of gyroscope sensors in the multi-sensor transfer function models, algorithms based on ordinary Kalman filters will lead to more calculations and low resistance.

The consecutive pulse response of gyroscope sensors and the calculated coefficients by an extended Kalman filter approach are illustrated for the obtaining angular velocities in spite of the applied noise in Figures 6-7, respectively.

The evaluation of the tables 1 and 2, show that the combination of data between the two gyroscope sensors will have more flexibility in relation to the proposed structures than a sensor.

The comparison scenario between extended Kalman filtering coefficients (K_k). By increasing the covariance of noise in the algorithm, the error variance and the Kalman interest coefficient increase.

From the obtained data fusion of the gyroscope multi sensors for different responses in tables 1 and 2, the selection of the described transfer functions of the dynamic models is correctly selected.

Consequently, the combined data is more accurate and reliable than the received data from a single sensor and all these features are combined for the logical decisions.

The combination of decision-making is the highest level of composition that incorporates the combination of different sensors and implies the importance of classifying the composition for the best result.

5 Conclusion and Future Work

The results of the various experiments indicate that the modeling and analyzing based on the data fusion and multi-sensors are of a higher degree than modeling based on a sensor system.

The data analysis is subjected to the applied technique, the numbers of sensors and the working conditions. Despite all the proper Kalman filtering capabilities for data fusion, the great numbers of modes are necessary for accurate estimation with no ability to determine the parameters changes.

To measure the angular velocities, the gyroscope sensors are modeled with a suggested second order transfer function and the parameters of these suggested models are calculated through an extended Kalman filter. These obtained models are investigated in the three structures of data fusion to provide the drawing accurate decision.

Consequently, the collection and analysis of data by obtaining multi-sensor sources can be applied to each system that is more accurate and in-depth than the obtained data results from a single sensor.

References

1. **Acosta, I. C. C., Khodadadzadeh, M., Tusa, L., Ghamisi, P., Gloaguen, R. (2019).** A machine learning framework for drill-core mineral mapping using hyperspectral and high-resolution mineralogical data fusion. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol. 12, No. 12, pp. 4829–4842. DOI: 10.1109/ jstars.2019.2924292.

2. **Angelov, P., Yager, R. (2013).** Density-based averaging – A new operator for data fusion. Information Sciences, Vol. 222, pp. 163–174. DOI: 10.1016/j.ins.2012.08.006.

3. **Bachmann, C., Abdulhai, B., Roorda, M. J., Moshiri, B. (2013).** A comparative assessment of multi-sensor data fusion techniques for freeway traffic speed estimation using microsimulation modeling. Transportation research part C: Emerging technologies, Vol. 26, pp. 33–48. DOI: 10.1016/j.trc.2012.07.003.

4. **Banerjee, T. P., Das, S. (2012).** Multi-sensor data fusion using support vector machine for motor fault detection. Information Sciences, Vol. 217, pp. 96–107. DOI: 10.1016/j.ins.2012. 06.016.

5. **Dehghan-Niri, E., Farhidzadeh, A., Salamone, S. (2012).** Adaptive multisensor data fusion for acoustic emission source localization in noisy environment. Structural Health Monitoring, Vol. 12, No. 1, pp. 59–77. DOI: 10.1177/1475921712462937.

6. **Doña, C., Chang, N. B., Caselles, V., Sánchez, J. M., Camacho, A., Delegido, J., Vannah, B. W. (2015).** Integrated satellite data fusion and mining for monitoring lake water quality status of the Albufera de Valencia in Spain. Journal of Environmental Management, Vol. 151, pp. 416–426. DOI: 10.1016/j.jenvman.2014.12.003.

7. **Dutta, R., Cohn, A. G., Muggleton, J. M. (2013).** 3D mapping of buried underworld infrastructure using dynamic bayesian network based multi-sensory image data fusion. Journal of Applied Geophysics, Vol. 92, pp. 8–19. DOI: 10.1016/j.jappgeo.2013.02.005.

8. **El-Din, D. M., Hassanein, A. E., Hassanien, E. E. (2020).** A proposed context-awareness taxonomy for multi-data fusion in smart environments: Types, properties, and challenges. DOI: 10.1007/978-3-030-47411-9_28.

9. **Forti, N., Gao, L., Battistelli, G., Chisci, L. (2022).** Unknown source in spatially distributed systems: Identifiability analysis and estimation. Automatica, Vol. 136, pp. 110025. DOI: 10.1016/j.automatica.2021.110025.

10. **Gan, Q., Harris, C. J. (2001).** Comparison of two measurement fusion methods for Kalman-filter-based multisensor data fusion. IEEE Transactions on Aerospace and Electronic Systems, Vol. 37, No. 1, pp. 273–279. DOI: 10.1109/7.913685.

11. **Hall, D. L., Llinas, J. (1997).** An introduction to multisensor data fusion. Proceedings of the IEEE, Vol. 85, No. 1, pp. 6–23. DOI: 10.1109/ 5.554205.

12. **Kior, A., Sukhov, V., Sukhova, E. (2021).** Application of reflectance indices for remote sensing of plants and revealing actions of stressors. Photonics, Vol. 8, No. 12, pp. 582. DOI: 10.3390/photonics8120582.

13. **Naderolasli, A. (2021).** Indirect self-tuning controller for a two degree of freedom tracker model. International Journal of Vehicle Autonomous Systems, Vol. 16, No. 1, pp. 15–37. DOI: 10.1504/IJVAS.2021.118028.

14. **Naderolasli, A., Ataei, M. (2016).** Identification of a two degree of freedom tracker system: Theoretical and experimental discussion. Majlesi Journal of Mechatronic Systems, Vol. 5, No. 2, pp. 1–6.

15. **Naderolasli, A., Ataei, M. (2020).** Stabilization of a two-DOF gimbal system using direct self-tuning regulator. International Journal on Electrical Engineering and

Informatics, Vol. 12, No. 1, pp. 33–44. DOI: 10.15676/ijeei.2020.12.1.3.

16. **Naderolasli, A., Chatraei, A. (2019).** One DOF robot manipulator control through type-2 fuzzy robust adaptive controller. Journal of Automation, Mobile Robotics and Intelligent Systems, Vol. 13, No. 1, pp. 65–70. DOI: 10. 14313/jamris_1-2019/7.

17. **Naderolasli, A., Tabatabaei, M. (2016).** Stabilization of the two-axis gimbal system based on an adaptive fractional-order sliding-mode controller. IETE Journal of Research, Vol. 63, No. 1, pp. 124–133. DOI: 10.1080/03772063.2016.1229581.

18. **Naderolasli, A., Tabatabaei, M. (2020).** Two-axis gimbal system stabilization using adaptive feedback linearization. Recent Advances in Electrical and Electronic Engineering, Vol. 13, No. 3, pp. 355–368. DOI: 10.2174/2352096512666181128095433.

19. **Prieto, J., Mazuelas, S., Bahillo, A., Fernandez, P., Lorenzo, R. M., Abril, E. J. (2012).** Adaptive data fusion for wireless localization in harsh environments. IEEE Transactions on Signal Processing, Vol. 60, No. 4, pp. 1585–1596. DOI: 10.1109/tsp.2012. 2183126.

20. **Qiu, S., Zhao, H., Jiang, N., Wang, Z., Liu, L., An, Y., Zhao, H., Miao, X., Liu, R., Fortino, G. (2022).** Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges. Information Fusion, Vol. 80, pp. 241–265. DOI: 10.1016/j.inffus. 2021.11.006.

21. **Ren, H., Rank, D., Merdes, M., Stallkamp, J., Kazanzides, P. (2012).** Multisensor data fusion in an integrated tracking system for endoscopic surgery. IEEE Transactions on Information Technology in Biomedicine, Vol. 16, No. 1, pp. 106–111. DOI: 10.1109/titb. 2011.2164088.

22. **Safari, S., Shabani, F., Simon, D. (2014).** Multirate multisensor data fusion for linear systems using Kalman filters and a neural network. Aerospace Science and Technology, Vol. 39, pp. 465–471. DOI: 10.1016/j.ast.2014. 06.005.

23. **Safizadeh, M. S., Latifi, S. K. (2014).** Using multi-sensor data fusion for vibration fault diagnosis of rolling element bearings by accelerometer and load cell. Information Fusion, Vol. 18, pp. 1–8. DOI: 10.1016/j.inffus. 2013.10.002.

24. **Sorber, L., Van-Barel, M., De Lathauwer, L. (2015).** Structured data fusion. IEEE Journal of Selected Topics in Signal Processing, Vol. 9, No. 4, pp. 586–600. DOI: 10.1109/jstsp.2015. 2400415.

25. **Wang, W., Chen, J., Hong, T. (2018).** Occupancy prediction through machine learning and data fusion of environmental sensing and Wi-Fi sensing in buildings. Automation in Construction, Vol. 94, pp. 233–243. DOI: 10.1016/j.autcon.2018. 07.007.

26. **Wu, R. T., Jahanshahi, M. R. (2018).** Data fusion approaches for structural health monitoring and system identification: Past, present, and future. Structural Health Monitoring, Vol. 19, No. 2, pp. 552–586. DOI: 10.1177/1475921718798769.

27. **Zitnik, M., Zupan, B. (2015).** Data fusion by matrix factorization. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 37, No. 1, pp. 41–53. DOI: 10.1109/tpami. 2014.2343973.

Article received on 08/07/2022; accepted on 09/11/2023.

** Corresponding author is Amir Naderolasli.*

ISSN 2007-9737

Thematic Section:

Advances in Pattern Recognition

This thematic section of Computación y Sistemas (CYS) contains a selection of seven papers presenting advances in the field of Pattern Recognition (PR). Pattern recognition is a branch of computer science focused on the study of algorithms and methodologies for identifying patterns within data. It encompasses the development of computational models and techniques that enable machines to recognize regularities, structures, or trends in diverse datasets. This field plays a crucial role in applications such as image and speech recognition, natural language processing, and machine learning, contributing to advancements in automation, decision-making, and artificial intelligence systems.

The guest editors meticulously curated the seven papers featured in this thematic section. Each manuscript underwent thorough evaluation by a minimum of three members of the scientific committee. Reviewers assessed various aspects such as originality, contribution to the field, soundness, and technical quality in determining the acceptance of a paper. Subsequent paragraphs offer an overview of the papers comprising this volume.

Velázquez-Arreola et al. evaluate the information heat maps provide and how they relate to the morphological characteristics of blood components in acute lymphoblastic leukemia. After considering four convolutional neural network (CNN) models to classify unsegmented images, they generated the respective heat maps with the LRP (Layer-wise Relevance Propagation), Deep Taylor, Input*Gradient, and Grad-Cam methods. They got the best results with the GoogleNet model and the Grad-Cam method, which were the ones that best related the natural morphological characteristics of the cell with the heat maps, reaching a good percentage of relevant pixels within at least one cellular morphological feature present, locating the most critical pixels within the nucleus.

Fortuna-Cervantes et al. present an analysis of the performance of three different CNNs with transfer learning for Art Media Classification (AMC) to answer the question of what challenges arise in this application. The authors introduced the Art Media Dataset (ArtMD) to train the CNNs. ArtMD contains five classes of art: Drawing, Engraving, Iconography, Painting, and Sculpture. Their results demonstrate that all the tested CNNs exhibit similar behavior. However, Drawing, Engraving, and Painting had the highest relationship, showing a strong relationship between Drawing and Engraving. Then, they removed Drawing and Engraving. By eliminating Drawing, they got the best performance. The experiments allowed them to conclude that Drawing and Painting have the lowest accuracy, showing a solid misclassification with the other classes. They also discussed the degree of relationship between the three CNN models and detailed AMC's challenges.

Molefe et al. introduce a multi-stage graph embedding approach for road-type classification tasks. The methodology involved deriving road segment feature vectors by extracting relevant edge and node attributes from the road network graph. Graph embedding techniques were employed to acquire the embedded vector representation of a designated road segment. The proposed method demonstrated superior performance to existing state-of-the-art methodologies addressing the Linkoping Road network dataset tasks. The authors compared the proposed method against alternative approaches that use raw features and different graph embeddings as input, in contrast to the deep autoencoder representation in this paper. The results show that using the DAE embedding method to obtain compact road segment features significantly improves the performance of graph embedding methods for road-type classification tasks. This work contributes to advancing intelligent road network systems, offering potential applications.

doi: 10.13053/CyS-28-1-4899

Luna-Lozoya et al. propose a lightweight CNN for detecting MicroCalcifications Clusters (MCCs) in digital mammograms using a reduced number of parameters (only 8301), which helps radiologists to accurately detect MCCs that plays an important role in early identification of breast cancer. This proposal constitutes a practical, efficient and effective solution for MCCs detection, requiring low computational resources. The proposed lightweight CNN achieves competitive results with more complex models like LeNet-5 and MobileNetV2; reaching an accuracy of up to 99.3%, but requiring much less parameters than these models. Currently a software application implementing the proposed model to detect MCCs in digital mammograms is being evaluated by expert radiologists. This research contributes to facilitate the practical use of Artificial Intelligence techniques for medical applications.

Torres-Rodríguez et al. conduct a comparative evaluation of various averaging methods for evoked potential estimation using realistic simulations. Simulated signals are crucial for assessing pattern recognition algorithms in the absence of gold standard records. The simulations are deemed realistic by introducing variations in potential latency, width, and amplitude, while background noise is simulated using an 8th order Burg autoregressive model derived from real auditory evoked potentials data. The simulations also incorporate actual instrumentation and acquisition channel effects, along with power line interference. Three averaging methods—consistent average, weighted average, and reported average—are compared in scenarios with and without artifacts. Results indicate that the trimmed average strikes the best balance between estimated signal-to-noise ratio (SNR) value and bias, particularly in the presence of artifacts.

Arevalo-Ancona et al. introduce a zero-watermarking scheme for medical image authentication using a Context Encoder neural network model. This approach enhances image reconstruction and robustness by extracting unique features, employing a halftone image of the patient's face as a watermark for identification. Extensive experiments demonstrate the method's resilience against various attacks, including geometric transformations and image processing manipulations, with low bit error rates and high normalized cross-correlation values confirming reliable watermark retrieval. The use of a large watermark size (160 x 160 pixels) facilitates easy patient identification, contributing to quick and accurate verification in medical applications. The deep neural network further improves the robustness, efficiency, and versatility of the proposed zero-watermarking scheme, making it a practical and effective solution for securing and authenticating medical images.

Garibaldi-Márquez et al. propose a two-stage deep learning approach for early weed detection in agriculture, addressing the challenge of uncontrolled conditions. Using a UNet-like architecture for image segmentation and ResNet101, VGG16, Xception, and MobileNetV2 for classification, the method achieves robust results, with a Dice Similarity Coefficient (DSC) of 87.48% and a mean Intersection over Union (mIoU) of 78.17% when images are divided into patches. Xception demonstrates the best classification performance with 97.43% accuracy. Despite confusion in segmenting and classifying specific plant classes, the approach proves advantageous for practical use in natural field conditions.

Guest editors:

Ansel Y. Rodríguez-González
Humberto Perez-Espinosa
José Francisco Martínez-Trinidad
Jesús Ariel Carrasco-Ochoa
José Arturo Olvera-López

ISSN 2007-9737

Secure Medical Image Authentication Using Zero-Watermarking based on Deep Learning Context Encoder

Rodrigo Eduardo Arevalo-Ancona, Manuel Cedillo-Hernandez*,
Ana Elena Ramirez-Rodriguez, Mariko Nakano-Miyatake,
Hector Perez-Meana

Instituto Politécnico Nacional, SEPI,
Mexico

[rarevaloa0900, aramirezr0906]@alumno.ipn.mx,
[mcedilloh, mnakano, hmperezm]@ipn.mx

Abstract. Zero-watermarking is a robust and lossless technique for digital image security, copyright protection, and content authentication. This paper introduces a novel zero-watermarking scheme for medical image authentication based on deep learning. The proposed approach leverages a neural network based on the Context Encoder to extract distinctive features from the image, enhancing the method. The training of the neural model increases robustness. The watermark consists of a halftone image of the patient's face, serving as a unique identifier for medical study. By revealing the watermark, medical professionals can verify the correspondence between the imaging study and the patient. Therefore, an XOR operation merges the watermark sequence and the extracted features. The proposed method offers continuous image protection, safeguarding sensitive medical data. Extensive experiments demonstrate the technique's robustness against various attacks, including geometric transformations (scaling, cropping, resizing, rotation) and image processing manipulations (filtering, blurring, JPEG compression, and noise addition). The detection watermark process achieves a low bit error rate and a high normalized cross-correlation, validating the method's robustness and effectiveness. The deep neural network improved the robustness of the presented zero-watermarking scheme making it suitable for practical applications in medical data security and integrity.

Keywords. Zero-watermarking, image security, image, authentication, deep learning, feature extraction.

1 Introduction

In recent years, the exponential growth in digital content use and distribution has increased with the development of digital technology. Digital images have facilitated services in the health area, such as telemedicine, e-learning, or remote assistance [1]. In addition, healthcare systems provide easy access to medical data, which could be manipulated or redistributed without authorization. Consequently, Medical imaging requires security, patient data privacy, and diagnostic accuracy [2], [3]. In this context, one of the principal challenges is the need for technology developments for image authentication, protection, and security.

Watermarking techniques are used for copyright protection, image authentication, and image protection [4]. Traditional watermarking techniques embed a signal into the image generating a distortion [5, 6]. Image distortion is not suitable for scenarios where the image integrity must be preserved. Arum Patel and Prabhat Patel proposed a novel hybrid watermark algorithm in their work, [7] which leverages wavelet coefficients for color components adapting.

They utilized the Singular Value Decomposition (SVD) technique on the LL and HH sub-bands obtained from the 2nd level of the Discrete Wavelet Transform (DWT) from the logo image. In another study by Sinhal and Ansari [8], a dual watermark scheme was designed to preserve the regions of interest in medical images.

First, a robust watermark is embedded into the image using the Integer Wavelet Transform (IWT). Simultaneously, to ensure the integrity of the regions of non-interest (RONI), a fragile watermark was incorporated by replacing the least significant bit (LSB).

doi: 10.13053/CyS-28-1-4898

During the detection stage, a deep neural network is deployed to extract the watermarks effectively. In their research [9], Qasim, Meziane, and Aspin proposed an innovative reversible and imperceptible watermarking scheme for detecting distortions on magnetic resonance images.

To minimize image distortion run-length encoding is used to compress the watermark. During the embedding process, the image undergoes segmentation for ROI detection, achieved through histogram thresholding. By identifying smooth blocks inside the ROI of the medical image that matches those in the watermark image.

Zero-watermarking has emerged as an innovative and effective technology for image authentication. Zero-watermarking schemes are a lossless technique for digital image security, copyright protection, and content authentication. Zero-watermarking hides information into a created stego-image or master share, merging features from the host image and the user's watermark sequence [10]. Zero-watermarking systems do not embed information into the image, keeping its quality intact without distortions.

One of the advantages of zero-watermarking lies in its non-intrusive nature. Unlike other methods that directly embed information into the image, zero-watermarking preserves the image's quality without introducing any distortions. This aspect is particularly critical in medical imaging, where maintaining the integrity of the original image is paramount for accurate diagnosis and analysis. Tayachi et al. [11] developed a hybrid watermark algorithm designed to protect DICOM images. The image is partitioned into two regions: ROI (Region of Interest) and RONI (Region of Non-Interest) using a thresholding technique.

In the zero-watermarking scheme, features from the ROI area are combined with the watermark. The non-zero-watermarking algorithm embeds multiple copies of the watermark in the RONI area. The embedding process employs a linear interpolation technique. Hosny and Darwish introduced an innovative zero-watermarking technique specifically designed for color medical images, as described in their research.

To enhance the security of the watermark, in [10] they applied the Arnold transform, effectively scrambling the watermark to prevent unauthorized access or tampering. The image features are obtained using the multi-channel fractional-order Gegenbauer moments (FrMGMs) of color images, which represents unique features of each image.

The master share construction combines the scrambled watermark with the extracted image features using the XOR logic operation. Rocek et al. introduced a reversible watermarking technique for medical imaging in their research [12]. They divided the image into ROI and RONI areas since the ROI areas cannot be distorted.

The zero-watermarking process employed the Dual-Tree Complex Wavelet Transform (DT-CWT) on the ROI. From the DT-CWT, they selected the LL coefficients to extract essential image features, which were combined with the watermark. To enhance the security, a reversible contrast mapping technique for RONI watermarking is applied.

Additionally, deep learning has revolutionized various fields, including computer vision, image processing, and image security. Deep learning techniques in zero-watermarking have shown promising results. Deep neural models help in the detection process of more robust and discriminative features.

Deep neural networks for watermarking systems enhance the security and resilience of the authentication process. Fierro-Radilla et al. generate a zero-watermarking using convolutional neural networks [13]. The authors presented an architecture centered around convolutional neural networks (CNN) comprising 13 layers and one fully connected layer for feature extraction.

This process allows the creation of a matrix containing the most relevant features extracted from the image. Subsequently, an XOR logic operation is applied to effectively merge the feature matrix with the watermark. Gong et al. introduced in their research a robust zero-watermarking for medical images based on the DenseNet model [14].

The DenseNet neural network is employed, and a dense block is added to obtain a residual block that facilitates the extraction of crucial feature maps related to the image. These feature maps were associated with the watermark. Furthermore, to enhance the security of the watermark, the authors implemented an encryption process using a logistic map algorithm.

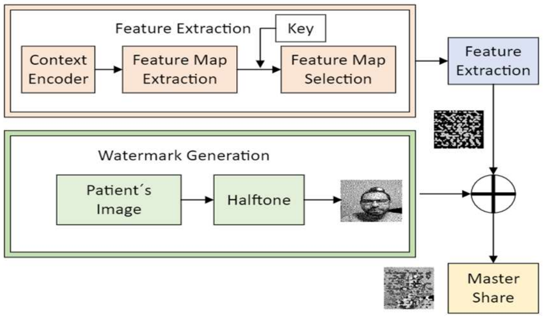


Fig. 1. Zero-watermarking encryption diagram

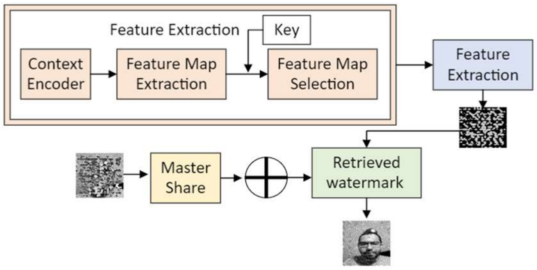


Fig. 2. Zero-watermarking diagram for watermark retrieval

Huang et al. presented a robust zero-watermarking system based on VGG network for healthcare information security [15]. A chaotic scrambling is applied for the encryption of the watermark, ensuring heightened security.

Next, the feature map is obtained using the VGG pre-trained model. Thus, it is combined with the image features to construct the hash. As a result, the hash becomes intricately linked with the scrambled watermark. Han et. al used the VGG19 neural network model to propose a zero-watermarking scheme for medical images [16].

The authors used the VGG19 model to extract deep feature maps directly related to the original medical image. These feature maps were fused to generate a comprehensive feature image. Therefore, it is applied the Discrete Fourier Transform to this image to obtain a feature matrix, which is combined with the watermark. To enhance the security of the watermark.

Several challenges associated with zero-watermarking methods can be summarized as follows: 1. Limited focus on specific image types: Many existing methods tend to concentrate on a particular type of image, which might limit their applicability to diverse image datasets and scenarios. 2. A narrow focus on specific attacks: Certain methods were designed to be robust against to specific attacks, which may leave the watermark vulnerable to other potential tampering. 3. Usage of small watermarks: Some proposed techniques employ small-sized watermarks, which might compromise the watermark's robustness and visibility in certain situations. 4. Overreliance on logo watermarks: The majority of utilized watermarks are often simple logo images, which might lack the capacity to embed complex information or provide sufficient security against sophisticated attacks.

The main contribution of this paper lies in the feature extraction using a Context Encoder, which significantly enhances the system's robustness. The Context Encoder is a deep neural model that learns specific features to facilitate watermark recovery, even in cases where the watermark has been tampered. The Context Encoder was designed for image reconstruction according to its context background.

Although the approach primarily focuses on medical images, the experiments also conduct tests with natural images, demonstrating excellent performance in image protection. This versatility allows the system to safeguard different types of images, beyond its initial scope. Furthermore, the watermark size is 160x160 pixels, constituting a halftone image derived from the patient's face.

This approach enables the treating doctor to determine if the study belongs to the patient. In the case of natural images or photographs, allow the author recognition to protect the image copyright. This additional level of detail and personalization enhances the overall security and authentication capabilities of the watermarking system.

In summary, the Context Encoder's application for feature extraction for watermarking methods, the adaptability to different image types, and the utilization of specific watermarks contribute to the paper's significant contributions in advancing image protection and authentication techniques.

The rest of the paper is organized as follows: In Section 2, the proposed method is comprehensively explained, detailing the innovative approach used for zero-watermarking. Section 3 presents the experimental results obtained from the conducted tests, showcasing the performance and effectiveness of the proposed

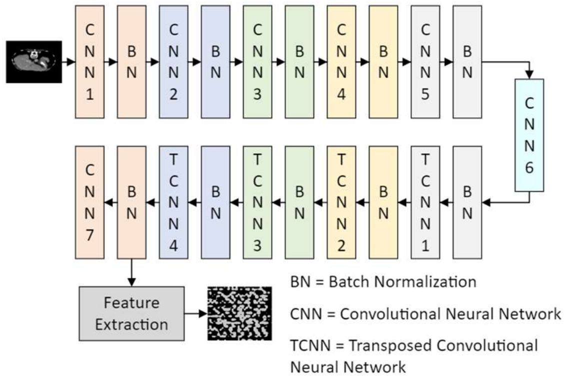


Fig. 3. Neural network architecture

Table 1. Neural network hyperparameters

Layer	Neurons	Stride	Padding	Kernel Size
CNN 1	64	2	1	4
CNN 2	128	2	1	4
CNN 3	512	2	1	4
CNN 4	256	2	1	4
CNN 5	512	2	1	4
CNN 6	1024	2	1	4
TCNN 1	512	2	1	4
TCNN 2	256	2	1	4
TCNN 3	128	2	1	4
TCNN 4 (Feature Map Extraction)	64	2	1	4
CNN 7	3	2	1	4

method. Finally, in Section 4, the conclusions drawn from this research are presented, summarizing the key findings, and highlighting the significance of the proposed approach in the context of image protection and watermarking.

2 Zero-Watermarking Proposed Method

This paper introduces a robust zero-watermarking scheme specifically designed for medical images. However, it shows robustness in natural images. The feature extraction stage is based on the Context Encoder, a generative deep learning model used for inpainting. Moreover, the watermark is a halftone image derived from the patient's face.

On the other hand, the watermark employed in this scheme is larger in size (160x160 pixels) compared to most of the watermarks used in previous research. The Context Encoder learns unique features from the image. This neural network model increases the robustness of the zero-watermarking system against different tampering attacks. The general diagram of the zero-watermarking technique is depicted in Fig. 1 and Fig. 2, providing a visual representation of the algorithm.

The algorithm is detailed in the following subsections. Context Encode model: This section explains the architecture of the Context Encoder for feature extraction. Watermark Generation: Describes the process of generating the watermark, ensuring it is suitable for embedding within medical and natural images. Master share generation: Illustrates the encryption method used to enhance the security of the watermark, ensuring its protection against geometric and advanced image processing attacks. Watermark Retrieval: Explains the procedure for retrieving the watermark, allowing verification and authentication purposes. The overview of the proposed zero-watermarking technique, this paper contributes to image protection and authentication in medical and natural image domains.

2.1 Context Encoder Model

Deep learning is a branch of machine learning that utilizes data to learn features through pattern recognition, enabling more informed decision-making. Deep learning algorithms aim to perform tasks more efficiently, resulting in better learning of extracted characteristics from the data [17].

One area where deep learning has made significant strides is image feature extraction. It enables automatic learning and discrimination of features from the data. In deep neural networks, lower layers obtain simple features like edges and textures, while deeper layers extract more complex and abstract features, such as high-level patterns.

This hierarchical feature extraction process allows the neural network to learn representations and structures from the image data, leading to a better understanding and recognition of visual patterns. The implementation of deep learning models for feature extraction has notably improved the performance of various computer vision tasks [18, 19] surpassing traditional approaches.

The Context Encoder [20], is specifically designed for inpainting techniques. The model focuses on approximating the likelihood of pixels to create new data samples using a generator. A discriminator is employed to test the similarity of the generated data to the original dataset through a probability distribution.

The generator (*G(z)*) utilizes features from the dataset to create new data, while the discriminator (*D(z)*) distinguishes the differences between the generated samples and the original dataset (1) [18]:

$$\min_G \max_D V(D,G) = E_{x\sim P(data)}[LogD(x)] + E_{x\sim P(z)}[Log(1-D(G(z)))], \tag{1}$$

where the loss function of the discriminator log(D(z)) learns the features from the dataset using the probability from the similarity between the dataset image and the reconstructed ($x\sim P(data)$) image to compare the loss function of the generator log(1-D(G(z))) and approximates the equilibrium between the features of the dataset and the new samples ($x\sim P(z)$).

This paper focus on the Context Encoder, due to its ability to learn unique features. These features are specific to each image, contributing to the robustness of the watermarking process.

The training process is conducted using different geometric and image advanced processing attacks on the medical images, ensuring the model can identify only the most crucial features relevant to watermarking.

The incorporation of such a refined feature selection approach facilitates the creation of a secure master share, ensuring the protection of sensitive patient information embedded in the watermark sequence. Fig. 3 provides a clear illustration of the neural network model's architecture.

Overall, our emphasis on the Context Encoder and the strategic training of the neural network result in an effective and robust watermarking technique, ensuring authentication and integrity in medical image applications.

The neural network employs the rectified linear activation function (ReLU) on each layer specifically for region of interest detection, effectively identifying essential features crucial for constructing the master share. The feature map is extracted from the TCNN 4 layer (Transposed Convolutional Neural Network), subsequently, the selected feature map is binarized.

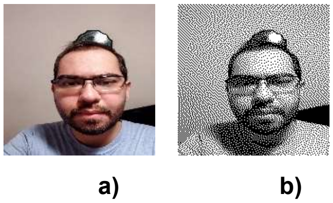


Fig. 4. a) Original patient's image, b) Halftone version of patient´s image

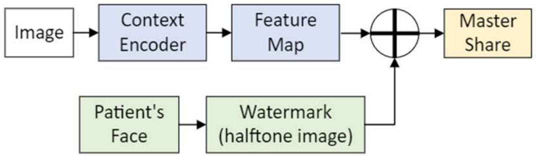


Fig. 5. Master share construction

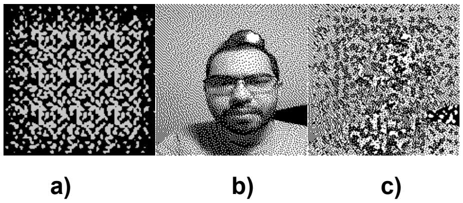


Fig. 6. a) Image features, b) Halftone image watermark, c) Master share

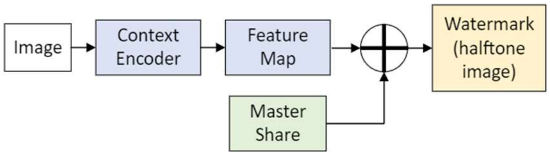


Fig. 7. Watermark retrieval

The Mean Squared Error (MSE) defined by (2) is used as the loss function. The MSE computes the average error and quantifies the distance between the image feature values (x) and the extracted features $(\overline{x})$. A lower error value indicates that the predictions are closer to the actual feature values [21]:

$$MSE = \frac{1}{n}\sum(x-\bar{x})^2. \tag{2}$$

Table 1 summarizes the configuration of the neural network's layers, specifying the number of neurons, kernel size, and stride.

A ReLU activation function is applied for each layer and MSE loss, the neural network effectively identifies and captures the significant image features, allowing for accurate and reliable construction of the master share. Finally, the feature maps are obtained from the TCNN. The selection of the specific feature map is accomplished using a key, which allows us to determine and extract the desired features from the network.

This selective feature map extraction process ensures that only relevant and critical features are used for the master share construction, enhancing the efficiency of the zero-watermarking algorithm. With the key-based selection, we can focus on the most important features needed for our zero-watermarking system, enabling better performance.

2.2 Watermark Generation

The watermark utilized in this technique is created from an image of the patient's face, which is transformed into a halftone image. Halftoning is a widely used image processing technique that converts continuous-tone grayscale images into binary halftone representations.

The main objective of halftoning is to replicate the appearance of various shades of gray by strategically distributing black and white pixels, thereby simulating the illusion of grayscale tones through carefully placed dots of varying sizes. This process ensures that the watermark effectively captures the visual information of the original image while being suitable for embedding and authentication purposes [22]. The halftone image (*Ih*) transform is calculated for each pixel from the gray image (*Ig*) using (3):

$$Ih(i,j) = Ig(i,j) + \sum h(m,n)e(i-m,j-n)\,, \tag{3}$$

where $h(m,n)$ in (4) is the error filter and $e(i-m,j-n)$ in (5) is the quantization error:

$$h(m,n) = \begin{bmatrix} 0 & Ig(i,j) & 7 \\ 3 & 5 & 1 \end{bmatrix}, \tag{4}$$

$$e = u(i,j) - Q, \tag{5}$$

where *Q* is the binarize pixel value (6):

$$Q = \begin{cases} 0 \quad if \quad Ig(i,j) < threshold \\ \quad 1 \quad if \quad other \end{cases}. \tag{6}$$

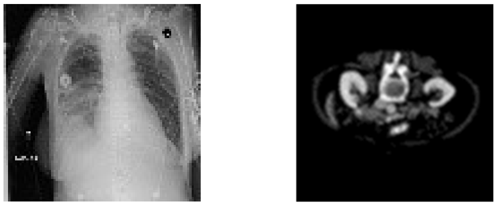
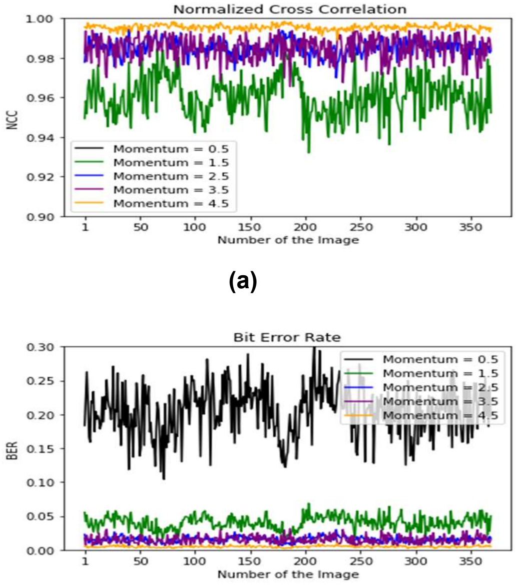


Fig. 8. Test medical images



(a)

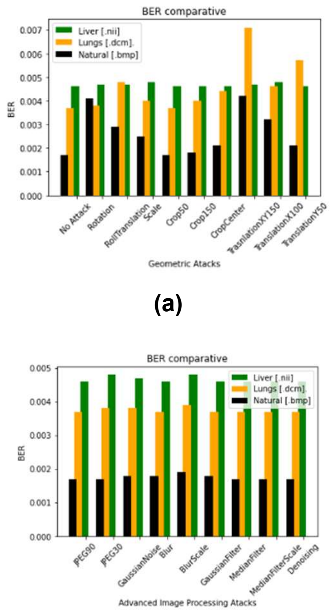
(b)

Fig. 9. a) Momentum BER, b) Momentum NCC

Fig. 4a displays the original image of the patient, while Fig. 4b illustrates the same image after undergoing the conversion process to a halftone representation. The halftone technique ensures the preservation of key visual features of the patient's image. In addition, it is suitable as watermark for the master share construction authentication purposes.

2.3 Master Share Generation

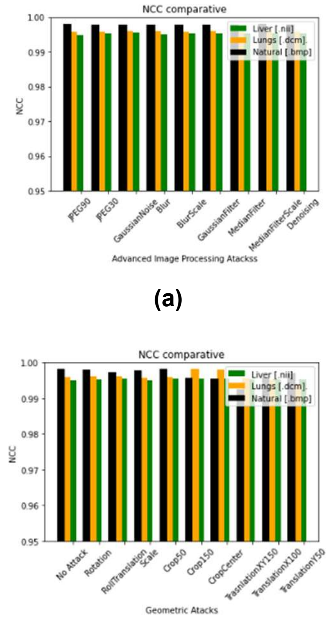
The master share plays a crucial role in image authentication and certification, which is store securely in an external device [23]. The master share (*MS*) consists of a stego-image that incorporates information from the host image.



(a)

(b)

Fig. 10. a) Geometric attacks, b) Common signal processing



(a)

(b)

Fig. 11. a) Geometric attacks, b) Common signal processing

This information is derived from the combination of extracted features (*ef*) through the Context Encoder, and the binarized watermark (*W*) by an XOR (⊕) logic operation given by (7):

$$MS = ef \oplus W. \tag{7}$$

This process can be visualized in the diagram presented in Fig. 5. In Fig. 6, we present a graphical representation of the extracted features from the image, as well as the watermark and master share generation process.

2.4 Watermark Retrieval

The watermark retrieval (*W'*) validates the patient's identity and certifies the ownership of the imaging study. The watermark is revealed by (8), which involves combining the master share (*MS*) with the extracted features (*ef*) obtained from the pre-trained Context Encoder.

This process allows us to securely reveal, verify the watermark, and confirm the ownership of the medical data:

$$W' = ef \oplus MS. \tag{8}$$

This process is in the diagram from the Fig. 7.

This process verifies the integrity and authenticity of the imaging study, providing a robust watermarking system for medical image applications.

3 Experiments Results

The experiments were conducted to evaluate the robustness and efficiency of the proposed algorithm. The main testing image database [24] is comprised with 123 liver images, each with a size of 512 x 512 in Neuroimaging Informatics Technology Initiative (NII) format.

However, to increase the diversity of the dataset, the image base was expanded to 369 images, capturing various perspectives of the same liver images.

Moreover, the method's robustness was tested using 369 images with 512 x 512 size in Digital Imaging and Communications in Medicine (DICOM) format in modality x-ray obtained from [25]. An Example of test medical images are shown in Fig. 8. Additionally, the algorithm was validated with 369 natural images (.bmp) from [26]. This extensive testing ensured the algorithm's versatility and suitability for various image types. The watermark consists of a halftone image derived from the patient's face, sized at 160 x 160 pixels.

This personalized approach reinforces the security and uniqueness of the watermark, making it highly effective for image authentication and ownership certification purposes. Additionally, data augmentation techniques were applied to enhance the robustness of the zero-watermarking scheme.

Advanced image processing attacks, such as blurring, median filtering, Gaussian filtering, denoising, and JPEG compression, were employed to simulate various image degradations and assess the algorithm's robustness under different conditions.

Furthermore, geometric attacks, including rotation, scaling, translation, and cropping, were performed to gauge the effectiveness of the zero-watermarking technique against spatial transformations and cropping scenarios.

The Bit Error Rate (BER) (9) serves as an evaluation metric to measure the detected bit errors between the original watermark (W) and the retrieved watermark (W'). A low BER value indicates a stronger level of watermark robustness [27]. In other words, a lower BER signifies that the retrieved watermark closely matches the original watermark, indicating a more reliable and accurate watermark retrieval process:

$$BER = \frac{Number\ of\ error\ bits}{Total\ bits}. \qquad (9)$$

In addition, the Normalized Cross-Correlation (NCC) (10) is used to assess the similarity between the original watermark (*W*) and the retrieved watermark (*W'*) [28].The NCC measures how closely the extracted watermark matches the original watermark, providing a quantitative evaluation of their similarity:

$$NCC = \frac{\sum(W(i,j)*W'(i,j))}{\sqrt{\sum(W(i,j)^2*\sum W'(i,j)^2)}}. \qquad (10)$$

Fig. 9 showcases the Bit Error Rate (BER) and Normalized Cross-Correlation (NCC) results with different momentums on the batch normalization layers. After evaluating the neural network's performance, a momentum value of 4.5 was selected.

Table 2. Retrieved watermark advanced processing attacks.

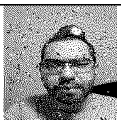
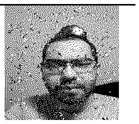
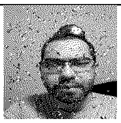
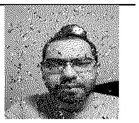
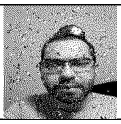
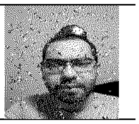
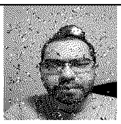
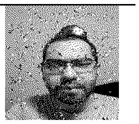
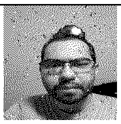
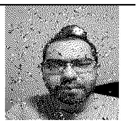
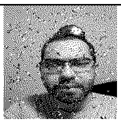
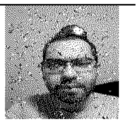
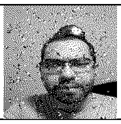
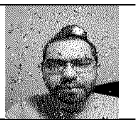
Attack	Retrieved Watermark	Attack	Retrieved Watermark
JPEG Lossy Compression Quality Factor 90		Gaussian Filtering 3x3	
JPEG Lossy Compression Quality Factor 30		Median Filter 3x3	
Gaussian noise μ=0, σ^2=0.9		Blurring with re-scaling	

Table 3. Retrieved watermark geometric attacks

Attack	Retrieved Watermark	Attack	Retrieved Watermark
Rotation 0°-360°		Cropping by 150x150	
Circular shifting by *x*= 150		Centered cropping	
Down sampling 64 x 64		Translation x=100, y=100	
Translation x=100, y=0		Translation x=0, y=100	

This choice was based on its ability to yield a stable system with minimal variations in the BER and NCC values. Furthermore, this momentum value produced an average BER of 0.0046 and an average NCC of 0.9950, indicating highly accurate and reliable watermark retrieval.

The selection of momentum as 4.5 demonstrates its effectiveness in optimizing the network's performance, ensuring consistent and precise watermark recovery. The stable system achieved through this momentum value further increases the algorithm's robustness.

Table 4. Comparison applying geometric attacks

Parameter	[10]	[11]	[14]	[15]	Proposed
Detection metrics	BER, NC	BER, NC	NC	PSNR, MSE, NC	BER, NCC
Watermark size	32x32	16x16	32x32	64x64	160x160
Geometric attacks					
Cropping	Yes	Yes	Yes	Yes	Yes
Rotation	Yes	No	Yes	Yes	Yes
Scaling	Yes	Yes	Yes	Yes	Yes
Translation	No	Yes	Yes	Yes	Yes
Signal processing distortions					
JPEG	Yes	No	Yes	Yes	Yes
Gaussian noise	Yes	Yes	Yes	Yes	Yes
Median filter	Yes	Yes	No	Yes	Yes
Gaussian Filter	Yes	Yes	Yes	No	Yes
Blurring	No	No	No	No	Yes
Feature extraction method	Fractional Order-Legendre Fourier Moments	Statistical feature (skewness, entropy and median)	Neural networks	Neural networks	Neural networks
Medical images	No	Yes	Yes	Yes	Yes
Natural images	Yes	No	No	No	Yes

Fig. 10 and Fig. 11 provide a visualization of the obtained (BER) and (NCC) values from the testing using databases of medical imaging and natural images, respectively.

The BER values reflect the accuracy of watermark retrieval process for both medical and natural images. Fig. 11 displays the NCC values, demonstrating the similarity between the original watermark and the retrieved watermark.

The results demonstrate the algorithm's versatility in effectively handling different image formats, including NII, DICOM, and BMP respectively. The graphics exhibit the algorithm's remarkable robustness, reflected in the low error value and similarity between the original watermark and the retrieved watermark.

The Table 2 and Table 3 clearly demonstrate the efficient recovery of the watermark, regardless the image is tampered or distorted. This finding highlights the effectiveness and robustness of the proposed system for authenticating medical images. The watermark retrieval process obtained few errors, indicating that the system can accurately reconstruct the watermark even if the image has been distorted.

The efficiency of the watermark retrieval confirms the reliability and practicality of the proposed method for ensuring the authenticity and integrity of medical images. These results underscore the algorithm's ability to perform consistently and accurately across diverse types of images. Table 4 provides a comprehensive comparison of the robustness of the proposed algorithm against other methods.

The comparison with Hosny's [10] method focuses on a zero-watermarking algorithm for medical images using ROI characteristics. Additionally, the comparisons with Huang [15], Tayachi [11], and Gong [14] methods are emphasized since these schemes concentrate on obtaining characteristics from a pretrained neural models for master share generation.

The proposed scheme outperforms these methods in terms of robustness, even when subjected to various attacks. This demonstrates the algorithm's superior ability to withstand image alterations and maintain watermark integrity. The main advantage of the proposed system lies in the rigorous analysis of its performance through the modification of hyperparameters in the neural model.

This approach allows the system to focus on learning specific image characteristics, enabling better watermark recovery. The main disadvantage of this method lies in the training stage of the neural network for medical images.

The proposed scheme effectively generates protection for a set of medical images, it does not provide specific individualized protection for each image, since many of the features from the image can be obtain in other imaging analysis, this applies for most of the zero-watermarking schemes for medical imaging.

However, for natural images, the method may indeed offer individualized protection for every single image. The training process could potentially focus on capturing specific features and patterns unique to individual natural images, enabling more personalized image protection and authentication.

4 Conclusions

In this paper, a robust Zero-watermarking algorithm based on the Context Encoder neural network model is proposed. The algorithm obtains image features to enhance image reconstruction and increase the robustness of the zero-watermarking system.

We conducted extensive experiments encompassing medical imaging and natural images, including Neuroimaging Informatics Technology Initiative (NII) and Digital Imaging and Communications in Medicine (DICOM) formats for medical images and .bmp for natural images.

The results demonstrate a high level of robustness in both image types, with low Bit Error Rate (BER) values indicating reliable watermark retrieval. Additionally, the high Normalized Cross-Correlation (NCC) values signify a strong similarity between the original and retrieved watermarks, validating the algorithm's effectiveness in preserving watermark integrity during the retrieval process. Overall, the proposed algorithm exhibits robustness, efficiency, and versatility, making it a reliable solution for image authentication and verification.

With the neural model fine-tuning and the extraction of specific image features, the algorithm increased its efficiency and improved the watermark retrieval process, highlighting its potential for robust image protection and authentication. The results presented here validate the effectiveness of the proposed algorithm and its suitability for medical applications.

Finally, one of the key contributions of this research is the watermark with a size of 160 x 160 pixels, which surpasses the typical size employed in other research papers. The watermark size is particularly valuable as it enables medical professionals to easily identify the patient and authenticate the medical imaging.

The increased size provides a clear representation of the patient's face, facilitating quick and accurate verification by the doctor.

Acknowledgments

The authors thank the Instituto Politécnico Nacional (IPN), as well as the Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) for the support provided during the realization of this research.

References

1. **Neymeen, H., Boles, W., Boyd, C. (2013).** A review of medical image watermarking requirements for technology. Journal Digit Imaging, Vol. 26, No. 2, pp. 326–343.
2. **Coatrieux, G., Quantin, C., Montagner, J., Fassa, M., Allaert, F. A., Roux, C. (2008).** Watermarking medical images with

anonymous patient identification to verify authenticity. MIE, Vol. 136, pp. 667–672.

3. **Cedillo-Hernandez, M., Cedillo-Hernandez, A., Nakano-Miyatakea, M., Perez-Meana, H. (2020).** Improving the management of medical imaging by using robust andsecure dual watermarking. Biomedical Signal Processing and Control, Vol. 56. DOI: 10.1016/j.bspc.2019.101695.
4. **Garcia-Nonoal, Z., Mata-Mendoza, D., Cedillo-Hernandez, M., Nakano-Miyatake, M. (2023).** Secure management of retinal imaging based on deep learning, zero-watermarking and reversible data hiding. The Visual Computer, Vol. 40, pp. 1–16. DOI: 10.1007/s00371-023-02778-1.
5. **Sinhal, R., Ansri, I. A., Ahn, C. W. (2020).** Blind image watermarking for localization and restoration of color images. IEEE Access, Vol. 8, pp. 200157–200169. DOI: 10.1109/ACCESS.2020.3035428.
6. **Wang, H., Yuan, Z., Chen, S., Su, Q. (2023).** Embedding color watermark image to color host image based on 2D-DCT. Optik, Vol. 274. DOI: 10.1016/j.ijleo.2023.170585.
7. **Patel, A. K., Patel, P. (2022).** Color adaptive robust DWT-SVD watermarking algorithm and limitations: Color space comparisons. 2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT), pp. 656–661.
8. **Sinhal, R., Ansari, I. A. (2023**). Machine learning based multipurpose medical image watermarking. Neural Computing and Applications, pp. 1–22. DOI: 10.1007/s00521-023-08457-5.
9. **Qasim, A. F., Meziane, F., Aspin, R. (2018).** A reversible and imperceptible watermarking scheme for MR images authentication. IEEE, 24th International Conference on Automation and Computing, pp. 1–6. DOI: 10.23919/IConAC.2018.8749000.
10. **Hosny, K. M., Darwish, M. M., Fouda, M. M. (2021).** New color image zero-watermarking using orthogonal multi-channel fractional-order legendre-fourier moments. IEEE Access, Vol. 9, pp. 91209–91219. DOI: 10.1109/ACCESS.2021.3091614.
11. **Tayachi, M., Nana, L., Pascu, A. C., Benzarti, F. (2023).** A hybrid watermarking approach for DICOM images security. Applied Sciences, Vol. 13, No. 10, pp. 6132. DOI: 10.3390/app13106132.
12. **Roček, A., Javorník, M., Slavíček, K., Dostál, O. (2017).** Reversible watermarking in medical imaging with zero distortion in ROI. 2017 24th IEEE International Conference on Electronics, Circuits and Systems, pp. 356–359. DOI: 10.1109/ICECS.2017.8292071.
13. **Fierro-Radilla, A., Nakano-Miyatake, M., Cedillo-Hernandez, M., Cleofas-Sanchez, L., Perez-Meana, H. (2019).** A robust image zero-watermarking using convolutional neural networks. 2019 7th International Workshop on Biometrics and Forensics (IWBF) IEEE. pp. 1–5. DOI: 10.1109/IWBF.2019.8739245.
14. **Gong, C., Liu, J., Gong, M., Li, J., Bhatti, U. A., Jixin, M. (2022).** Robust medical zero-watermarking algorithm based on residual-densenet. IET Biometrics, Vol. 11, No. 6, pp. 547–556. DOI: 10.1049/bme2.12100.
15. **Huang, T., Xu, J., Tu, S., Han, B. (2023).** Robust zero-watermarking scheme based on dephtwise overparameterized VGG network in healthcare information security. Biomedical Signal Processing and Control, Vol. 81.
16. **Han, B., Du, J., Jia, Y., Zhu, H. (2021).** Zero-watermarking algorithm for medical image based on VGG19 deep convolution neural network. Journal of Healthcare Engineering, pp. 104478. DOI: 10.1155/2021/5551520.
17. **Ganguly, K. (2017).** Learning generative adversarial networks: next generation deep learning simplified. Birmingham, United Kingdom, Packt Publishers Ltd.
18. **Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Ward-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014).** Generative adversarial networks. Computer and Information Sciences.
19. **Arjovsky, M., Chintala, S., Bottou, L. (2017).** Wasserstein generative adversarial networks. Proceedings of the 34th International Conference on Machine Learning, PMLR, Vol. 70, pp. 214–223.

20. **Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A. A. (2016).** Context encoders: Feature learning by inpainting. Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544.
21. **Dohi, N., Rathnayake, N., Hoshino, Y. (2022).** A comparative study for COVID-19 cases forecasting with loss function as AIC and MSE in RNN family and ARIMA. 2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS&ISIS), IEEE, pp. 1–5. DOI: 10.1109/SCISISIS55246.2022.10001870.
22. **Lo, S. Y., Patel, V. M. (2021).** Error diffusion halftoning against adversarial examples. arXiv. pp. 3892–3896. DOI: 10.1109/ICIP42928.2021.9506591.
23. **Arevalo-Ancona, R. E., Cedillo-Hernandez, M. (2023).** Zero-watermarking for medical images based on regions of interest detection using K-means clustering and discrete fourier transform. International Journal of Advanced Computer Science and Applications, Vol. 14, No. 6. DOI: 10.14569/IJACSA.2023. 0140662.
24. **Medical Segmentation Decathlon**. http://medicaldecathlon.com/.
25. **SIIM-ACR Pneumothorax Segmentation**. https://www.kaggle.com/competitions/siim-acr-pneumothorax-segmentation/data.
26. **FREE STOCK IMAGES**. https://www.stockvault.net.
27. **Dai, Z., Lian, C., He, Z., Jiang, H., Wang, Y. (2022).** A novel hybrid reversible-zero watermarking scheme to protect medical images. IEEE Access, Vol. 10, pp. 58005–58016. DOI: 10.1109/ACCESS.2022.3170030.
28. **Morales-Ortega, A., Cedillo-Hernandez, M. (2022).** Ownership authentication and tamper detection in digital images via zero-watermarking. 2022 45th International Conference on Telecommunications and Signal Processing IEEE, (TSP), pp. 122–125. DOI: 10.1109/TSP55681.2022.9851253.
29. **Oueslati, S., Cherif, A., Solaimane, B. (2011).** Adaptive image watermarking scheme based on neural network. International Journal of Engineering Science and Technology, Vol. 3, No. 1, pp. 748–757.
30. **Hosny, K., Darwish, M. (2021).** New geometrically invariant multiple zero-watermarking algorithm for color medical images. Biomedical Signal Processing and Control, Vol. 70. DOI: 10.1016/j.bspc.2021.103007.

Article received on 04/07/2023; accepted on 13/10/2023.
**Corresponding author is Manuel Cedillo-Hernandez.*

ISSN 2007-9737

Evaluation of Heat Map Methods Using Cell Morphology for Classifying Acute Lymphoblastic Leukemia Cells

José de J. Velázquez-Arreola, Nohemí Sánchez-Medel,
Oliver A. Zarraga-Vargas, Raquel Díaz-Hernández*,
Leopoldo Altamirano-Robles

National Institute of Astrophysics, Optics and Electronics, Puebla, Mexico

raqueld@inaoep.mx

Abstract. Explainable artificial intelligence (XAI) is a field of research that has attracted the interest of researchers in recent years. These algorithms seek to provide transparency to artificial intelligence (AI) models. One application of these algorithms is in the medical area, created as an auxiliary tool for corroborating predictions obtained by an AI when classifying pathologies, for example, Acute Lymphoblastic Leukemia (ALL). The present work evaluates the amount of information heat maps provide and how they relate to the blood components' morphological characteristics. For the assessment, four Convolutional Neural Network (CNN) models were retrained and fine-tuned to classify unsegmented images (ALL_IDB2 database). Subsequently, their respective heat maps were generated with the LRP (Layer-wise Relevance Propagation), Deep Taylor, Input*Gradient, and Grad-Cam methods. The best results were obtained with the GoogleNet model and the Grad-Cam heat map generation method, having a percentage of 43.61% of relevant pixels within at least one cell morphological feature present. Moreover, the most significant pixels are within the nucleus, with 73.97% of important pixels inside. According to the results, the Grad-Cam method best relates the relevant pixels generated in the heat map to the morphology of the cell of interest to classify a healthy or diseased cell.

Keywords. Explainable artificial intelligence (XAI), heatmaps, acute leukemia lymphoblastic (ALL), grad-cam, cell morphology.

1 Introduction

Deep learning applications, surpassing human capabilities in tasks like image and speech recognition and recommendation systems, have received substantial attention.

Despite their achievements, these applications cope with a critical shortfall in both explainability and reliability.

Deep learning models are commonly perceived as complex black boxes, presenting challenges in understanding their intricate underlying mechanisms. Their inability to justify decisions and predictions undermines human trust, heightening concerns, especially considering the potential life-threatening errors that artificial intelligence algorithms may make depending on the application.

For example, a flaw in the computer vision system of an autonomous car could result in a catastrophic crash, while in healthcare, where decisions directly impact human lives, the stakes are considerably high.

In response to these challenges, many methods have emerged to address the need for transparency and reliability in deep learning applications. Notably, explainable Artificial Intelligence has emerged as a focal point in machine learning research.

These methods aim to explain machine and deep learning models in a manner easily understandable by humans. The categorization of interpretability methods is based on how they provide explanation information, encompassing visual, textual, and mathematical or numerical approaches. This paper evaluates different visual interpretability methods for classifying Acute Lymphoblastic Leukemia cells.

doi: 10.13053/CyS-28-1-4897

1.1 The Problem

In the medical field, there has been an increase in the activity of digitizing pathological studies for medical diagnosis. The digitalization opens the door to life-saving artificial intelligence (AI) applications. One of the branches of application of these explanatory methods is to use them as an auxiliary tool in validating the predictions made by a neural network.

However, to diagnose some diseases through digital microscopic images, as in the case of Acute Lymphoblastic Leukemia (ALL), it is necessary to pay attention to the morphological characteristics present in the cells of interest.

From this arises the need to evaluate whether heat maps, as a visual explanatory method, are appropriate to highlight the morphological characteristics present. Thus, the expert can consider them as an aid for the corroboration of the classification, giving rise to the diagnosis of the disease.

1.2 Cell Morphology of Acute Lymphoblastic Leukemia

White blood cells are essential to the human body's immune system. They have a specific morphology depending on the type of blood component. Leukemia is an alteration in the production and malformation of these cells.

The French-American-British (FAB) classifies Leukemia into Acute and Chronic Lymphoblastic Leukemia. It also categorizes them into ALL-L1: small uniform cells; ALL-L2: large, varied cells; and ALL-L3: large, mixed cells with vacuoles (bubble like features).

The nuclear and cytoplasmic structures can differentiate between healthy and diseased cells. Acute Myeloid Leukemia (AML) and Chronic Myeloid Leukemia (CML) are also caused by abnormal myelocytes.

The authors in [1, 2, 3], argue that benign and malignant cells can be discriminated by their nuclear structure, nucleus-to-cytoplasm ratio, color, and texture. Fig. 1 illustrates an example of the difference between the structure of healthy and cancerous cells.

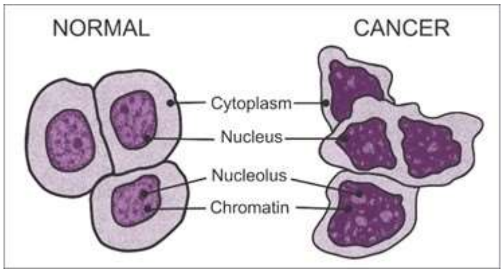


Fig. 1. Example of the difference between the structure of healthy and cancerous cells

It shows that cancer cells have an irregular structure, and the shape of the nucleus is collapsed; this is how hematologists describe the ALL.

This work evaluates whether the heat maps produced by selected methods are related to at least one of these morphological features. This work evaluates whether the heat maps produced by selected methods are related to at least one of these morphological features.

1.3 Explanatory Methods

In this work, we use methods that generate heat maps; these are considered a way to explain the functioning of a neural network. In [4] define three concepts that are usually misused and interchanged. As a result of their research, they conclude the following definitions:

- Interpretability is the ability to explain or provide meaning in terms understandable to a human being.
- Explainability is associated with the notion of explanation as an interface between humans and decision-makers. At the same time, it accurately represents the decision-maker and is understandable to humans.
- Transparency: A model is considered transparent if it is understandable. Since a model can have different degrees of comprehensibility, transparent models are divided into three categories: simulatable, decomposable, and algorithmically transparent.

Explainability is critical for the safety, approval, and acceptance of AI systems for clinical use. At work [5] is a comprehensive overview of techniques that apply XAI to improve various properties of ML models and systematically classifies these approaches, comparing their respective strengths and weaknesses.

In recent years, different heat map generation algorithms have been proposed to understand neural networks better. These methods include Deep Taylor, Input*Gradient, and LRP, among others. However, comparing results between these methods is somewhat complex because it is necessary to replicate each of these methods separately.

The author in [6] made available the iNNvestigate library; this tool solves the problem of method comparison, providing a standard interface and implementing several published methods for heat map generation, facilitating the analysis of neural network predictions by generating heat maps. This work uses this library to create heat maps of the strategies implemented therein, specifically the Deep Taylor, Input*Gradient, and LRP methods.

Grad-Cam [7], uses the gradient of the classification score related to the convolutional features determined by the network to understand which parts of the image are most important for classification. For this work, the algorithm implemented in MATLAB software is used.

1.4 Retrained Models

Four different models of CNNs were used in this work:

- VGG-19 is a convolutional neural network with 19 depth layers [8].
- ResNet18 is a neural network with 18 depth layers [9].
- ResNet50 is a neural network with 50 depth layers [9].
- GoogleNet is a neural network with 22 depth layers [10].

These neural networks were pre-trained with more than one million images from the ImageNet database. The pre-trained network can classify images into 1000 object categories (e.g., keyboard, mouse, pencil, and many animals). As a result, the network has learned feature-rich representations for a wide range of objects. The size of the network's image input is 224 by 224 pixels.

The idea behind selecting these architectures was to experiment with small and large models. In addition to belonging to the best-known models, these models usually perform better when transfer learning is done using other datasets. Therefore, these models were retrained with the database images described in section 3.1.

2 Related Work

Table 1 compares recent works that present different techniques to solve the problem. All authors focus on classifying images containing ALL cell types, using techniques to relate them to the morphology of the cell. Most authors using CNN models highlight the ResNet50, VGG's, and Inceptions models. On the other hand, few authors use a method of visual explanation.

The authors in [11], evaluate different algorithms to calculate heat maps using a hematologist specialized in ALL diagnosis. Generated heat maps were assessed with the help of five hematologists and experts in morphological cell classification. The evaluation focused on the amount of information provided by the heat maps and how they relate to morphological characteristics present in the classified cells.

Results of the best heatmaps and hematologist evaluations are presented in this work. The central outcome is that the heatmaps must include morphological information to be a valuable tool for medical diagnosis systems.

Following the same line of research expressed in [11], the present work represents an extension in the sense that a reference map with the morphology of each cell in the analyzed images was produced, which allows quantification of what percentages of the pixels marked as significant by the heat map, fall into morphologically coherent entities. This way, evaluating which algorithm is more relevant concerning cell morphology is possible.

In [22] it is proposed a method of classification and explanation. The proposed method contemplates segmentation of the morphological characteristics of the cells. Subsequently, the method uses a ResNet50 network that performs the classification, obtains the respective heat map, and generates an explanation of spatial features.

Table 1. Comparative table of related works

Author	Type Blood Cell	Model	XAI Method
N. Jiwani et al. in [12]	ALL	No	No
Jiang et al. [13]	ALL	Wavelet	No
Abir et al. [14]	ALL	Resnet50, DenseNet121 and VGG16	LIME
Nayeon Kim [15]	ALL Pro-B	InceptionV3, Res-Net101V2, InceptionResNetV2, and VGG19	LIME and DCN
Ochoa-Montiel et al. [16]	ALL	Random Forest, LeNet, AlexNet	No
Maaliw R.R. et al. [17]	ALL	Transfer learning InceptionV3, Xception InceptionResNetV2	No
Velázquez-Arreola et al. [19]	ALL	VGG16, VGG19, ResNet50, and MobileNet V1	LRP, Deep Taylor, and Input*Gradient methods
Diaz R. J. et al. [22]	ALL	Modified ResNet-50	Grad CAM

Both steps are considered visual explanatory methods. In addition, they perform heat map generation experiments, with the cell segmented and unsegmented. They conclude that better results are obtained when the heat map is generated with the segmented cell.

The main difference between the work presented here and the work discussed above is that we evaluate the number of most relevant pixels according to the segmentation categories. Experiments are also performed by combining different CNN models and heat mapping methods.

The experiments aim to identify which combination is the most efficient in relating relevant pixels against morphological features using the unsegmented image. Consequently, our work differs substantially from the work explained above.

3 Methodology

This section describes the database used for retraining and evaluation. It also presents the semi-manual segmentation procedure to create the reference map. This map will be used to compare the heat maps and thus evaluate the position of the most relevant pixels. Finally, the general methodology of this article is described.

3.1 Image Database

The Acute Lymphoblastic Leukemia Imaging Database for Image Processing (IDB-ALL) [17] is publicly accessible, and the categories are balanced. It features microscopic images of blood samples. It is a database intended to evaluate and compare algorithms for image segmentation and classification in Acute Lymphoblastic Leukemia (ALL).

Each image in the database was identified and classified by a group of oncologists with expertise in identifying ALL diseased cells. The photos are divided into diseased and healthy, with 130 images for each category. The images have a resolution of 257 x 257 pixels in RGB.

This work divided the images into 100 images for each category (healthy and diseased). With the remaining subset of images, 30 per category, a section of never-before-seen images was created. The latter will be used exclusively to evaluate the models after the entire retraining process and generate heat maps.

As described in [16], the typical datasets used for leukemia cell recognition have drawbacks related to category imbalance problems. In other cases, they were constructed from different sources or acquisition conditions. Leukemia cells

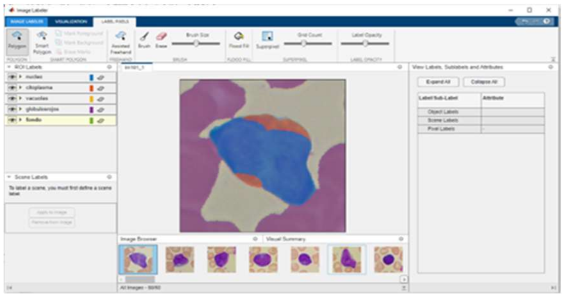


Fig. 2. Segmentation of an image from the IDB-ALL2 database using Image Labeler software

contain two or more categories of Leukemia, including its subtypes (Lymphocytic Leukemia and Myeloid Leukemia, Chronic or Acute). For our purpose, this work is focused on the ALL type.

The ALL-IDB2 dataset [17] is small. However, it is one of the most widely used datasets. For example, in [3, 16, 18], is publicly accessible, and the categories are balanced. For this reason, an ALL-IDB2 data set was selected for this work.

3.2 Data Segmentation (Semi-Manual)

As mentioned in section 1.2, cell morphology is used to identify ALL diseased cells. The main characteristics focus on the nucleus and cytoplasm. For this reason, a semi-manual segmentation is performed, highlighting five classes: nucleus, cytoplasm, vacuoles, red blood cells, and background.

The segmentation results will be considered the base reference (ground truth), which will later be used to evaluate the heat maps. The evaluation compares the most relevant pixels in each heat map and the ground truth corresponding to the original image.

The segmentation was performed using the MATLAB Image Labeler [19]. This application allows labeling reference images from a collection of pictures, defining rectangular region of interest (ROI) labels with aligned or rotated axes, line ROI labels, pixel ROI labels, polygon ROI labels, point ROI labels, projected cuboid ROI labels, and scene labels.

Fig. 2 illustrates how image segmentation from the IDB-ALL2 database was performed with the Image Labeler application. The segmentation process was carried out by two doctoral students who worked on this research and supervised by two hematologists with experience in cellular morphology from the Mexican public health system. The image was segmented into five categories: nucleus, cytoplasm, vacuole, red blood cells, and background. These categories were chosen at the suggestion of the hematologists.

3.3 General Methodological Process

The general methodology is composed of the following steps:

1. Obtaining the image database (ALL-IDB2). Then, separate the images into two folders: 200 images for training and 60 images that will be used as never-seen-before. Photographs of healthy and diseased cells are included.
2. Segmentation of each image using the Image Labeler application [19]. A matrix of the same size as the segmented image will be generated. These matrices will be the ground truth reference used for the evaluation.
3. Using the 200 images for retraining, we applied data augmentation by the traditional method (rotation and reflection) [20] to have 1000 images at the end for each type of cell (Healthy or ALL). Using these images, we finally retrained the neural networks GoogleNet, ResNet18, ResNet50, and VGG19.
4. Generate the heat maps using the retrained models. Heat maps are generated with the Deep Taylor, Input*Gradient, and LRP methods using the iNNvestigate library. Grad-Cam type heat maps are generated with Matlab software.
5. Evaluation of the heat maps. For this process, only the most essential pixels for the neural network during classification are considered. Due to the color map used, it is evident that the pixels with the highest relevance are located in the red channel of the heat map images. Therefore, this channel is used to evaluate the heat maps. The evaluation process is described as follows:
    a. We used the pixels of the red channel. Calculate the mean color depth of all pixels in this channel. The obtained value will be the reference value to

narrow down the pixels with the highest relevance.

b. Identify the pixels above the reference value, then locate each pixel in the image of heat maps and the ground truth matrix. Then, count all pixels for each class according to the segmentation.

c. Save the number of pixels in each category in a table for later analysis.

6. We produce some graphs of the results and analyze them. The results and their analysis are described in the following section.

4 Experiments and Results

This section describes the experiments and their results, such as images generated, results tables, and graphs to analyze them.

4.1 Heatmaps Generated with iNNvestigate

As mentioned at the end of section 2, this work is an extension of [11]. That work explains the reasons why the iNNvestigate library is used and why only the Deep Taylor, Input*Gradient, and LRP methods were used. The same procedures are analyzed in the present work by continuing that research. Fig. 3 shows some heat maps obtained for a cell classified as healthy and a cell classified as unhealthy.

The heat map generated with the Deep Taylor method shows over the entire image different degrees of relevance, using a single color to show the significance of the pixels for the neural network. However, most of the time, it offers higher levels of relevance at the edges of the cells of interest.

The heat maps obtained by the Input*Gradient method generate pixels according to the color scale ranging from blue to red, with blue being the pixels with the lowest relevance and red pixels with the highest relevance.

With this method, identifying the shapes or regions of greater importance to the neural network in classification is a little more complex. The complexity arises because it generates relevant and non-relevant pixels very close to each other and with poorly defined regions compared to other methods.

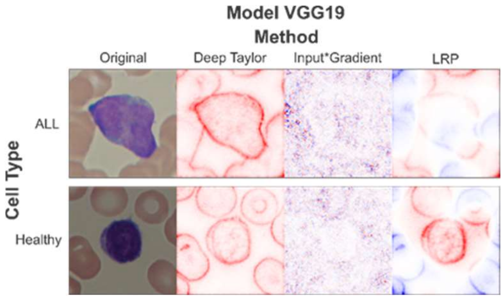


Fig. 3. Examples of heat maps obtained with the iNNvestigate library

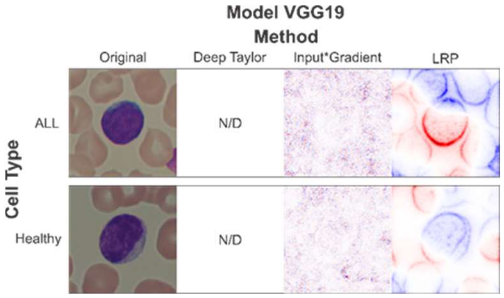


Fig. 4. Examples of heat maps where the Deep Taylor method is not defined

Finally, with the LRP method, the heat maps obtained, like the previous method, handle the color scale of blue and red. Unlike the Input*Gradient method, these heat maps show more clearly the regions of greater relevance to the neural network than those unimportant.

Fig. 4 shows two cells where the heat map is not defined for the Deep Taylor method. These cases, according to [6], are inherent to the technique since a value that works as a threshold is calculated. If this threshold is not exceeded, the heat map is not generated.

4.2 Grad-Cam Heat Maps

The authors in [7] propose a method of generating heat maps to provide "visual explanations" for decisions of a large class of convolutional neural network (CNN) based models, making them more transparent.

Their approach, Gradient Weighted Class Activation Mapping (Grad-CAM), uses the gradients of any target concept, flowing into the final convolutional layer to produce an approximate

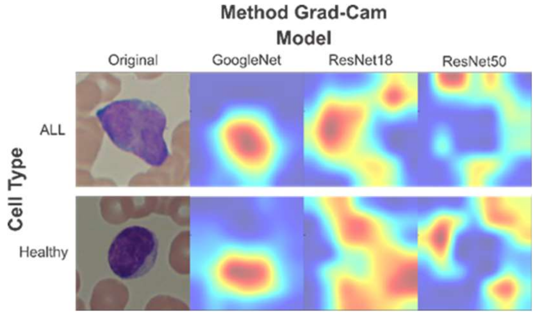


Fig. 5. The Figure shows results obtained with the Grad-Cam method and the GoogleNet, ResNet18, and ResNet50 models

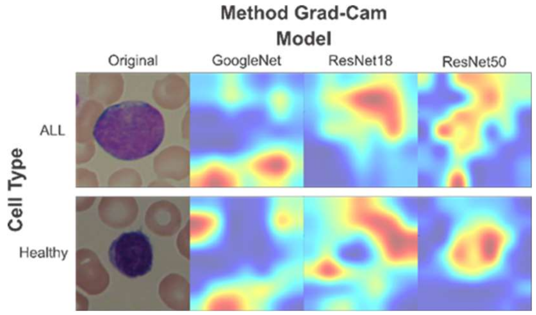


Fig. 6. The Figure shows results obtained with the Grad-Cam method and the GoogleNet, ResNet18, and ResNet50 models

Table 2. Table of heat map evaluation results. Retrieved from [19]

General Evaluation of the three models and cells types			
Method	**Average**	**% Morphological Information**	**Num. Heat maps with the highest score**
LRP	1.19	24%	4
Deep Taylor	1.69	34%	8
Input*Gradient	**2.30**	**46%**	**25**

location map that highlights the most essential pixels for the CNN network in class prediction.

Grad-CAM applies to a wide variety of CNN model families. These include CNNs with fully connected layers (e.g., VGG), CNNs used for structured outputs (e.g., subtitles), CNNs used in tasks with multimodal inputs (e.g., visual response to questions), or reinforcement learning without architectural changes or retraining.

In the context of image classification models, heat maps generated with this method provide insight into the failure modes of these models (showing that seemingly unreasonable predictions have reasonable explanations), outperforming the approaches described in the previous section.

Finally, the authors designed and conducted human studies to measure whether Grad-CAM explanations help users establish adequate confidence in deep network predictions and showed that Grad-CAM enables untrained users to successfully discern a "stronger" deep network from a "weaker" one, even when both make identical predictions.

Figure 5 visualizes some heat maps obtained with the Grad-Cam method for the GoogleNet, ResNet18, and ResNet50 models, for a diseased and healthy cell image. This figure shows that the most relevant pixels with the GoogleNet model are mainly located within the same location as the cell of interest.

In the heat maps obtained with the ResNet18 model, some regions of interest are positioned within the concerned cell. However, most of these pixels are located outside the target cell.

Finally, in the heat maps that correspond to the ResNet50 model, the regions of interest usually are not related to the cell represented in the image. At first glance, it could be said that the Grad-Cam method, in combination with the GoogleNet model, is the best combination to generate heat maps that relate to the morphology of the target cell.

However, in Figure 6, we can see the results obtained with cells different from the previous figure. In this image, it is visualized that the heat map obtained is not always the best.

4.3 Evaluation by Experts

The authors in [11] present the results of evaluating the heat maps generated with the Deep Taylor, Input*Gradient, and LRP methods. The evaluation was performed by five pathologists with expertise in ALL cell classification.

From this work, it was obtained that the Input*Gradient method was the one that best visually related to the morphology of the target cell. The results are shown in Table 2. However, a database of segmented cells from the background was used in that work.

The paper presents feedback from one of the experts. It indicates that heat maps have low

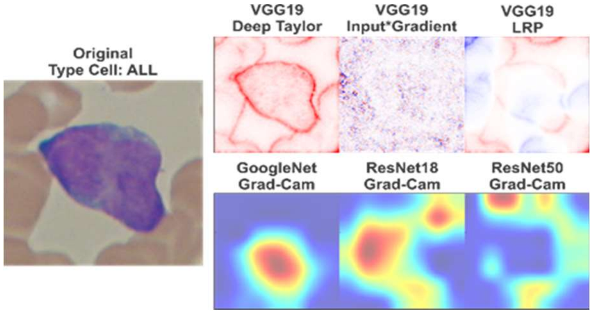


Fig. 7. Comparative image of the results obtained with the models and heat map generation methods evaluated

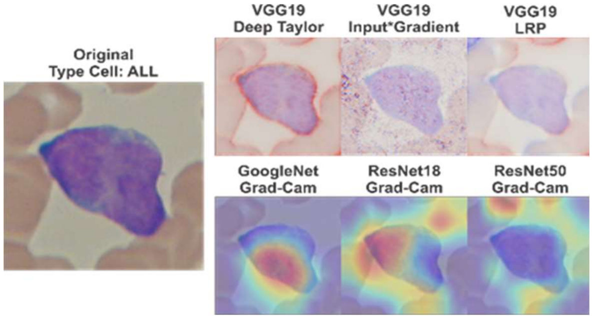


Fig. 8. Comparative image of the heat maps obtained superimposed on the ALL cell with which they were generated

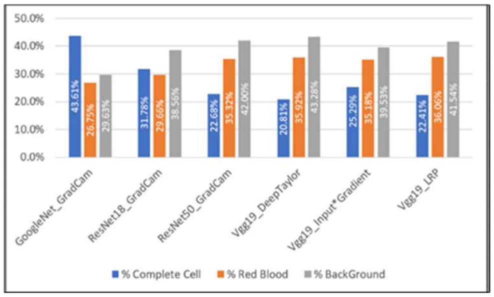


Fig. 9. Comparison between model and heat map generation method and their relationship with the central cell, red blood cells, and background, regarding percentages of relevant pixels

correlated information with the morphology of interest.

The present work is derived from the results presented in that paper. Here, we perform a computational evaluation of the heat maps using a semi-manual segmentation of the most significant morphological features, adding to the assessment of the Grad-Cam method.

We decided to use unsegmented images, i.e., the photos contain the cell of interest, the background, and other blood elements.

4.4 Heatmap Evaluation

Section 3.3 describes the methodological process used to evaluate the heat maps. The results obtained are shown here. In Fig. 7 shows the heat maps generated by the approaches and Fig. 8 shows an overlay of the heat map and the diseased cell with which they were produced as comparative images of the heat maps generated with the models and methods described above are displayed for an ALL-diseased cell.

If we know the number of relevant pixels placed within each segmented region, each category's percentage of pixels can be calculated based on a total of appropriate pixels.

The results of the evaluation performed in this work are shown in the graph in Fig. 9. From these results, it can be seen that the GoogleNet model and the Grad-Cam method is the combination of model and approach that best relates to the morphological characteristics of the cell of interest since 43.61% of the pixels marked as significant are located on the cell, 26.75% of the pixels are positioned in the red blood cells and the remaining 29.63% in the background.

In the second place, the ResNet18 model and the Grad-Cam method were set with 31.78% of the relevant pixels inside the interest cell, 29.66% in the red blood cells, and 38.56% in the background. The combination that obtained the worst result was the VGG19 model with the Deep Taylor method, with percentages of relevant pixels within the target cell of 20.81%, 35.92% in red blood cells, and the highest rate in the background at 43.28%.

We calculate the mean number of relevant pixels for each category. It is observed that the combination of the GoogleNet model and the GradCam architecture has a mean of 8071 pixels within the cell. In contrast, the red blood cell and background categories have the values of 4951 and 5484, respectively.

It is the only combination where the difference is more significant to a considerable extent between the complete cell versus the red blood cells and background categories. The results are shown in the graph of Fig. 10. The evaluation

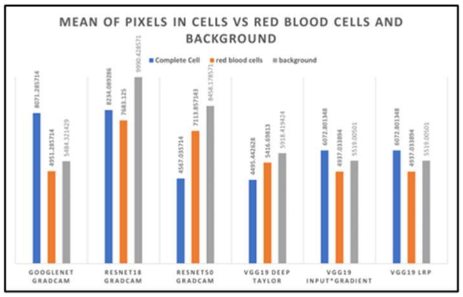


Fig. 10 Mean number of relevant pixels for each category using a combination of heatmap generation versus model

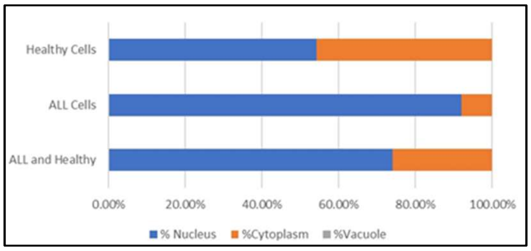


Fig. 11. Percentage of pixels with higher relevance located within the cell morphology of interest. Heat maps were created with the Google Net model and the Grad-Cam method

results in percentage terms of the best model-method combination are shown in Fig. 11.

The results show that the most relevant pixels for healthy cells are 54.26% within the nucleus and 45.74% in the cytoplasm. In the case of ALL cells, 91.90% are located within the nucleus, and 8.10% are located in the cytoplasm. The latter is related to the morphological characteristics that describe the disease because, in ALL cells, the cell's nucleus tends to have a larger area than the nucleus of a healthy cell.

In the database with which the heat maps were evaluated, there are three images where vacuoles were visualized. For this combination of the CNN model and heat map generation method, no pixels are of great relevance in their locations. Therefore, the vacuole category has 0% relevance.

5 Discussion

Performing a computational evaluation of the generated heat maps based on a map of morphological features (nucleus, cytoplasm, vacuoles) for the classification of ALL cells, as well as red blood cells and background, allows to evaluate whether the CNN focuses on cell features of interest or other elements present in the image. In addition, it will enable the comparison of heat map generation methods to define which correlates with such morphological features.

According to the results obtained in this research, the GoogleNet model and the Grad-Cam method are the ones that best relate the natural morphological characteristics of the cell with the heat maps.

According to the results obtained in the present work, the evaluation made by expert pathologists in [11] can be corroborated.

6 Conclusion

Implementing heat maps in a neural network aims to identify the most critical regions or pixels for the neural network classification process. In this work, we generated heat maps with four different methods (Deep Taylor, Input*Gradient, LRP, and Grad-Cam) implemented on four different architectures (GoogleNet, ResNet18, ResNet50, VGG19).

The ALL-IDB2 database containing unsegmented images of the cell of interest with background and other blood elements present was used.

A ground truth map was generated and divided into three morphological features (nucleus, cytoplasm, and vacuoles), red blood cells, and background. Using the reference map, we evaluated the generated heat maps.

This evaluation concludes that the GoogleNet model focuses primarily on features present in the cell of interest. The Grad-Cam method is the heat map generation method that best expresses the relevance of CNNs. Combined with the GoogleNet model, it yields results that focus exclusively on the target cell.

7 Future Work

The generation of heat maps as a tool to explain the result of a prediction in an image is promising.

However, research is still required because the results of the heat maps should focus on showing the outcomes that hematologists expect. Most importantly, the construction of heat maps must include morphological features to be useful for medical specialists, so we will continue to explore the line of generating visual explanatory methods that focus exclusively on morphological features present in the cell of interest.

Acknowledgments

José de Jesús Velázquez-Arreola acknowledges CONAHCyT for the scholarship granted towards pursuing his PhD studies, CVU 778129.

References

1. **Hariprasath, S., Dharani, T., Shaikh-Mohammad, B. N. (2019).** Automated detection of acute lymphocytic leukemia using blast cell morphological features. 2nd International Conference on Advances in Science & Technology (ICAST).
2. **Laosai, J., Chamnongthai, K. (2018).** Classification of acute leukemia using medical-knowledge-based morphology and CD marker. Biomedical Signal Processing and Control, Vol. 44, pp. 127–137. DOI: https://doi.org/ 10.1016/ j.bspc.2018.01.020.
3. **Lamberti, W. F. (2022).** Classification of white blood cell leukemia with low number of interpretable and explainable features. DOI: 10.48550/arXiv.2201.11864.
4. **Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Herrera, F. (2020).** Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information fusion, Vol. 58, pp. 82–115. DOI: 10.1016/j.inffus.2019.12.012.
5. **Weber, L., Lapuschkin, S., Binder, A., Samek, W. (2023).** Beyond explaining: opportunities and challenges of XAI-based model improvement. Information Fusion, Vol. 92, pp. 154–176, DOI: 10.1016/j.inffus. 2022.11.013.
6. **Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K. T., Montavon, G., Kindermans, P. J. (2019).** iNNvestigate neural networks! Journal of Machine Learning Research, Vol. 20, pp. 1–8.
7. **Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D. (2017).** Grad-cam: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE international conference on computer vision, pp. 618–626.
8. **Simonyan, K. (2015).** Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations (ICLR 2015), Computational and Biological Learning Society, pp. 1–14.
9. **He, K., Zhang, X., Ren, S., Sun, J. (2016).** Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.
10. **Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A. (2015).** Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-9.
11. **Velázquez-Arreola, J., Zarraga-Vargas, O. A., Díaz-Hernández, R., Altamirano-Robles, L. (2023).** Evaluation of heatmaps as an explicative method for classifying acute lymphoblastic leukemia cells. Proceedings 15th Mexican Conference, MCPR 2023, pp. 252–260. DOI: 10.1007/978-3-031-33783-3_24.
12. **Jiwani, N., Gupta, K., Pau, G., Alibakhshikenari, M. (2023).** Pattern recognition of acute lymphoblastic leukemia (ALL) using computational deep learning. IEEE Access, Vol. 11, pp. 29541–29553. DOI: 10.1109/ACCESS.2023.3260065.
13. **Jiang, Z., Dong, Z., Wang, L., Jiang, W. (2021).** Method for diagnosis of acute lymphoblastic leukemia based on ViT-CNN ensemble model. Computational Intelligence and Neuroscience, Vol. 2021. DOI: 10.1155/ 2021/7529893.
14. **Abir, W. H., Uddin, M. F., Khanam, F. R., Tazin, T., Khan, M. M., Masud, M., Aljahdali,**

S. (2022). Explainable AI in diagnosing and anticipating leukemia using transfer learning method. Computational Intelligence and Neuroscience, Vol. 2022. DOI: 10.1155/2022 /5140148.

15. **Nayeon, Kim. (2021).** Deep learning technology-based model to identify benign and Pro-B acute lymphoblastic leukemia (ALL): Xception + LIME. American Journal of Biomedical and Life Sciences, Vol. 9, No. 5, pp. 279–285.
16. **Ochoa-Montiel, R., Olague, G., Sossa, H. (2020).** Expert knowledge for the recognition of leukemic cells. Applied Optics, Vol. 59, No. 14, pp. 4448–4460. DOI: 10.1364/AO.385208.
17. **Labati, R. D., Piuri, V., Scotti, F. (2011).** All-IDB: The acute lymphoblastic leukemia image database for image processing. 2011 18th IEEE International Conference on Image Processing, IEEE, pp. 2045–2048. DOI: 10.11 09/ICIP.2011.6115881.
18. **Maaliw, R. R., Alon, A. S., Lagman, A. C., Garcia, M. B., Susa, J. A. B., Reyes, R. C., Hernandez, A. A. (2022).** A multistage transfer learning approach for acute lymphoblastic leukemia classification. 2022 IEEE 13th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) pp. 0488–0495. DOI: 10.1109/ UEMCON54665.2022.9965679.
19. **Mathworks.** Matlab Images Labeler application. https://mathworks.com/help/ vision/ref/imagelabeler-app.html.
20. **Mathworks.** Matlab augmented Images Datastore. https://la.mathworks.com/help/deeplearning/ref/augmentedimagedatastore.html.
21. **Mamalakis, A., Barnes, E. A., Ebert-Uphoff, I. (2022).** Investigating the fidelity of explainable artificial intelligence methods for applications of convolutional neural networks in geoscience. Artificial Intelligence for the Earth Systems, Vol. 1, No. 4. DOI: 10.1175/ AIES-D-22-0012.1.
22. **Diaz-Resendiz, J. L., Ponomaryov, V., Reyes-Reyes, R., Sadovnychiy, S. (2023).** Explainable CAD system for classification of acute lymphoblastic leukemia based on a robust white blood cell segmentation. Cancers, Vol. 15, No. 13, pp. 3376. DOI: 10.3390/ cancers15133376.

Article received on 04/07/2023; accepted on 13/10/2023.
[*]Corresponding author is Raquel Díaz-Hernández.

ISSN 2007-9737

Benchmarking of Averaging Methods Using Realistic Simulation of Evoked Potentials

Idileisy Torres-Rodríguez[*,1], Roberto Díaz-Amador[2], Beatriz Peón-Pérez[3],
Alberto Hurtado-Armas[1] , Alberto Taboada-Crispi[1]

[1] Universidad Central “Marta Abreu” de Las Villas,
Informatics Research Center, Santa Clara,
Cuba

[2] Universidad Católica del Maule,
Departamento de Medicina Traslacional,
Facultad de Medicina, UCM, Talca,
Chile

[3] Hospital Manuel Piti Fajardo,
Departamento de Electromedicina, Santa Clara,
Cuba

{Itrodriguez, ataboada]@uclv.edu.cu, rodiaz@ucm.cl

Abstract. The objective of this research is to conduct a comparative evaluation of various averaging methods for estimating evoked potentials using realistic simulations. Simulated signals are commonly employed to assess pattern recognition algorithms for event-related potential estimation since obtaining gold standard records is challenging. The simulations used are considered realistic because they allow for variations in potential latency, component width, and amplitudes. Background noise is simulated using an 8th order Burg autoregressive model derived from the analysis of a real dataset of auditory evoked potentials. The simulations incorporate actual instrumentation and acquisition channel effects, as well as power line interference. Three averaging methods for estimating the evoked potential waveform are compared: classical consistent average, weighted average, and reported average. The comparisons are conducted in two scenarios: one without artifacts and the other with 20% contamination by artifacts. The results of the comparative evaluation indicate that the trimmed average offers the best trade-off between the estimated signal-to-noise ratio (SNR) value and bias.

Keywords. Evoked Potentials, averaging methods, realistic simulation, benchmarking, SNR, bias.

1 Introduction

The utilization of simulated signals allows for training, evaluating, or comparing different digital signal processing techniques or pattern recognition algorithms, providing researchers with an unlimited number of test signals for experimentation [1]. While monitoring brain activity through electroencephalographic recordings is widely practiced, assessing the methods for analyzing these signals poses a challenge due to the absence of a reliable gold standard for comparison.

However, to assess various algorithms proposed for signal analysis and pattern detection, researchers often resort to using simulated signals instead of real signals, which typically conform to oversimplified models that do not accurately represent reality. In [2], a system is introduced for generating simulations of evoked potential recordings that exhibit a high level of realism.

This simulation takes into consideration potential variations in latency, width, and amplitude, which are common in real-world scenarios. Event-related potentials in actual

doi: 10.13053/CyS-28-1-4894

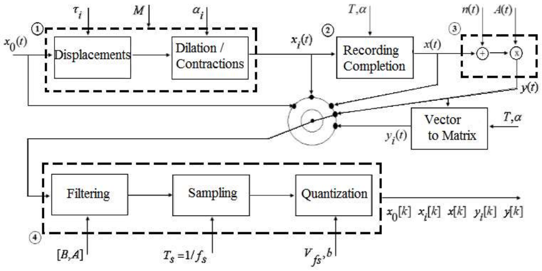


Fig. 1. General scheme for event-related potentials in wide sense simulation

Table 1. Characteristics of the designed Butterworth high-pass filter

Filter's design characteristics	Value
f_c in the passband	30 Hz
f_c in the stopband	15 Hz
Attenuation in the passband	1 dB
Attenuation in the stopband	-6 dB
Order	2

contexts can be contaminated by additive and multiplicative noise, as well as affected by recording instrument effects such as analog filtering, sampling, and quantization. Unfortunately, these aspects are often overlooked in most current evaluations.

All these parameters were estimated from real signals described in [3]. Using realistic simulations of evoked potentials and their associated noise and interference, different methods of robust estimation of the evoked response waveform will be evaluated.

2 Methods

2.1 Selection of the Parameters for Realistic Simulation

The simulation scheme (Fig. 1) used was previously proposed in [3] to simulate event-related potentials in a wide sense. The selection of the parameters for the realistic simulations of evoked potentials was carried out in [2].

In block 1 of Fig.1, $x_o(t)$ represents the initial fundamental epoch, which is defined for all t ∈ $[0, a)$, and serves as a reference for generating $(M-1)$ additional epochs. The goal is for the initial waveform to closely resemble the waveform of the potential under study.

In this specific case, the clean recordings of Auditory Evoked Potentials from healthy individuals, obtained from the database published in [4] and described in [3], are chosen as the basic waveforms for simulation. These signals specifically correspond to auditory brainstem responses (ABR) and are characterized by their short-latency potentials.

The parameter τ_i , which accounts for the variations in relative displacements due to latency in $x_o(t)$, is simulated using a normal distribution with a mean of zero and a standard deviation on the order of 0.2 ms. This value is derived from the average standard deviations of the component V latencies, as indicated in the study conducted in [5].

Similarly, the parameter α_i, representing the variations in the width of $x_o(t)$, is simulated using a normal distribution with a mean of zero and a standard deviation on the order of 0.07 ms, based on the values reported in [5]. The simulation allows for selecting different values of M, depending on the desired size of the resulting matrix set.

The selection of the event period T is determined based on the stimulation period, which in this instance comprised 2002 samples (equivalent to 41.7 ms), representing the time interval between each applied stimulus. In this particular case, the width of the analysis window, denoted as a, was set to 884 samples (equivalent to 18.4 ms).

These parameter values can be adjusted to accommodate the specific choice of the initial basic epoch and the potential being simulated. Regarding the additive noise component, denoted as $n(t)$, it is generated as a sum of various sources.

In this case, it consists of the estimated background noise, the 60 Hz interference, and its harmonics, as well as the alpha rhythm commonly present in many signals from the selected database. To process this noise, a low-pass filter is applied using the coefficients estimated by an 8th-order Burg model. Subsequently, a high-pass filter is employed, following the specifications detailed in Table 1, as outlined in the approach presented in [2].

To simulate the alpha rhythm, which appears in certain signals and must be considered when analyzing signal non-homogeneities that can impact the estimation of the average signal, a white noise signal is employed. This white noise signal is subjected to bandpass filtering using a second-order Butterworth approximation with cutoff frequencies ranging from 9 to 11 Hz, as detailed in [6].

The amplitude of the alpha rhythm is randomly distributed between 30 μV and 50 μV, following a normal distribution. This distribution aligns with the analysis carried out on the dataset, ensuring consistency with the characteristics of the actual signals. The parameters associated with filtering, sampling, and quantization are derived from the description provided in the database acquisition documentation.

2.2 Average Methods

The coherent average also referred to as the arithmetic mean (M_Mean as denoted in this study), can be computed using the ensemble matrix formed by the simulated M evoked responses [7].

In this context, the response p_i to the i-th stimulus is considered to be the sum of the deterministic component of the evoked signal or response s, along with an asynchronous random noise r. The model for each of the M simulated responses can be expressed using formula 1:

$$p_i = s + r_i\,, \qquad 1 \le i \le M. \tag{1}$$

The estimation of the deterministic component of the signal, denoted as $\hat{s}$, can be obtained using formula 2, with N representing the number of samples comprising each response [1]:

$$\hat{s}(n) = \frac{1}{M}\sum_{i=1}^{M} p_i(n), \quad 1 \le n \le N. \tag{2}$$

The application of signal averaging assumes that the underlying noise is stationary and follows a normal distribution with a mean of zero. Additionally, the noise variance should be consistent and equal across all potentials.

However, this condition is not always met, which can impact the effectiveness of the coherent average. To address this limitation, various methods have been proposed in the literature, including weighted averaging and robust averaging [1]. In the case of estimating the deterministic component of the signal, a weighted average approach is employed [8], as described by formula 3:

$$\hat{s}_w(n) = \sum_{i=1}^{M} w_i p_i(n), \quad 1 \le n \le N. \tag{3}$$

In the weighted average (Weighted_Mean) approach, each evoked response is assigned a weight based on specific criteria. One commonly used criterion is to assign weights based on the variance of the estimated noise in each response.

In this method, a smaller weight is assigned to potentials with higher levels of noise [5, 9, 11, 12]. Equation (4) represents the formulation corresponding to these weight assignment criteria:

$$w_i = \frac{1}{\sigma_i^2}\left(\sum_{j=1}^{M}\frac{1}{\sigma_j^2}\right)^{-1}, \quad i = 1, \cdots, M. \tag{4}$$

In formula 4, σ_i^2 represents the noise variance in the i-th potential. If the noise variance were constant across all records, the optimal value of w_i would be $1/M$, corresponding to the traditional average. Both the coherent average and the weighted average are linear techniques and perform well when the noise follows a Gaussian distribution [1, 8].

However, these techniques have limitations when occasional artifacts with large amplitude values (outliers) are present. In the literature, a family of estimators known as trimmed mean has been proposed to mitigate contamination by outliers [1, 5, 8]. The trimmed estimators are based on the median, which serves as the Maximum Likelihood (ML) estimator of s when the noise is assumed to follow a Laplacian distribution [1].

To compute the median, the samples in the ensemble matrix are ordered based on their amplitudes independently for each time point relative to the stimulus, regardless of other time points. This independence allows the median averaging to be unaffected by the nonstationarity of the noise within an epoch.

In the trimmed mean methods, the coherent average is combined with the median to obtain the

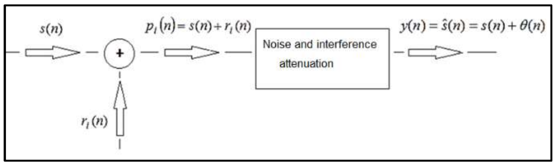

$s(n)$
$p_i(n) = s(n) + r_i(n)$
Noise and interference attenuation
$y(n) = \hat{s}(n) = s(n) + \theta(n)$
$r_i(n)$

Fig. 2. Signal and noise modelling

final response. The ensemble matrix is ordered, and a portion of extreme values is discarded or modified, while all other values are used for averaging similarly to conventional mean averaging. It is important to note that the rejection of extreme values differs from the concept of artifact rejection.

The *α*-trimmed mean (Trimmed_Mean) is one of the most popular trimmed estimators [9]. Equation (5) represents the estimation of the deterministic signal using the α-trimmed mean:

$$\hat{s}_{rec}(n) = \frac{1}{M - 2 \cdot K} \sum_{i=K+1}^{M-K} p_i\,(n). \tag{5}$$

If we compare formula 5 with formula 3, it becomes apparent that the weights can be assigned using the following equation:

$$w_i = \begin{cases} \dfrac{1}{M - 2 \cdot K}, & K + 1 \leq i \leq M - K, \\ 0, & in\ other\ case, \end{cases} \tag{6}$$

where α represents the percentage of trimming, M denotes the number of responses, and $K = \alpha M$ corresponds to the number of observations that are eliminated from each extreme of the ordered matrix.

2.3 Quality Measures

Given that the acquired signal (*p*) can be modelled as the desired signal (*s*) plus additive noise *(r*), as shown in formula 1, by applying different techniques to estimate the desired signal, attenuating the different existing interferences (Fig. 2). The output (*y*), after applying these techniques to estimate the desired signal ($\hat{s}$), can be seen as the combination of the desired signal (*s*) plus a remaining noise (*θ*).

The quality of the estimation of a signal segment can be expressed in terms of several parameters. Some of them are the signal-to-noise ratio (SNR) and the bias, which are expressed in equations 7 and 8, respectively [10]:

$$SNR = \frac{\sum_{j=1}^{N} s^2[j]}{\sum_{j=1}^{N} \theta^2[j]}, \tag{7}$$

$$b_\theta = \frac{1}{N} \sum_{j=1}^{N} |\theta[j]|. \tag{8}$$

In each of the above equations, *N* represents the total number of samples of the segment to be evaluated, *θ* is the remaining noise in the signal (signal obtained after attenuating the noise minus the ideal signal). The subscript *j* refers to the *j*th sample of the affected parameter and *s* is pure ideal signal. It is important to reach a compromise between bias and SNR (Equations 7 and 8).

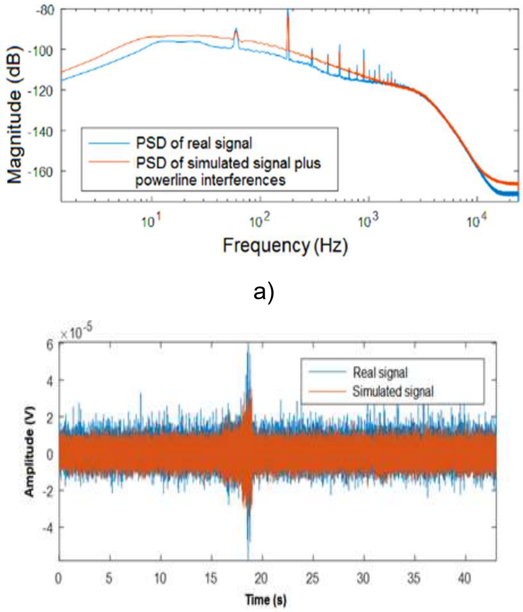
Since bias is the factor that indicates the distortion that is introduced by using a given noise and interference attenuation method, it is necessary to achieve high SNR values but low bias values. Unfortunately, in real situations, the pure ideal signal is not available a priori, so it is impossible to use these measures. But in a controlled environment, such as when using simulated signals, these measures can be used.

2.4 Experiment Description

To evaluate the averaging methods described using simulated signals, 100 data sets of 2,000 epochs each were obtained, without adding artifacts, and then the same 100 data sets of 2,000 epochs, where the 10%, 20%, and 30 % of the total samples of the array were randomly contaminated with outliers.

It was decided to add this level of artifacts based on other experiments found in the literature [6]. From each data set, 512 epochs were randomly selected, 100 times, thus forming a Monte Carlo experiment of 100 runs.

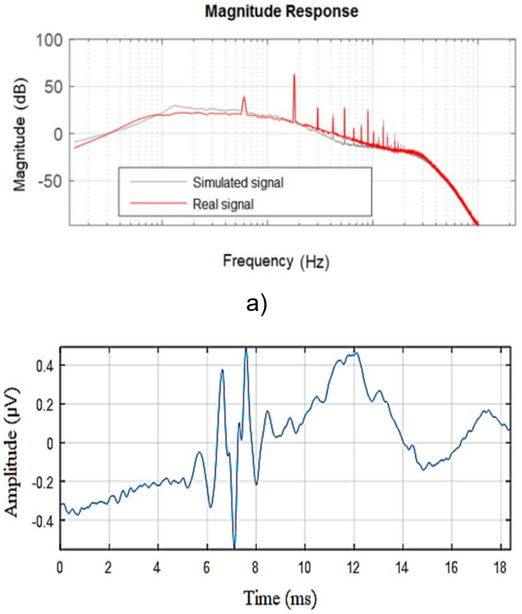
Evoked responses were then estimated using the classic Ensemble Average (M Mean), the Trimmed Average (Trimmed Mean), with a 20% trim factor, and the Weighted Average (Weighted Mean). The signal-to-noise ratio and bias values were calculated on the estimated signals to compare the estimation methods.



a)

b)

Fig. 3. a) The spectra of the simulated background noise and the reference signal. b) Example of the simulated background noise and the reference signal in the time domain



a)

b)

Fig. 4. a) Spectral comparison between simulated and real records. b) initial epoch

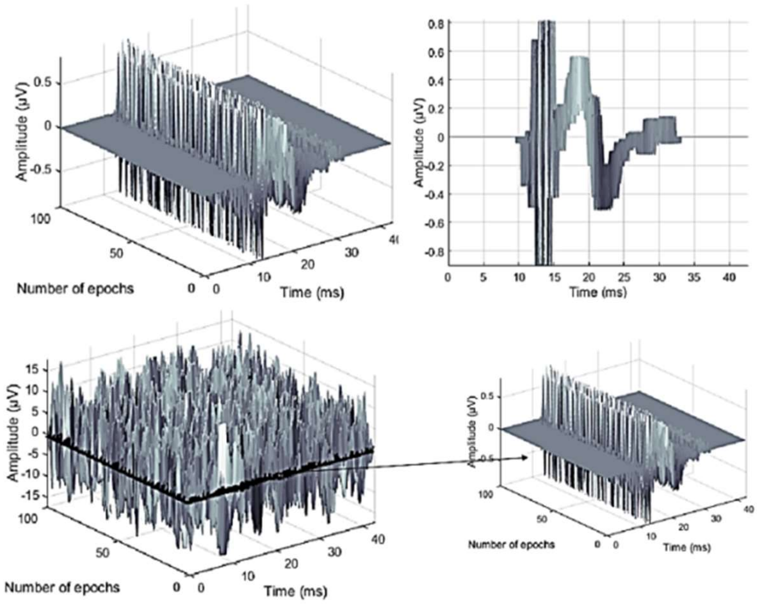


Fig. 5. Effects of relative displacements that may be present in a real scenario and of low SNR that may be present in a real environment

3 Results and Discussion

3.1 Simulated Noised

To generate the simulated noise signals, we first generated Gaussian white noise, which was subsequently subjected to low-pass filtering using coefficients estimated by the model. Next, the noise was high-pass filtered using a filter with the specifications outlined in Table 1.

Additionally, 60 Hz powerline interference was added to the noise signal. In Fig. 3a, we can observe a comparison of the spectra between one of the background noise signals simulated using this approach and the actual reference signal.

Figure 3b illustrates the comparison of the signals in the time domain. It is worth noting that the analysis was conducted on one-second segments of the signal to ensure stationary conditions. The simulated signal demonstrates the variations that have occurred in the signal's variance over time.

3.2 Simulated EP (Evoked Potential) Records

Figure 4a presents a comparison in the frequency domain between a simulated signal generated according to the specifications described earlier. In this case, the initial epoch, denoted as $x_o(t)$, corresponds to subject 6 and was selected randomly. To perform the spectral comparison, the "dirty" record that served as the source for the initial epoch (Fig. 4b) was chosen.

The tests yielded a consistent NRMSE adjustment of 92.5%. Randomly selected initial epochs were used, and simulated noise, interferences, and artifacts were added, resulting in a signal-to-noise ratio of -26.71 dB. The top part of Figure 5 visually demonstrates the effects of relative displacements that can occur in real scenarios.

This simulation example includes noise, interference, and artifacts. The lower part of Figure 5 shows the ensemble matrix, which combines evoked responses with noise and interference. Due to the high level of noise and interference, with an initial signal-to-noise ratio of -26.71 dB, it is not possible to discern any waveform associated with the desired signal.

Table 2. SNRs in dB were obtained with a Monte Carlo experiment of 100 runs on Simulated Data Sets

Average Methods	0% artifacts	10% artifacts	20% artifacts	30% artifacts
Initial SNR (dB)	-26.04 ± 1.17	-30.02 ± 1.06	-32.54 ± 0.93	-34.15 ± 1.01
Mean	-0.20 ± 1.09	-5.90 ± 0.35	-5.68 ± 0.78	-7.92 ± 0.23
Weighted Mean	1.96 ± 0.29	0.82 ± 0.04	-0.25 ± 0.20	-1.30 ± 0.04
Trimmed Mean	-0.66 ± 0.77	-3.88 ± 0.53	-0.81 ± 0.35	-1.83 ± 0.33

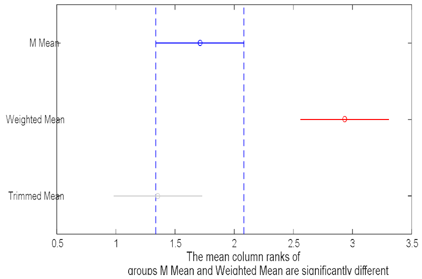


Fig. 6. Differences between the mean SNR values obtained for the averaging methods in the data set without artifacts

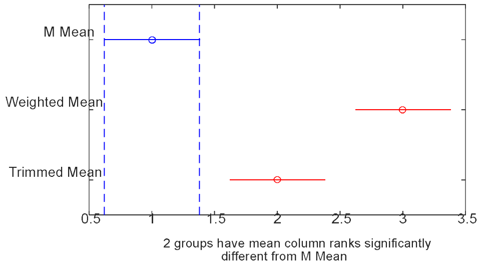


Fig. 7. Differences between the mean SNR values obtained for the averaging methods in the data set with 10% of the samples with artifacts

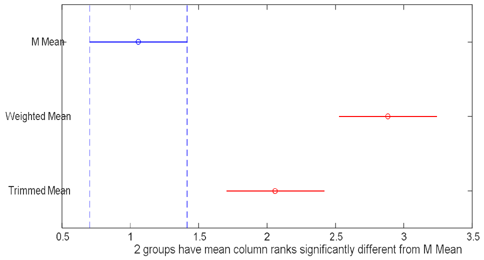


Fig. 8. Differences between the mean SNR values obtained for the averaging methods in the data set with 20% of the samples with artifacts

It should be noted that the achieved level of realism in this simulation is significantly higher than in previous simulations, making direct comparisons with previous simulations inappropriate.

The codes used in these simulations are available in the GitHub repository for benchmarkingpurposes[1].

3.3 Estimation of Event-Related Potentials Using Realistic Simulation

Table 2 displays the average Signal-to-Noise Ratio (SNR) values and their corresponding standard deviations obtained from the experiment described in section 2.4.

The results indicate that the Weighted Mean method consistently yielded the highest SNR values across all cases. This suggests that the Weighted Mean method performed better in terms of minimizing the impact of noise and maximizing the clarity of the desired signal compared to the other methods evaluated in the experiment.

Based on the results obtained, a Friedman test was performed to analyze whether there were significant differences in at least one of the averaging methods used for estimation. A value of $p < 0.05$ was obtained, so at least two combinations have significant differences concerning their means.

A post-hoc was performed using the Bonferroni test with an alpha of 0.05 to determine which combinations have differences. Figure 6 shows the multicomparison between the three average methods used, it can be seen how there are differences between the Weighted Mean method with respect to the other two. The differences between the M Mean and Trimmed Mean are not significant.

A similar analysis was performed when 10%, 20%, and 30% of the samples were contaminated (Fig.7 - Fig.9). In this case, the results of the SNR values have significant differences between the three methods. No critical distance overlaps. In all cases, with 0% artifacts, 10%, 20% and 30%, the Weighted_Mean method offered the best results in terms of the value of the signal-to-noise ratio.

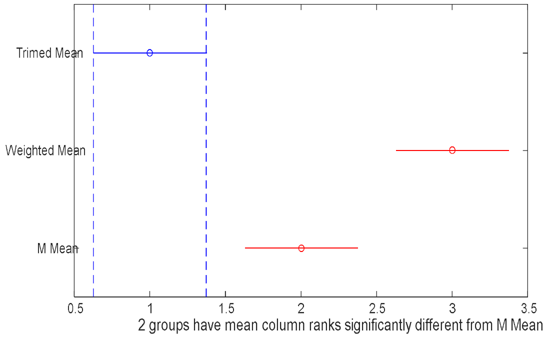


Fig. 9. Differences between the mean SNR values obtained for the averaging methods in the data set with 30% of the samples with artifacts

Table 2. Modified bias in μV obtained with a Monte Carlo experiment of 100 runs on a Simulated Database

Average Methods	0% artifacts	10% artifacts	20% artifacts	30% artifacts
Mean	0.16 ± 0.07	0.24 ± 0.03	0.28 ± 0.02	0.31 ± 0.02
Weighted Mean	0.09 ± 0.01	0.10 ± 0.01	0.23 ± 0.01	0.19- ± 0.01
Trimmed Mean	0.17 ± 0.01	0.21 ± 0.01	0.20 ± 0.01	0.15 ± 0.01

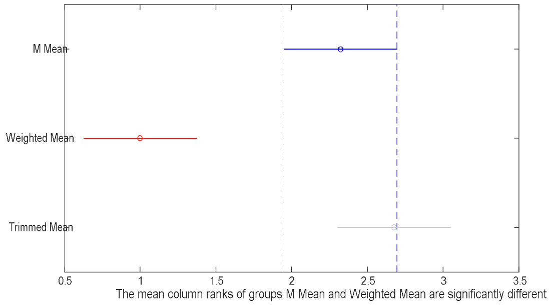


Fig. 10. Differences between the mean values of bias obtained for each averaging method in the data set without artifacts

With the bias, an analysis similar to that performed with the SNR was performed, and the lowest distortion values of the resulting signal were obtained for the weighted average when there

[1] https://github.com/itrodriguez/SimuldorEP/tree/main

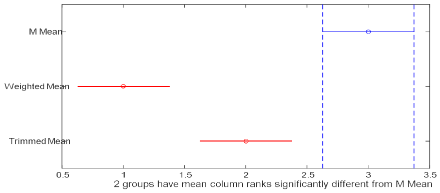


Fig.11. Differences between the mean values of bias obtained for each averaging method in the data set with 10% of the samples with artifacts

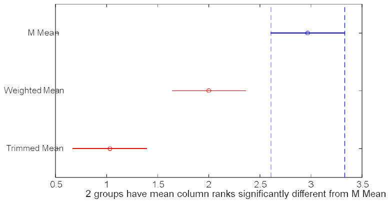


Fig. 12. Differences between the mean values of bias obtained for each averaging method in the data set with 20% of the samples with artifacts

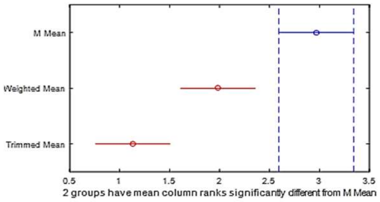


Fig.13. Differences between the mean values of bias obtained for each averaging method in the data set with 30% of the samples with artifacts

were 0% artifacts and for the trimmed average when there were 30% artifacts.

Figures 10, 11, 12, and 13 show the Bonferroni-Holm post hoc analysis with alpha equal to 0.05 to assess the differences between the mean values of bias obtained for the data set without artifacts and with outliers, after finding through a Friedman test that at least one of the methods had significant differences.

When the data set has 0% artifacts, the lowest degree of distortion is presented by the weighted average, with significant differences concerning the average and trimmed average, however, there are no significant differences between these last two methods.

When the samples are contaminated with artifacts, the best results are offered by the trimmed mean for 20% and 30%, significantly different from the other two methods.

In the case of the SNR calculation, it is not the one that offers the highest value, but let us remember that the objective of these two measures is to provide a compromise ratio. So, when there are artifacts, the best compromise is offered by the trimmed mean.

4 Conclusions

Simulated raw recordings of evoked potentials provide a controlled dataset for benchmarking new methods in detecting evoked responses.

Burg's method, utilizing an 8th-order autoregressive (AR) model, offers a reliable estimate of the background noise. Simulations can also incorporate interferences commonly found in real signals, such as 60 Hz powerline interference, alpha rhythm, and instrumentation channel noises. Furthermore, the simulation scheme allows for the introduction of out-of-range values and impulsive noise.

In the benchmarking study of Averaging Methods using Realistic Simulation of Evoked Potentials, it was observed that the weighted average method yields superior results when the data is free from artifacts.

However, in cases where artifacts are present, the trimmed mean method provides the best compromise in terms of performance.

Acknowledgments

This work has been carried out thanks to the doctoral program of Informatics and Computational Sciences of the Central University "Marta Abreu" of Las Villas.

References

1. **Krol, L. R., Pawlitzki, J., Lotte, F., Gramann, K., Zander, T. O. (2018).** SEREEGA: Simulating event-related EEG activity. Neurosci Methods, Vol. 309, pp. 13–24. DOI: 10.1016/j.jneumeth.2018.08.001.

2. **Torres-Rodríguez, I., Díaz-Amador, R., Peón-Pérez, B., Hurtado Armas, A., Taboada-Crispi, A. (2023).** Realistic simulation of event-related potentials and their usual noise and interferences for pattern recognition. In: Rodríguez-González, A.Y., Pérez-Espinosa, H., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Olvera-López, J.A. (eds) Pattern Recognition, Proceedings of 15 th Mexican Conference Pattern Recognition, 2023. Lecture Notes in Computer Science, pp. 201–210. DOI: 10.1007/978-3-031-33783-3_19.

3. **Silva, I., Epstein, M. (2010).** Estimating loudness growth from tone-burst evoked responses. The Journal of the Acoustical Society of America, Vol. 127, No. 6, pp. 3629–3642. DOI: 10.1121/1.3397457.

4. **Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Stanley, H. E. (2000).** Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. Circulation, Vol. 101, No. 23, pp. e215–e220. DOI: 10.1161/01.CIR.101.23.E215.

5. **Valderrama, J. T., De la Torre, A., Alvarez, I., Segura, J. C., Thornton, A. R. D., Sainz, M., Vargas, J. L. (2014).** Automatic quality assessment and peak identification of auditory brainstem responses with fitted parametric peaks. Computer methods and programs in biomedicine, Vol. 114, No. 3, pp. 262–275. DOI: 10.1016/j.cmpb.2014.02.015.

6. **Leonowicz, Z., Karvanen, J., Shishkin, S. L. (2005).** Trimmed estimators for robust averaging of event-related potentials. Journal of Neuroscience Methods, Vol. 142, No. 1, pp. 17–26. DOI: 10.1016/j.jneumeth.2004.07.008.

7. **Torres-Rodríguez, I., Ferrer-Riesgo, C. A., P. de Morales-Artiles, M. M., Taboada-Crispi, A. (2020).** Performance evaluation of average methods in the time domain using quality measures for automatic detection of evoked potentials. VIII Latin American Conference on Biomedical Engineering, CLAIB 2019, IFMBE prodeedings, Vol. 75, pp. 12–20, DOI: 10.1007/978-3-030-30648-9_2.

8. **Pander, T. (2015).** A new approach to robust, weighted signal averaging. Biocybernetics and Biomedical Engineering, Vol. 35, No. 4, pp. 317–327, DOI: 10.1016/j.bbe.2015.06.002.

9. **Torres-Rodríguez, I., Ferrer-Riesgo, C. A., Oliva Pérez, J. C., Taboada-Crispi, A. (2019).** Performance of different average methods for the automatic detection of evoked potentials. In: Nyström, I., hernández-Heredia, Y., Milián-Núñez, V. (eds.) Iberoamerican Congress on Pattern Recognition, Springer, Vol. 11896, pp. 629–636, DOI: 10.1007/978-3-030-33904-3_59.

10. **Novis, K. Bell, S. (2019).** Objective comparison of the quality and reliability of auditory brainstem response features elicited by click and speech sounds. Ear Hear, Vol. 40. No. 3, pp. 447–457, DOI: 10.1097/AUD.0000000000000639.

Article received on 04/07/2023; accepted on 13/10/2023.
Corresponding author is Idileisy Torres-Rodríguez.

ISSN 2007-9737

Evaluation of CNN Models with Transfer Learning in Art Media Classification in Terms of Accuracy and Class Relationship

Juan Manuel Fortuna-Cervantes[1], Carlos Soubervielle-Montalvo[*,2],
Cesar Augusto Puente-Montejano[2], Oscar Ernesto Pérez-Cham[3],
Rafael Peña-Gallardo[2]

[1] Instituto Tecnológico de San Luis Potosí,
Tecnológico Nacional de México,
Mexico

[2] Universidad Autónoma de San Luis Potosí,
Facultad de Ingeniería,
Mexico

[3] Universidad del Mar,
Instituto de Industrias,
Mexico

juan.fc@slp.tecnm.mx, {carlos.soubervielle, cesar.puente,
rafael.pena}@uaslp.mx, operezcham@zicatela.umar.mx

Abstract. The accuracy obtained in Art Media Classification (AMC) using CNN is lower compared to other image classification problems, where the acceptable accuracy ranges from 90 to 99%. This article presents an analysis of the performance of three different CNNs with transfer learning for AMC, to answer the question of what challenges arise in this application. We proposed the Art Media Dataset (ArtMD) to train three CNNs. ArtMD contains five classes of art: Drawing, Engraving, Iconography, Painting, and Sculpture. The analysis of the results demonstrates that all the tested CNNs exhibit similar behavior. Drawing, Engraving, and Painting had the highest relationship, showing a strong relationship between Drawing and Engraving. We implemented two more experiments, removing first Drawing and then Engraving. The best performance with 86% accuracy was achieved by removing Drawing. Analysis of the confusion matrix of the three experiments for each CNN confirms that Drawing and Painting have the lowest accuracy, showing a strong misclassification with the other classes. This analysis presents the degree of relationship between the three CNN models and details the challenges of AMC.

Keywords. Art media classification, convolutional neural networks, transfer learning.

1 Introduction

Art restorers and collectors frequently classify art media by evaluating their physical features, subjective characteristics, and historical periods [16]. However, this classification process can be challenging because specific attributes may need to fit neatly into predefined styles, genres, or art periods, leading to potential misclassification.

A favorable solution to this challenge involves the utilization of Convolutional Neural Networks (CNNs). These deep learning algorithms have garnered recognition in the scientific community for their prowess in image classification and object detection tasks [2, 17, 22].

Although there is growing interest in CNNs for Art Media Classification (AMC), limited research delves deeply into their classification performance and class relationship [12, 20]. Furthermore, there is a growing inclination towards pre-trained models over traditional computer vision methods, demonstrating the potential for achieving more accurate dataset classification [7].

doi: 10.13053/CyS-28-1-4895

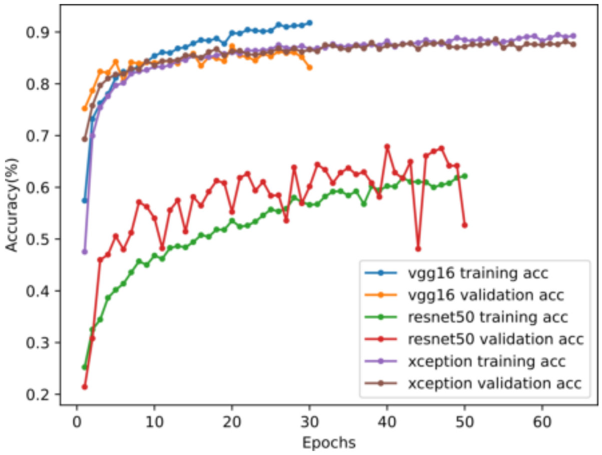


Fig. 1. Image distribution and composition for the Art Media Dataset (ArtMD)

In a primary study, serving as a basis for this work [8], an assessment of the performance accuracy was conducted on three well-established CNN architectures in AMC.

The principal objective is to significantly emphasize the resilience of CNN learning models in art media classification when leveraging transfer learning. This study of the three proposed CNN architectures seeks to determine the optimal choice for future applications.

Based on the insights gained from previous work, this study presents a comprehensive evaluation and performance analysis of three well-known CNN architectures in the context of AMC, aiming to address the challenges that arise when using CNNs with transfer learning [14].

In addition, it investigates the relationship between classes to shed light on poor classification performance and how dataset characteristics influence CNN learning. The main contributions of this study are as follows:

1) Introduction of an experimental approach to evaluate CNN performance in the Art Media Classification (AMC) context and to demonstrate that AMC represents a problem in the accuracy of the classifier, being an area of opportunity in the development of CNN.

2) Creation of the Art Media Dataset (ArtMD), used for training and evaluating the classification model. The dataset combines digitized artworks sourced from diverse repositories, including the Kaggle website, the WikiArt database, and institutional archives from the Prado National and the Louvre National Museum. The proposal can be considered a standard for evaluating CNN models in AMC.

3) Evaluation of three state-of-the-art CNN models in AMC highlights that accurate inferences can be drawn for most classes of art media, with a notable finding that Drawing and Engraving exhibit a strong relationship with each other.

4) Conducting additional experiments by removing Drawing and Engraving, which accentuates a slight relationship with the Painting class across all remaining classes (Iconography, Sculpture, and Engraving).

Furthermore, a high relationship is observed between the predicted class and the original label for Iconography and Sculpture classes. These relationship effects can be seen in these experiments for all CNN models, as presented in the Experiments and Results section. This article unfolds as follows: Section 2 briefly overviews the work related to AMC.

Section 3 delves into the materials and methods. Section 4 contains experimental details, presents results, and analyzes the classification outcomes. We showcase the accuracy and interclass relationships of the devised image classifiers, which remain unexplored in the current state-of-the-art. Finally, in Section 5, we end the paper with some conclusions and ideas for future work.

2 Related Work

Computer vision has become an intriguing approach for recognizing and categorizing objects across various applications. It is an auxiliary tool that mimics human visual perception, opening doors to various practical applications. One of these applications pertains to safeguarding data against adversarial attacks. Deep Genetic Programming (DGP) employs a hierarchical structure inspired by the brain's behavior to extract image features and explore the transfer of adversarial attacks within artwork databases.

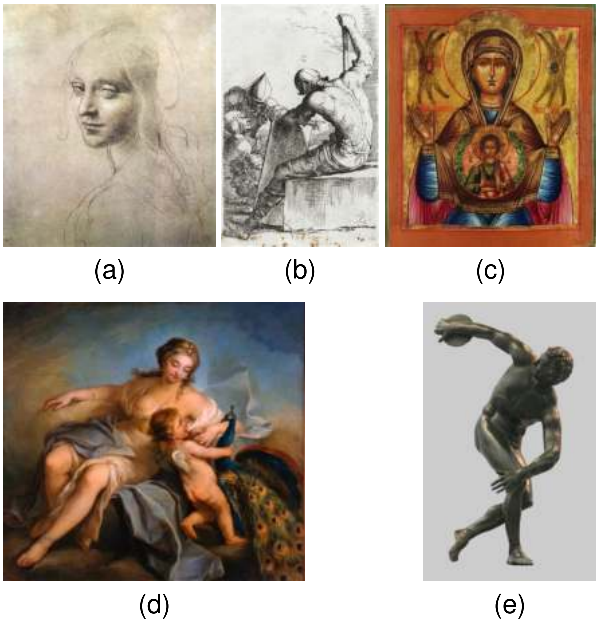


Fig. 2. Art Image training set exhibits the five art categories: (a) Drawings produced using a pencil, pen, or similar tools on paper or another medium; (b) Engravings, images crafted through cutting or etching into a surface; (c) Iconography, encompassing religious images or symbols; (d) Paintings, artworks generated by applying pigments onto a surface; and (e) Sculptures, representing three-dimensional art forms shaped or modeled from materials to achieve a specific form

In this context, the application focuses on adversarial attacks in categorization [20]. The paper [11] presents a comparative study on the impact of these attacks within the art genre categorization, involving feature analysis and testing with four Convolutional Neural Networks (AlexNet, VGG, ResNet, ResNet101) alongside brain-inspired programming.

Deep learning algorithms have significantly advanced image classification, particularly in [18], where pre-trained networks like VGG16, ResNet18, ResNet50, GoogleNet, MobileNet, and AlexNet are utilized on the Best Artworks of all Time dataset.

After adjusting training parameters, the study selects the best model, finding that ResNet50 achieves the highest accuracy among all other deep networks.

In [15], the focus shifts to style classification using the Painter by Numbers dataset, encompassing five classes: impressionism, realism, expressionism, post-impressionism, and romanticism. The model is based on a pre-trained ResNet architecture from the ImageNet dataset and is refined by different transformations, such as random affine transform, crop, flip, color fluctuations and normalization.

Additionally, the papers [6, 5] explore further the correlation between feature maps, which effectively describe the texture of the images. These correlations are transformed into style vectors, surpassing the performance of CNN features from fully connected layers and other state-of-the-art deep representations.

Furthermore, the introduction of inter-layer correlations is proposed to enhance classification efficiency. In [21], a novel approach is presented to improve the classification accuracy of fine art paintings. This approach combines transfer learning with subregion classification, utilizing the weighted sum of individual patch classifications to obtain the final statistical label for a given painting.

The method offers computational efficiency and is validated using standard artwork classification datasets with six pre-trained CNN models. Further, [1] employs two machine learning algorithms on an artwork dataset to demonstrate that features derived from the artwork play a significant role in accurate genre classification.

These features encompass information about the nationality of the artists and the era in which they worked. Finally, in [9], VGG19 and ResNet50 are applied to classify artworks based on their style. The study compares their performance in recognizing underlying features, including aesthetic elements.

The dataset is derived from The Best Artworks in the World, selecting five subsets from artists with distinct styles. The results indicate that CNNs can effectively extract and learn these underlying features, with VGG19 showing preference for subjective items and ResNet50 with favoring objective markers. In summary, our work has two main differences from related works: Firstly, this work presents an in-depth study of CNN models in AMC, which can be used to

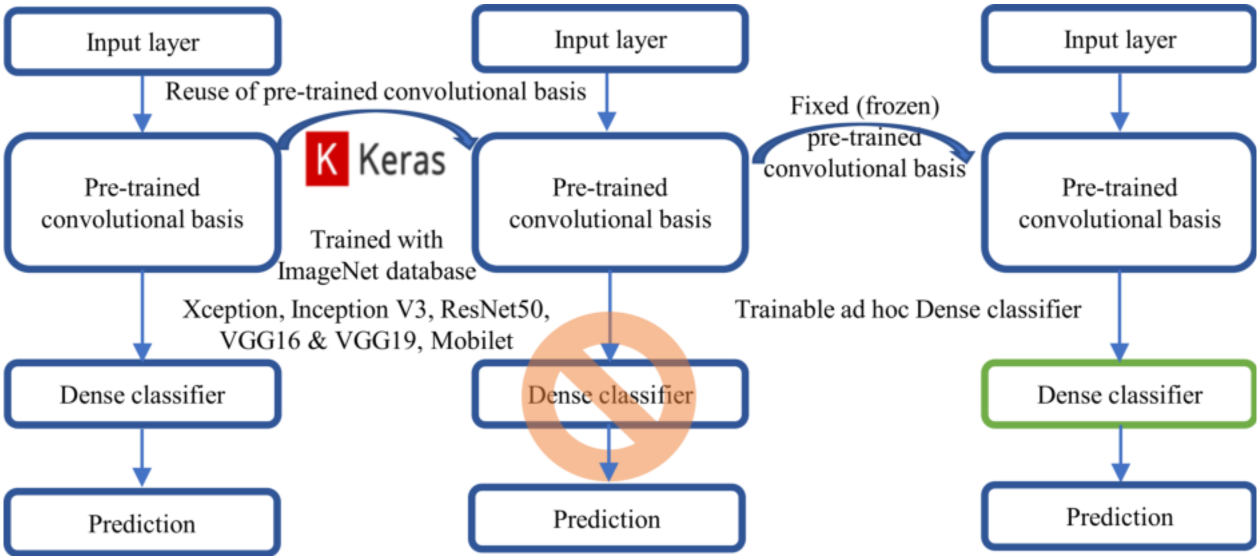

Input layer
Reuse of pre-trained convolutional basis
Keras
Pre-trained convolutional basis
Trained with ImageNet database
Xception, Inception V3, ResNet50, VGG16 & VGG19, Mobilet
Dense classifier
Prediction
Input layer
Pre-trained convolutional basis
Dense classifier
Prediction
Fixed (frozen) pre-trained convolutional basis
Trainable ad hoc Dense classifier
Input layer
Pre-trained convolutional basis
Dense classifier
Prediction

Fig. 3. Process for the reuse of convolutional pre-trained networks (feature-based transfer learning)

understand the difficulties in this task and find new alternatives to improve the performance. Secondly, a detailed analysis of accuracy and class relationship is presented using a proposed dataset consisting of the Art Image dataset, the WikiArt database, and digital artworks from The Louvre and Prado Museums.

3 Materials and Methods

3.1 Dataset

Information is the paramount element in deep learning tasks, particularly in the Art Media Classification (AMC) domain. The Art Image dataset [20] assumes significance. This dataset includes training and validation images sourced from the Kaggle website's repository of digitized artworks.

The dataset contains five art media categories: Drawing, Painting, Iconography, Engraving, and Sculpture. We opted to formulate the Art Media Dataset (ArtMD)[1], as illustrated in Fig. 1. This decision was prompted by the existence of corrupted or preprocessed images within the original dataset.

The dataset consists of the same five classes, each comprising 850 images for training and 180 for validation, originating from the Art Image dataset. For the test set, 180 images per category were curated from the WikiArt database and digital artworks from the Louvre National Museum[2] for Painting and the Prado National Museum[3] for Engraving. A notable characteristic of this dataset is the RGB format, each with a size of 224×224, ideal for the input requirements of the proposed architectures. Fig. 2 showcases a selection of random images from the training set.

3.2 CNN Architecture and Transfer Learning

Several Convolutional Neural Network (CNN) architectures are available for addressing real-world challenges associated with image classification, detection, and segmentation [3, 10, 24]. However, each architecture has distinct advantages and limitations concerning training and implementation. Choosing the most suitable architecture involves experimentation and relies on the specific performance requirements and intended application.

[1] github.com/JanManuell/Art-Media-Classification---Dataset.git

[2] collections.louvre.fr/en/

[3] www.museodelprado.es/coleccion/obras-de-arte

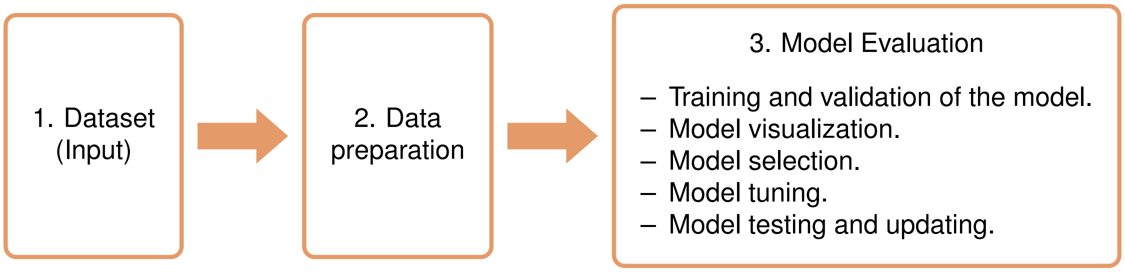


Fig. 4. Process to improve the classification model

When trading with limited datasets in deep learning, Transfer Learning emerges as a popular approach [4]. The idea behind Transfer Learning is that a Convolutional Neural Network (CNN) previously trained on a large and diverse dataset, such as ImageNet, has already acquired knowledge about general and useful features present in the images, such as edges, textures, and shapes.

These features can be reused in a specific task without the need to train a network from scratch. The CNN architecture proposed contains two elements: the feature extraction stage and the classification stage. Feature extraction involves the use of previously learned representations during the original training.

The pre-trained network is taken in this stage, and the output layers designed for the original task are removed. The convolutional layers in charge of feature extraction are retained, which will process the images of the new task.

Then, in the classification stage, additional layers, such as fully connected and output layers, are added at the end of the network to adapt it to the new features of the specific dataset (feature-based transfer learning). After that, the complete network is trained with the dataset, and its performance is evaluated using task-relevant metrics, as shown in Fig. 3.

3.3 Improving Model Classification

The proposed methodology for improving the learning model's performance can be summarized in three key stages. In the first stage, the integration of the dataset is carried out.

It is essential that this dataset presents a balance between classes and contains images representative of the problem being addressed. In the second stage, the images are processed. The pixel values are normalized to ensure that the model converges efficiently during training. The third stage focuses on model validation. In this stage, the training parameters are adjusted and updated, allowing the learning model to be retrained to perform better, as shown in Fig. 4.

3.3.1 Model Evaluation

The model's classification accuracy improvement process involves iterative testing, selecting initial training parameters, and automatic feature extraction through optimal kernel filters. This enables subsequent model adjustments. Evaluation relies on Accuracy, measuring the percentage of correct predictions, while the confusion matrix, an N×N table (N being the number of classes), analyzes patterns of prediction errors by revealing the relationship between predicted and actual labels.

3.3.2 Network Training and Parameter Settings

The models are implemented using the Python programming language and the `Keras API` with `Tensorflow` as the backend. The training was conducted utilizing an NVIDIA Tesla K80 GPU within the `Google Colaboratory`[4] (Colab) environment. Colab's GPU, a graphics processor in the system, accelerates the result epoch.

[4] colab.research.google.com/

Table 1. Training parameters of the proposed model

Hyperparameter	Value
Learning rate	0.0001
Minibatch	16 or 32
Loss function	’categorical_crossentropy’
Metrics	’acc’,’loss’
Epochs	500
Optimizer	Adam
Callbacks API	
ModelCheckpoint	Monitor = ‘val_loss’, save_best_only = True, mode=‘min’
EarlyStopping	Monitor = ‘val_acc’, patience = 15, mode = ‘max’
CVLogger	‘model_history.csv’, append = True
ReduceLROnPlateau	Monitor = ‘val_los’, factor=0.2, patience=10, min_lr = 0.001

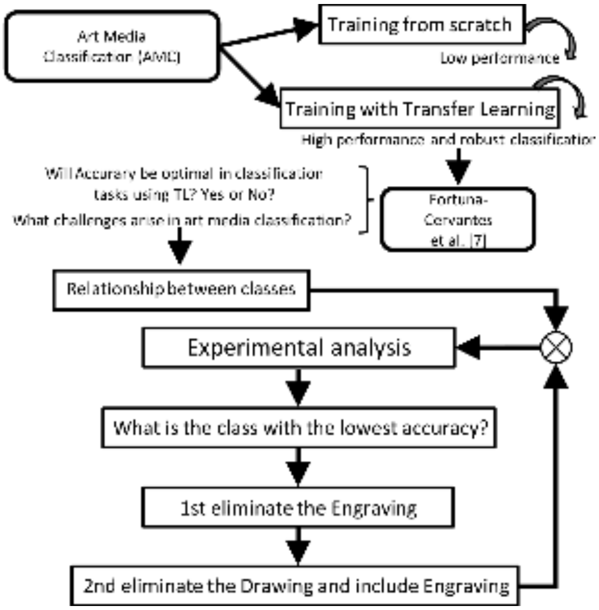

Art Media Classification (AMC)
Training from scratch
Low performance
Training with Transfer Learning
High performance and robust classification
Will Accuracy be optimal in classification tasks using TL? Yes or No?
What challenges arise in art media classification?
Fortuna-Cervantes et al. [7]
Relationship between classes
Experimental analysis
What is the class with the lowest accuracy?
1st eliminate the Engraving
2nd eliminate the Drawing and include Engraving

Fig. 5. Workflow to analyze the performance of CNNs and the relationship between classes

Notably, Colab determines itself by offering free GPU and TPU support during runtime, extending up to 12 hours in some instances, unlike other cloud systems. The base architectures used are the VGG16, ResNet50, and Xception networks, renowned for their early success in large-scale visual recognition challenges such as ILSVRC [24].

Before training each CNN, it is essential to define the loss function-indicating how the network measures its performance on the training data and guides itself in the desired direction (also known as the objective function) and the optimizer-dictating how the network updates itself based on the observed data and its loss function.

These parameters control the adjustments to the network weights during training. Additionally, regularization techniques, including DropOut (DO) [25], Data Augmentation [19], and Batch Normalization (BN) [23], are incorporated.

A Callback, serving as an object capable of executing actions at different stages of training (e.g., ModelCheckpoint for saving the Keras model, EarlyStopping to halt training when a metric plateau, CSVLogger for logging epoch results in a CSV file, and ReduceLROnPlateau to decrease learning rate on metric stagnation), is integrated.

This holistic approach yields a learning model capable of predicting art media in dataset (test) images with enhanced Accuracy. The training parameters for the proposed models are detailed in Table 1.

4 Experiments and Results

In a previous study [8], three CNN architectures were evaluated for classifying art media, demonstrating the robustness of CNN learning models with a focus on transfer learning. This current work builds on those results, and a detailed evaluation of the same architectures in the context of AMC is performed. The main objective is to address the challenges when employing CNNs with transfer learning in this domain, in addition to analyzing the relationship between ArtMD classes to understand the poor classification performance and how the dataset influences the learning process of CNNs. The workflow for the proposed experimental study is depicted in Fig. 5. As described earlier, the learning models are built upon three foundational architectures: VGG16, ResNet50, and Xception. The models are trained using the ArtMD, incorporating images from the Kaggle website, WikiArt database, and digital artworks sourced from the Louvre Museum in France and the Prado Museum in Spain.

doi: 10.13053/CyS-28-1-4895

Table 2. Overview of the classification model performance on the Art Media Dataset [8]

<table>
<tr><th colspan="10">Setup 1: Pre-trained CNN base+Dense Classifier (GlobalAveragePooling2D(GAveP2D)+DO(0.2))</th></tr>
<tr><td>CNN</td><td>Params [M]</td><td>Epoch</td><td>Time [min]</td><td>loss</td><td>acc</td><td>val_loss</td><td>val_acc</td><td>test_loss</td><td>test_acc</td></tr>
<tr><td>VGG16</td><td>14.7</td><td>91 (90)</td><td>173</td><td>0.5983</td><td>0.7832</td><td>0.5699</td><td>0.7868</td><td>0.8017</td><td>0.6911</td></tr>
<tr><td>ResNet50</td><td>23.6</td><td>50 (49)</td><td>84</td><td>1.2745</td><td>0.4981</td><td>1.2295</td><td>0.5335</td><td>1.5442</td><td>0.4122</td></tr>
<tr><td>Xception</td><td>20.8</td><td>64 (64)</td><td>136</td><td>0.2920</td><td>0.8927</td><td>0.3470</td><td>0.8761</td><td>0.6792</td><td><b>0.7444</b></td></tr>
<tr><th colspan="10">Setup 2: Pre-trained CNN base+Dense Classifier (Dense(128)+D0(0.4)+Dense(64)+DO(0.2))</th></tr>
<tr><td>CNN</td><td>Params [M]</td><td>Epoch</td><td>Time [min]</td><td>loss</td><td>acc</td><td>val_loss</td><td>val_acc</td><td>test_loss</td><td>test_acc</td></tr>
<tr><td>VGG16</td><td>17.9</td><td>30 (14)</td><td>89</td><td>0.3313</td><td>0.8707</td><td>0.4026</td><td>0.8527</td><td>0.7551</td><td><b>0.7544</b></td></tr>
<tr><td>ResNet50</td><td>36.4</td><td>51 (47)</td><td>107</td><td>1.3590</td><td>0.3860</td><td>1.2926</td><td>0.4275</td><td>1.4392</td><td>0.3822</td></tr>
<tr><td>Xception</td><td>33.7</td><td>25 (15)</td><td>44</td><td>0.3437</td><td>0.8654</td><td>0.3614</td><td>0.8862</td><td>0.7967</td><td>0.7422</td></tr>
<tr><th colspan="10">Setup 3: Pre-trained CNN base+Dense Classifier (GAveP2D+Dense(64)+BN()+DO(0.4)+Dense(64)+BN()+DO(0.5))</th></tr>
<tr><td>CNN</td><td>Params [M]</td><td>Epoch</td><td>Time [min]</td><td>loss</td><td>acc</td><td>val_loss</td><td>val_acc</td><td>test_loss</td><td>test_acc</td></tr>
<tr><td>ResNet50</td><td>23.7</td><td>50 (40)</td><td>100</td><td>1.0438</td><td>0.6024</td><td>0.8845</td><td>0.6786</td><td>1.3535</td><td><b>0.5422</b></td></tr>
</table>

Table 3. Performance of classification models (Only four classes)

<table>
<tr><th colspan="10">Setup 4 (Engraving class was removed): Pre-trained CNN base+Dense Classifier (Dense(128)+DO(0.3)+Dense(64)+D0(0.2))</th></tr>
<tr><td>CNN</td><td>Params [M]</td><td>Epoch</td><td>Time [min]</td><td>loss</td><td>acc</td><td>val_loss</td><td>val_acc</td><td>test_loss</td><td>test_acc</td></tr>
<tr><td>VGG16</td><td>17.9</td><td>35 (18)</td><td>57</td><td>0.1422</td><td>0.9524</td><td>0.3070</td><td>0.8991</td><td>0.6550</td><td>0.8208</td></tr>
<tr><td>ResNet50</td><td>36.4</td><td>26 (4)</td><td>68</td><td>0.1806</td><td>0.9351</td><td>0.2454</td><td>0.9304</td><td>0.7702</td><td><b>0.8514</b></td></tr>
<tr><td>Xception</td><td>33.7</td><td>27 (14)</td><td>68</td><td>0.1318</td><td>0.9548</td><td>0.2615</td><td>0.9056</td><td>0.6025</td><td>0.8278</td></tr>
<tr><th colspan="10">Setup 5 (Drawing class was removed): Pre-trained CNN base+Dense Classifier (Dense(128)+DO(0.3)+Dense(64)+D0(0.2))</th></tr>
<tr><td>CNN</td><td>Params [M]</td><td>Epoch</td><td>Time [min]</td><td>loss</td><td>acc</td><td>val_loss</td><td>val_acc</td><td>test_loss</td><td>test_acc</td></tr>
<tr><td>VGG16</td><td>17.9</td><td>29 (19)</td><td>73</td><td>0.0696</td><td>0.9747</td><td>0.1412</td><td>0.9631</td><td>0.6434</td><td>0.8375</td></tr>
<tr><td>ResNet50</td><td>36.4</td><td>26 (4)</td><td>64</td><td>0.1147</td><td>0.9649</td><td>0.1195</td><td>0.9645</td><td>0.6549</td><td>0.8514</td></tr>
<tr><td>Xception</td><td>33.7</td><td>17 (4)</td><td>36</td><td>0.1549</td><td>0.9461</td><td>0.1343</td><td>0.9597</td><td>0.4589</td><td><b>0.8611</b></td></tr>
</table>

4.1 Classification Performance Evaluation

Table 2 illustrates a comparison between the reference models' accuracy and loss across different datasets (training, validation, and test) and the proposed setups to the base structure.

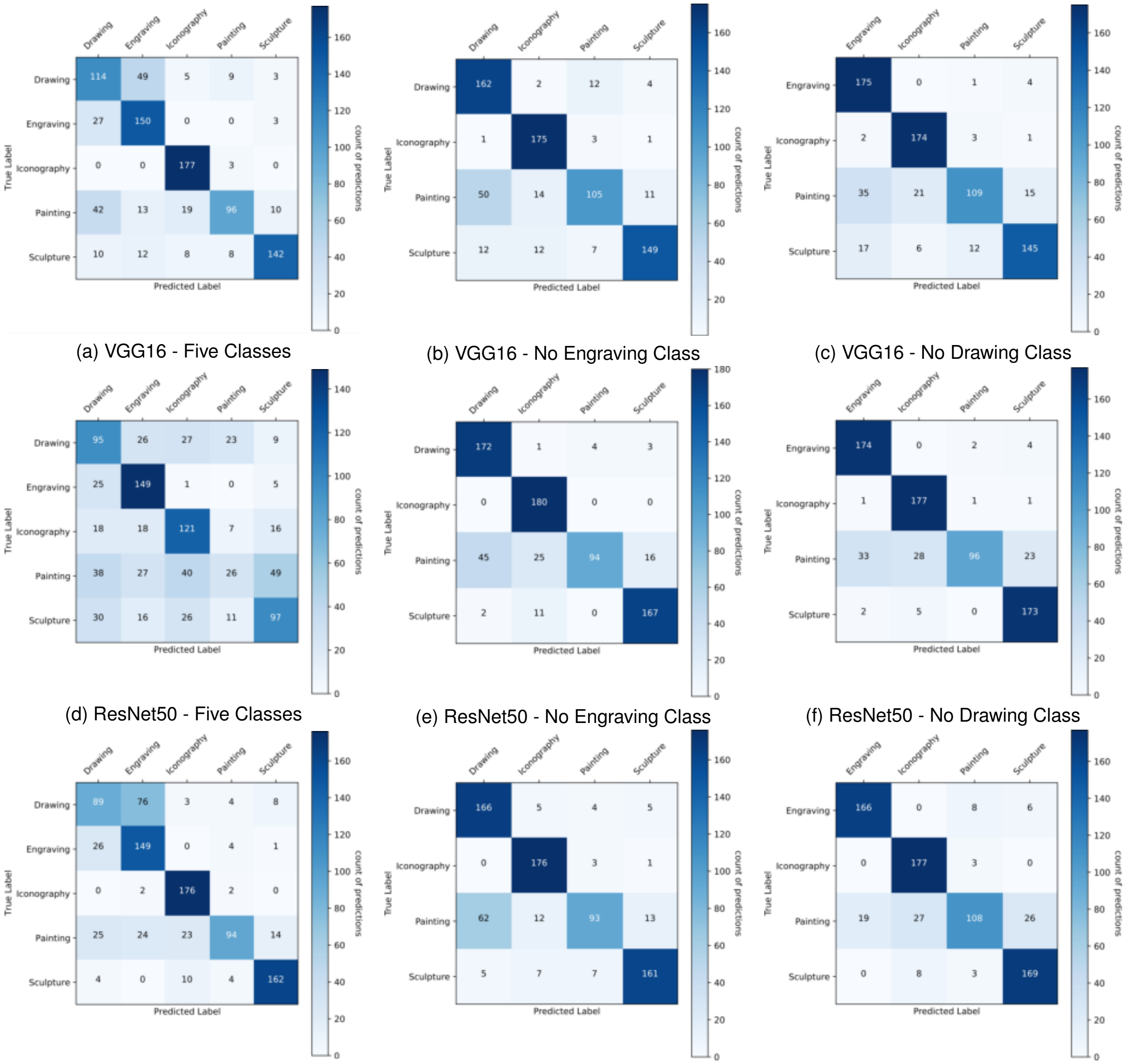
This initial investigation delves into the CNNs' performance concerning each dataset class. Notably, the Xception model excels, achieving the highest classification accuracy of 74% in the first setup. Conversely, the VGG16 model attains its peak performance with 75% accuracy in the second setup.

The ResNet50 model exhibits a lower accuracy in the test set compared to the training and validation sets. In a third setup focusing on enhancing classification performance through the dense classifier, the ResNet50 model demonstrates acceptable performance with an accuracy of 54%.

Furthermore, this proposed approach features a reduced number of training parameters compared to its predecessor. The accuracy of the proposed models, in particular, maintains homogeneity when training with the training and validation sets. This is expected because there is a control to avoid model overfitting.

The proposed regularization methods and Callbacks are integrated into the architecture to eliminate overfitting to monitor the learning process. With the test information, the base models achieve an accuracy below the training and validation set.

Interestingly, the models predict images (test) that have never been used for training, meeting the goal of generalization of knowledge in CNNs, but not enough to achieve the optimal performances reported in classification tasks. Fig. 6a, 6d and 6g show the confusion matrix for the test set (with five classes) in the three models.



(a) VGG16 - Five Classes

(b) VGG16 - No Engraving Class

(c) VGG16 - No Drawing Class

(d) ResNet50 - Five Classes

(e) ResNet50 - No Engraving Class

(f) ResNet50 - No Drawing Class

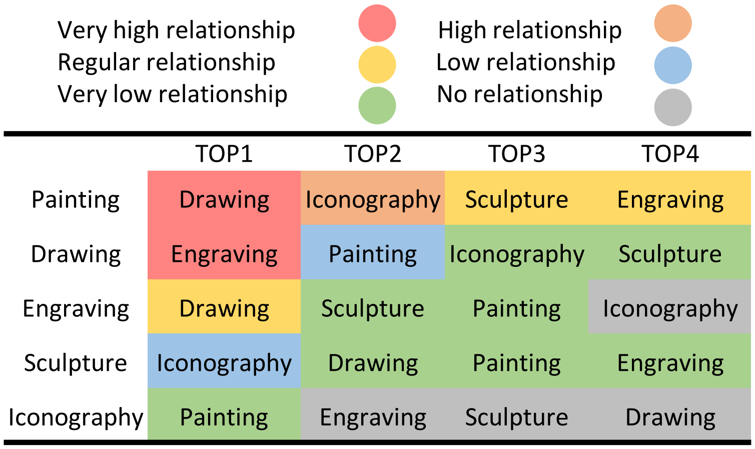
(g) Xception - Five Classes

(h) Xception - No Engraving Class

(i) Xception - No Drawing Class

Fig. 6. Confusion matrix for the Art Media Dataset (Test)

As illustrated, the Iconography class has a high classification performance by the VGG16 and Xception model (177 and 176 images correctly classified). Also, the Xception model improves classification performance with respect to the Sculpture class (162 images correctly classified). In both cases (Iconography and Sculpture) with a classification performance above 90%. Some categories share similarities in color, composition, and texture. Therefore, misclassification errors in the three CNN models, such as the Drawing and Engraving class, are common. On the other hand, the Painting class shows a classification rate of about 95 images in the three CNN models.



Very high relationship 🔴 High relationship 🟠
Regular relationship 🟡 Low relationship 🔵
Very low relationship 🟢 No relationship ⚪

	TOP1	TOP2	TOP3	TOP4
Painting	🔴 Drawing	🟠 Iconography	🟡 Sculpture	🟡 Engraving
Drawing	🔴 Engraving	🔵 Painting	🟢 Iconography	🟢 Sculpture
Engraving	🟡 Drawing	🟢 Sculpture	🟢 Painting	⚪ Iconography
Sculpture	🔵 Iconography	🟢 Drawing	🟢 Painting	🟢 Engraving
Iconography	🟢 Painting	⚪ Engraving	⚪ Sculpture	⚪ Drawing

Fig. 7. Summary of the class relationship effect in the CNN models using ArtMD

This means that the class is highly connected with the other classes and that it is difficult for CNN to predict which category it belongs to.

4.2 Classes Relationship Effects in the CNN Models

The relationship between classes refers to the similarity between the characteristics of each class, which can confuse CNN models [13]. In addition, errors in the confusion matrix can occur for various reasons, such as the quality and quantity of training data, the complexity of the classification problem, or the suitability of the learning algorithm used.

Therefore, it is essential to analyze further the nature of the errors and the dataset's characteristics to understand why the three CNN models are making errors and to determine if there is a real relationship between classes or if they are due to other causes. To get an idea of which class (Drawing or Engraving) has fewer characteristics in common, it is proposed to modify the dataset to only four classes.

This involves modifying the dense classifier stage setup of the three models (VGG16, ResNet50, and Xception): Dense(128) + DO(0.3) + Dense(64) + DO(0.2) + Dense(4). In the first additional study (setup 4), the Engraving class was removed, increasing the accuracy of the VGG16, ResNet50, and Xception models, reaching a top accuracy of 85% (ResNet50).

In the second study (setup 5), the Drawing class was removed, and a similar behavior was obtained with a maximum accuracy of 86% (Xception). It should be noted that this increase in accuracy was mainly observed in the test set, while in the training and validation sets, top accuracy exceeded 90%, as detailed in Table 3.

The confusion matrices shown in Figures 6b-c, 6e-6f, and 6h-6i reveal that three of the four classes (Drawing or Engraving, Iconography, and Sculpture) have a classification performance above 90% in the ResNet50 and Xception models in setup 4 and setup 5.

Furthermore, it is noted that in all three CNN models, the Painting class is highly related to the other categories, as they share characteristics of style, period, and techniques. This suggests that the main challenge lies in the complexity of the field of study, particularly in the Drawing and Engraving classes and the Painting class.

The summary of the three CNN models yields the following Fig. 7 In which we observe that the Drawing class presents the most problems for the classification task, with two (Painting and Engraving) of the four remaining classes. The Engraving class shows a very high relationship with the Drawing class. As for the Sculpture class, it has a shallow relationship with the Iconography class. The class with minor problems is the Iconography class, achieving almost null relationships.

The color selection was made based on the miss-classification in the three CNN models: $(125 <$ Very high$)$, $(100 <$ High $< 125)$, $(75 <$ Regular $< 100)$, $(50 <$ Low $< 75)$, $(15 <$ Very low $< 50)$, and (No relationship < 15).

In the setup and implementation of the network, it was decided to use a function of the Keras library, preprocess_input, which allows processing the images with the same characteristics as the CNN pre-trained with the ImageNet database. The function is only applied to the ResNet50 architecture due to its low performance.

5 Conclusion and Future Work

This paper proposes an evaluation and performance analysis of three different CNNs applied to Art Media Classification (AMC) in order to answer the question of what challenges arise in AMC using CNNs with transfer learning. The features previously obtained in training the CNNs allow improving the accuracy of each learning model, without the need to start from scratch.

Given the need to evaluate the learning model, the Art Media Dataset (ArtMD) was introduced. The dataset includes the art classes: Drawing, Engraving, Iconography, Painting, and Sculpture. Initially, the VGG16 model obtained the best accuracy with 75%, but when analyzing that the main challenge lies in the dataset and that the CNNs have a difficult field of study, a new configuration is proposed.

Instead of using five classes, it was decided to evaluate only four (Drawing or Engraving, Iconography, Painting, and Sculpture). Therefore, the three proposed models now obtain a top accuracy of 86%. These experiments allow us to analyze miss-classification and discuss the relationship effects in the three CNN models to understand the artwork's composition.

The results show that all the tested CNNs present a high relationship in the classification of Painting due to characteristics of style, period, etc., followed by the relationship between classes of Drawing and Engraving due to the similarities of both classes. Separately, both classes are unrelated and have a classification performance above 90%.

In the case of Iconography and Sculpture (with low or no relationship), it can be inferred that any model will be able to perform a correct classification. In our experimental study, we applied Data Augmentation, DropOut, and Batch Normalization to the dataset to mitigate the overfitting of CNNs.

As future work, we will design a classification system based on the results obtained in this research. To achieve this, a more detailed analysis of different styles of artwork will be carried out to extract additional information that reduces the class relationship effect.

Furthermore, we propose to use wavelet analysis as a preprocessing module to obtain spectral information and improve the accuracy of the proposed CNN architectures. Finally, the results can be used to enhance the design of image classification systems applied in other areas, such as medical, surveillance, aerial robotics, and automation.

Acknowledgments

This work was founded by CONACYT through the grant “Convocatoria de Ciencia Básica y/o Ciencia de Frontera 2022”, project ID 320036.

References

1. **Abidin, D. (2021).** The effect of derived features on art genre classification with machine learning. Sakarya University Journal of Science, Vol. 25, No. 6, pp. 1275–1286. DOI: 10.16984/saufenbilder.904964.
2. **Berrouane, N., Benyettou, M., Ibtissam, B. (2022).** Deep learning and feature extraction for COVID-19 diagnosis. Computación y Sistemas, Vol. 26, No. 2, pp. 909–920. DOI: 10.13053/cys-26-2-4268.
3. **Chollet, F. (2016).** Xception: Deep learning with depthwise separable convolutions. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1800–1807. DOI: 10.1109/cvpr.2017.195.

4. **Chollet, F. (2021).** Deep learning with Python. Simon and Schuster.

5. **Chu, W. T., Wu, Y. L. (2016).** Deep correlation features for image style classification. Proceedings of the 24th ACM international conference on Multimedia, pp. 402–406. DOI: 10.1145/2964284.2967251.

6. **Chu, W. T., Wu, Y. L. (2018).** Image style classification based on learnt deep correlation features. IEEE Transactions on Multimedia, Vol. 20, No. 9, pp. 2491–2502. DOI: 10.1109/ TMM.2018.2801718.

7. **Fortuna-Cervantes, J. M., Ramírez-Torres, M. T., Mejía-Carlos, M., Martínez-Carranza, J., Murguía-Ibarra, J. S. (2021).** Texture classification for object detection in aerial navigation using transfer learning and wavelet-based features. 12th International Micro Air Vehicle Conference, pp. 210–215.

8. **Fortuna-Cervantes, J. M., Soubervielle-Montalvo, C., Perez-Cham, O. E., Peña-Gallardo, R., Puente, C. (2023).** Experimental study of the performance of convolutional neural networks applied in art media classification. Mexican Conference on Pattern Recognition, pp. 169–178.

9. **Gao, J., Zhou, H., Zhang, Y. (2020).** The performance of two CNN methods in artworks aesthetic feature recognition. Proceedings of the 12th International Conference on Machine Learning and Computing, pp. 289–296. DOI: 10.1145/3383972.3383974.

10. **He, K., Zhang, X., Ren, S., Sun, J. (2016).** Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778. DOI: 10. 1109/cvpr.2016.90.

11. **Ibarra-Vazquez, G., Olague, G., Chan-Ley, M., Puente, C., Soubervielle-Montalvo, C. (2022).** Brain programming is immune to adversarial attacks: Towards accurate and robust image classification using symbolic learning. Swarm and Evolutionary Computation, Vol. 71, pp. 101059. DOI: 10.1016/j.swevo.2022.101059.

12. **Ibarra-Vazquez, G., Olague, G., Puente, C., Chan-Ley, M., Soubervielle-Montalvo, C. (2021).** Automated design of accurate and robust image classifiers with brain programming. Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 1385–1393. DOI: 10.1145/ 3449726.3463179.

13. **Jiang, Y. G., Wu, Z., Wang, J., Xue, X., Chang, S. F. (2018).** Exploiting feature and class relationships in video categorization with regularized deep neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 40, No. 2, pp. 352–364. DOI: 10.1109/TPAMI.2017.2670560.

14. **Kandel, I., Castelli, M. (2020).** Transfer learning with convolutional neural networks for diabetic retinopathy image classification. A review. Applied Sciences, Vol. 10, No. 6, pp. 2021. DOI: 10.3390/app10062021.

15. **Kovalev, V. Y., Shishkin, A. G. (2020).** Painting style classification using deep neural networks. IEEE 3rd International Conference on Computer and Communication Engineering Technology, pp. 334–337. DOI: 10.1109/ ccet50901.2020.9213161.

16. **Lombardi, T. E. (2005).** The classification of style in fine-art painting. ETD Collection for Pace University, pp. 1–158.

17. **Lugo-Sánchez, O. E., Sossa, H., Zamora, E. (2020).** Reconocimiento robusto de lugares mediante redes neuronales convolucionales. Computación y Sistemas, Vol. 24, No. 4, pp. 1589–1605. DOI: 10.13053/cys-24-4-3340.

18. **Masilamani, G. K., Valli, R. (2021).** Art classification with pytorch using transfer learning. International Conference on System, Computation, Automation and Networking, pp. 1–5. DOI: 10.1109/icscan53069.2021. 9526457.

19. **Mikołajczyk, A., Grochowski, M. (2018).** Data augmentation for improving deep learning in image classification problem. International Interdisciplinary

PhD Workshop, pp. 117–122. DOI: 10.1109/iiphdw.2018.8388338.

20. **Olague, G., Ibarra-Vázquez, G., Chan-Ley, M., Puente, C., Soubervielle-Montalvo, C., Martinez, A. (2020).** A deep genetic programming based methodology for art media classification robust to adversarial perturbations. Proceedings of the 15th International Symposium on Visual Computing. Advances in Visual Computing, pp. 68–79. DOI: 10.1007/978-3-030-64556-4_6.

21. **Rodriguez, C. S., Lech, M., Pirogova, E. (2018).** Classification of style in fine-art paintings using transfer learning and weighted image patches. 12th International Conference on Signal Processing and Communication Systems, pp. 1–7. DOI: 10.1109/icspcs.2018.8631731.

22. **Rojas-Pérez, L. O., Martínez-Carranza, J. (2020).** Autonomous drone racing with an opponent: A first approach. Computación y Sistemas, Vol. 24, No. 3, pp. 1271–1279. DOI: 10.13053/cys-24-3-3486.

23. **Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986).** Learning representations by back-propagating errors. Nature, Vol. 323, No. 6088, pp. 533–536. DOI: 10.1038/323533a0.

24. **Simonyan, K., Zisserman, A. (2014).** Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations, pp. 1–14. DOI: 10.48550/ARXIV.1409.1556.

25. **Srivastava, N. (2013).** Improving neural networks with dropout. University of Toronto, Vol. 182, No. 566, pp. 7.

Article received on 27/08/2023; accepted on 13/10/2023.
Corresponding author is Carlos Soubervielle-Montalvo.*

ISSN 2007-9737

Lightweight CNN for Detecting Microcalcifications Clusters in Digital Mammograms

Ricardo Salvador Luna-Lozoya[1], Humberto de Jesús Ochoa-Domínguez*,[1],
Juan Humberto Sossa-Azuela[2], Vianey Guadalupe Cruz-Sánchez[1],
Osslan Osiris Vergara-Villegas[1]

[1] Universidad Autónoma de Ciudad Juárez,
Ciudad Juárez,
Mexico

[2] Instituto Politécnico Nacional,
Ciudad de México,
Mexico

al216618@alumnos.uacj.mx, hsossa@cic.ipn.mx
{hochoa, vianey.cruz, overgara}@uacj.mx

Abstract. Digital mammogram plays a key role in breast cancer screening, with microcalcifications being an important indicator of an early stage. However, these injuries are difficult to detect. In this paper, we propose a lightweight Convolutional Neural Network (CNN) for detecting microcalcifications clusters in digital mammograms. The architecture comprises two convolutional layers with 6 and 16 filters of 9×9, respectively at a full scale, a global pooling layer that eliminates the flattening and dense layers, and a sigmoid function as the output layer for binary classification. To train the model, we utilize the public INbreast database of digital mammograms with labeled microcalcification clusters. We used data augmentation techniques to artificially increase the training set. Furthermore, we present a case study that encompasses the utilization of a software application. After training, the resulting model yielded an accuracy of 99.3% with only 8,301 parameters. This represents a considerable parameter reduction as compared to the 67,797,505 used in MobileNetV2 with 99.8 % accuracy.

Keywords. Microcalcifications clusters detection, shallow convolutional neural network, deep learning.

1 Introduction

Breast cancer is a significant public health challenge, with the highest incidence among women [14].

The detection of small calcium deposits from 0.1 mm to 1 mm in length called Microcalcifications (MCs) [4], plays a vital role in identifying early breast cancer, leading to a 99% survival rate at 5 years or more [3]. Microcalcifications clusters (MCCs) are conformed by at least three MCs per cm^2. These lesions are present in up to 50% of the confirmed cancer cases [29, 36, 37].

The detection of MCs is a complex process due to their size, shape, and distribution [11]. Among the medical imaging techniques, mammography is the most widely used to detect MCCs [4, 6]. The use of Artificial Intelligence (AI) techniques is safe and reliable [9] and can be used to detect the initial signs of diseases [12].

Among these techniques, the Deep Learning (DL) models [21] have achieved high degrees of accuracy and Convolutional Neural Networks (CNNs) are being studied in the field of MCCs detection [4]. As CNN architectures evolve, they have become more complex and deeper.

Hence, the complexity has posed challenges, particularly in medical entities where resource-intensive models for diagnosis can be impractical. A solution is to develop lighter CNN architectures where training and/or retraining times can be minimized, making the network more accessible and efficient, all while requiring

doi: 10.13053/CyS-28-1-4892

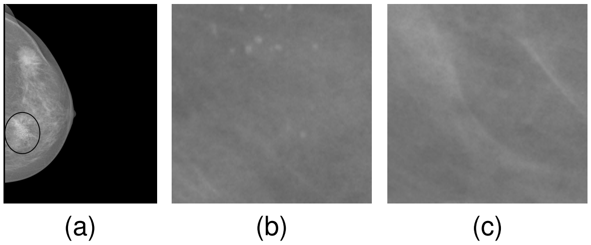


Fig. 1. Digital mammogram showing (a) a circled MCCs, (b) a patch of tissue with MCCs, and (c) with normal tissue

Table 1. Most representative architectures yielded by the Hyperband search algorithm

CL	Filter size	Number of filters	MPL	Parameters	Accuracy
2	5 × 5	CL1: 6 CL2: 16	2	2,589	99.1%
2	5 × 5	CL1: 6 CL2: 16	0	2,589	99.1%
2	5 × 5	CL1: 4 CL2: 10	0	1,125	98.8%
1	5 × 5	CL1: 16	0	433	97.8%
6	5 × 5	CL1 - CL6: 4	0	2,129	99%
2	3 × 3	CL1 - CL2: 16	0	957	98%
2	7 × 7	CL1: 6 CL2: 16	0	5,037	99.1%
2	11 × 11	CL1: 6 CL2: 16	0	12,381	99.3%
2	**9 × 9**	**CL1: 6** **CL2: 16**	**0**	**8,301**	**99.3%**

fewer computational resources. In light of the challenges exposed, we present a novel approach incorporating a lightweight and shallow CNN for detecting the presence or absence of MCCs in digital mammograms.

This research builds upon the foundations laid in our prior work [19], representing a continuation and refinement of our previous findings. The paper makes significant contributions, which can be outlined as follows:

- A lightweight CNN specifically designed for the detection of MCCs in digital mammograms using a reduced number of parameters. The network's efficiency is attributed to its notably reduced number of parameters, making it an attractive and practical solution for medical entities seeking efficient MCCs detection.
- A case study of the proposed model. We are primarily concerned with the theoretical and practical applications of our model. Therefore, we developed a software application to detect MCCs. The application is being evaluated by expert radiologists.

The article is organized as follows: Section 2 reviews the related work. Section 3 outlines materials and methods. Section 4 presents the results. Section 5 discusses outcomes. Lastly, Section 6 offers conclusions.

2 Related Work

Efforts to improve accuracy are the main driver behind recent trends in the detection of MCCs. Here, we briefly review the works we consider the most significant because they put our work into context. Gómez et al. [10] proposed a methodology for preprocessing 832 digital mammograms specifically from the mini-MIAS [31] and the UTP [7] databases.

This CNN model comprises seven Convolutional Layers (CL) with a kernel size of 3×3. Following each CL, a Max Pooling Layer (MPL) and a layer of Rectified Linear Unit (ReLU) activation functions were incorporated. The CNN achieved a testing accuracy of 95.83%.

Rehman et al. [25] proposed a Fully Connected Deep-Separable CNN (FC-DSCNN) for detecting and classifying MCCs as benign or malignant. The system involves four steps including image processing, grayscale transformation, suspicious region segmentation, and MCCs classification.

They tested the system on 6,453 mammograms from the public DDSM [27] dataset and from the private Punjab Institue of Nuclear Medicine (PINUM) database, achieving results with 99% sensitivity, 82% specificity, 89% precision, and 82% recall.

Hsieh et al. [11] implemented a VGG-16 network to detect MCCs in 1586 mammograms from the Medical Imaging Department of the Chung-Shan Medical University. They used a Mask R-CNN for MCC segmentation and InceptionV3 for MCC classification (benign or malignant).

Table 2. Tuned hyperparameters and optimal values via Hyperband method

Hyperparameter		Evaluated values	Best value
CL1	Number of filters	4, 6, 8, 10, 12	6
	Filter size ($n \times n$)	n = 3, 5, 7, 9, 11	9
CL2	Number of filters	16, 20, 24, 28, 32	16
	Filter size ($n \times n$)	n = 3, 5, 7, 9, 11	9
	Batch size	16, 32, 64, 128	64
Learning rate		0.01, 0.001, 0.0001	0.001

The method achieved a 93% accuracy for classification and detection, 95% for MCs labeling, and 91% for MCC classification. The overall precision, specificity, and sensitivity were 87%, 89%, and 90%, respectively.

Valvano et al. [35] developed two CNNs for the detection and segmentation of Regions of Interest (ROIs) or patches containing MCs. They employed a private database consisting of 283 mammograms with a resolution of 0.05 mm.

Each patch was labeled positive if it contained MCs and negative if it did not. The presence or absence of MCs in each patch was then detected using a CNN. Both CNNs were constructed with six CLs. They achieved an accuracy of 98.22% for the detector and 97.47% for the segmenter.

The most intuitive idea to improve accuracy is to use deeper CNNs. This requires a lot of time to train and use it. There is a clear sacrifice of computational complexity and, in some cases, an incipient gain in precision. Recently, Luna et al. [19] showed that, for MCCs detection, very deep CNN performed similarly to the shallow ones.

They compared different CNNs, in the state-of-the-art, used for classification purposes and found that the networks yielded accuracies between 99.71% and 99.84%. Therefore, for this type of lesion, shallow networks with a reduced number of parameters can be designed to be accommodated in little hardware.

To the best of our knowledge, among these networks, only the VGG-16 architecture has been employed for MCCs detection [11]. Nevertheless, the authors did not report any comparison with other DL networks or structures, lacking sustain the use of this network for this type of lesions.

3 Materials and Methods

In this section, we present an overview of the materials utilized and the methods adopted to investigate MCCs detection in digital mammograms using CNNs.

3.1 Data

We used the INbreast database [22] for training, validating, and testing the model. It comprises 410 grayscale digital mammograms of 2,560 × 3,328 and 3,328 × 4,084 pixels, each pixel is 70 microns. The mammograms are labeled with various types of lesions. In this study, we selected exclusively the ten mammograms labeled as MCC in the database.

3.1.1 Data Preparation

We converted the Digital Imaging and Communication In Medicine (DICOM) images database into Portable Network Graphic (PNG) format. The labeling and coordinates of the breast lesions were available in separate Extensible Markup Language (XML) files and independently associated with the images.

In order to accurately mark the MCCs on the digital mammograms, we developed a custom software, in Python 3.0, to read and extract the MCCs coordinates from XML files for precise localization and annotation of these lesions within mammograms.

3.1.2 Patch Extraction

The proposed model processes mammograms in patches of 1 cm^2 equivalent to 144 × 144 pixels as those shown in Figs. 1 (b) and (c). We developed another dedicated computer program in Python 3.0 to extract annotated patches from the mammograms.

In total, 1,576 patches with MCCs and 1,692 patches without lesions were selected. The initial CNN training sessions were conducted using the dataset [22] as it is. The results were not as expected in all the tested architectures [19]. We asked an expert radiologist to clean our database. She noticed that some patches, labeled as MCCs,

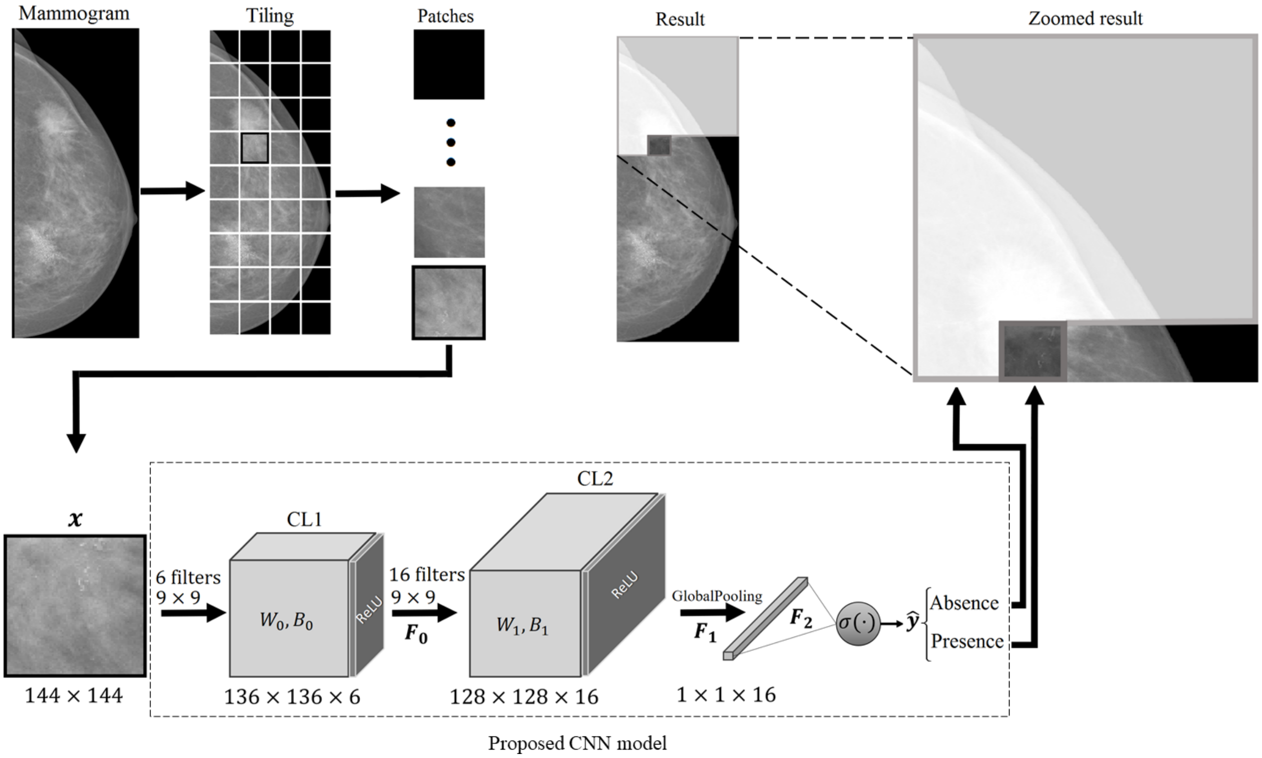


Fig. 2. Case study of the proposed CNN model. If the patch is classified as absence of MCCs, it becomes lighter, and if it is classified as presence the patch darkens

did not contain MCCs, and some unlabeled ones did contain them. Now, with the cleaned database the results exceeded 98% on accuracy [19].

3.1.3 Data Augmentation

The availability of mammograms labeled with MCCs in the INbreast database is limited. Since DL models depend on the quantity and contextual meaning of training data, we artificially increased the number of examples in the database by applying reflection, 180° turn, reflection and 180° turn, and 90° turn, to each patch to obtain 6,304 extra patches with MCCs and 6,768 extra patches without MCCs.

Notice that only geometric transformations were applied to preserve the original features. Consequently, we ended up with a total of 7,880 patches with MCCs and 8,460 patches without MCCs, resulting in a comprehensive dataset of 16,340 patches.

3.1.4 The Datasets

When training a DL model, it is very important to have a dataset with almost the same number of samples in each class. This prevents the model from becoming biased toward one class.

Hence, 7,880 patches with MCCs and 7,880 patches with normal tissue from the database were used. By Pareto's Principle [2], from the dataset we assigned 80% of the data for both training and validation, while the remaining 20% for testing purposes.

More specifically, we utilized 64% (10,088 patches) for training and 16% (2,520 patches) for validation, and for testing, we reserved the remaining 20% (3,152 patches).

To ensure consistency, all patches were normalized by dividing their pixel values by 255. Notice that the data augmentation process was applied to each dataset individually to avoid overfitting.

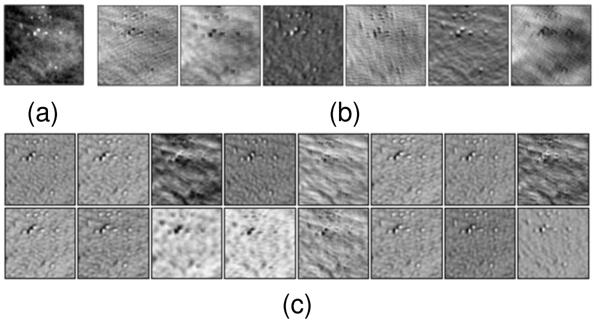


Fig. 3. Patch x (a) with MCCs and feature maps (b) F_0 and (c) F_1 (see Fig. 2)

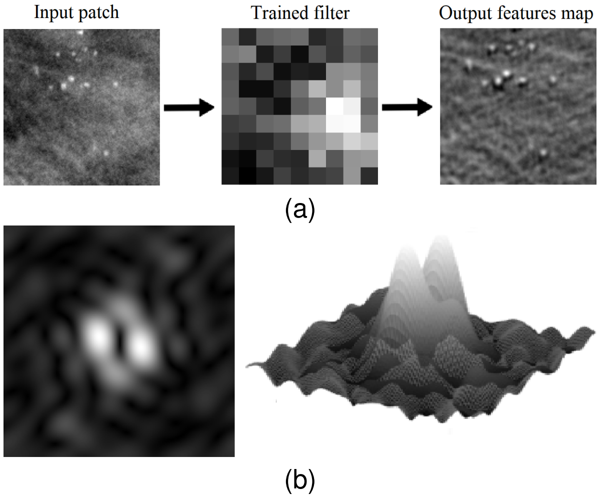


Fig. 4. Filtering process of (a) the third trained filter of $\mathrm{CL}\,1$ and (b) magnitude response of the filter

3.1.5 The Proposed Architecture

The proposed architecture was conceived on the premise that biological models of MCs and their surrounding tissue exhibit a reduced number of features [38]. The MC is modeled as a sum of Gaussian functions [38] with limited frequency support (from 0.1 to 1 millimeter) [4].

Therefore, we concluded that it is unnecessary to use a very deep CNN to classify MCCs. This was demonstrated in [19] where CNN models like LeNet-5 [16] with only 5 layers or AlexNet [15] with 8 layers can effectively detect MCCs with the same accuracy.

Besides, these two networks were specifically designed to classify numbers and natural images with a large set of features. Furthermore, in the literature, the current networks are pre-trained on natural images [20]. Hence, it is essential to capture a greater number of low- and high-level features. In the reported works on MCCs detection and classification [24, 18, 23, 26, 28], there is a notable absence of experiments.

The authors typically bring the knowledge of a pre-trained CNN to their own domain by retraining it to observe the prediction or classification results regardless of the depth of the network. However, models of MCCs proposed from biological analyses [38] report that these lesions have a limited number of features, often described as a sum of Gaussian functions.

Therefore, we decided to experiment with one convolutional and one MPL first. Then, we increased the number of layers and noticed that, after two or more layers, the performance was similar. Afterward, we experimented by suppressing the Pooling Layers (PLs) and noticed an improved performance.

Finally, we replaced the Flattening and FCLs with a Global Max Pooling Layer (GMPL) and noticed that the performance was not compromised. However, the number of parameters drastically decreased. Finally, for training, Hyperband search [17] was used to tune the hyperparameters. Table 1 shows the most representative combinations yielded by the algorithm. We propose the lightweight CNN depicted in the case study of Fig. 2.

Each model was trained using TensorFlow framework 2.0 [1] in Google Colaboratory [5]. The platform automatically adjusted the computer resources as needed. For instance, in the latest session, the model accessed a 108GB hard drive, an Intel Xeon (R) CPU @ 2.20GHz processor, and 13GB of memory.

Notice that, we will call the architecture to the structure of the CNN (number of layers, how they are connected, and the type of activation function) and the model to the function that the CNN is approximating after training. The architecture consists of two CLs with a ReLU layer at the output of each, followed by a GMPL.

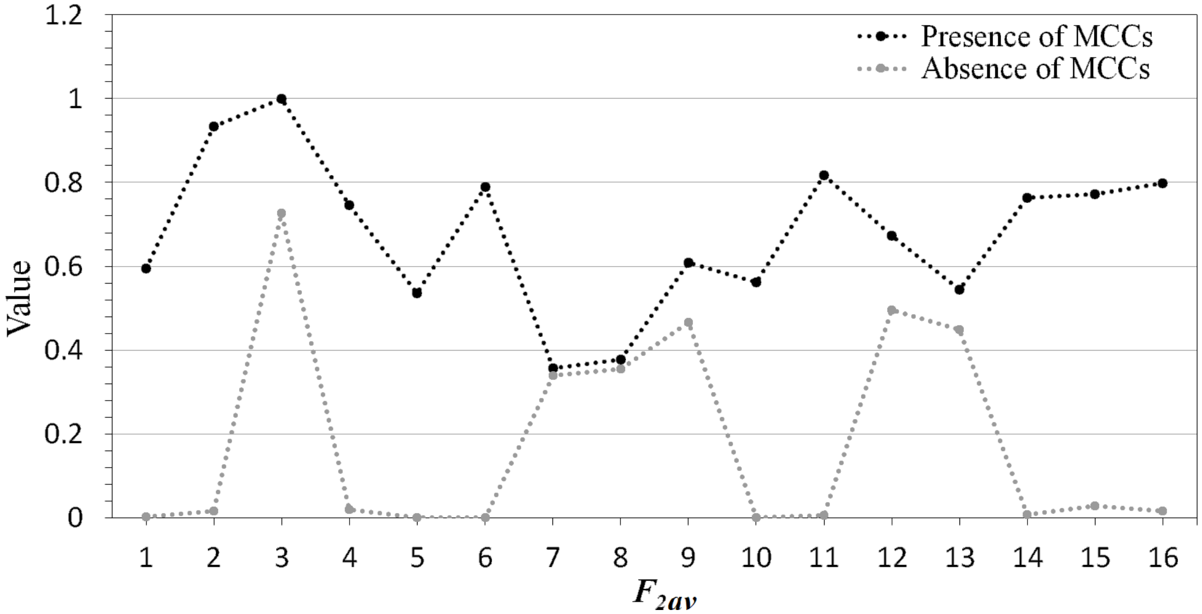


Fig. 5. Plots of the element-wise average of the components of the F_2 classified as presence (black dotted graph) and absence of MCCs (gray dotted graph)

The output layer consists of a sigmoid function. The two CLs are used at full scale, that is, no PLs are inserted to reduce dimensionality. The Binary Cross Entropy (BCE) cost function used is shown in Eq. (1):

$$L(y, \hat{y}) = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right], \quad (1)$$

where $-\frac{1}{m}\sum_{i=1}^{m}$ is the average loss of the whole batch, m denotes the training set size, y_i is the label, taking binary values 0 or 1 and $\hat{y}_i$ is the predicted value. $-1/m$ ensures that the cost is always greater or equal to 0.

3.1.6 Hyperparameter Tuning

Searching for optimal hyperparameters was a challenge because of the limited computational resources. Hence, we employed the Hyperband search method [17] for hyperparameters tunning by exploring the number and filter sizes, batch size, and learning rate within a relatively narrow range of options.

We used dropout regularization with a permanency of 80% throughout the training process and Adaptive Moment Estimation (ADAM) regularization. Table 2 shows the values of the hyperparameters evaluated by the method along with the best results.

3.2 The Proposed Model

From the previous section, the resulting CNN model consists of two CLs, each followed by a ReLU layer. The first layer has 6 filters of size 9×9, denoted by W_0 with biases B_0. The output is represented as:

$$F_0 = \max(0,\ W_0 \cdot x + B_0), \quad (2)$$

where max$(0,\ z)$ denotes the largest value between zero and z. Similarly, the second layer comprises 16 filters of size 9× 9, denoted by W_1 with biases B_1. The output can be modeled as:

$$F_1 = \max(0, W_1 \cdot x + B_1). \quad (3)$$

The resulting 16 feature maps are sent to a GMPL to obtain the maximum value of each map

Table 3. Performance comparison of the proposed CNN versus MobileNetV2 and LeNet-5

Architecture	Accuracy	Parameters
MobileNetV2	**99.8%**	67,797,505
LeNet-5	99.3%	2,233,365
Proposed	99.3%	**8,301**

to yield a vector of 16 features represented as $F_2 = \max(F_1)$. The F_2 vector is sent to the output layer where a predicted value between 0 and 1 is assigned according to the vector values. The proposed CNN model is shown in Fig. 2.

3.2.1 Software Application

We developed a web-based software application to test the model's ability to analyze digital mammograms in real time with the domain used to train the network (INbreast database [22]). The user interface allows to import digital mammograms in a PNG format. The software extracts progressively 1 cm^2 patches from the mammogram scanning it from top to bottom and from left to right.

The patch undergoes analysis by the proposed model that yields results between 0 and 1. A near 0 result indicates the absence of MCCs, prompting the application to display the patch in a light gray color. Conversely, a result close to 1 indicates the presence of MCCs, displaying the patch as it is. The application can be configured to display the patch with a color depending on the class it belongs to.

Additionally, counters for each class are maintained to display the number of patches found with and without MCCs during the scanning. The application is hosted on a local server equipped with a 100GB hard drive, an Intel Xeon (R) CPU @ 2.20GHz processor, and 8GB of memory.

Debian [30] serves as the operating system, Apache 2 [32] as the HTTP server, and PHP 8 [34] as the backend. PHP handles tasks such as uploading mammograms to the server, removing the black background, and splitting images into patches for analysis.

Angular v14 [8] is used as the frontend, fetching patches from the backend and utilizing a web service to implement the proposed model. The application's aesthetic is styled using the Bootstrap library [33].

3.2.2 Case Study

Fig. 2 shows a case study implemented for the proposed model. The input mammogram is split into patches of 144 $\times$144 pixels. The coordinates of each patch are stored and the patch $\boldsymbol{x}$ is sent to the trained CNN model where it undergoes classification. The classified patch is seamlessly integrated back into the mammogram at its original location with a different grayscale that depends on the output classification result $\hat{\boldsymbol{y}}$.

The result is shown in a displayed mammogram with detected normal tissue in light gray and injured tissue in dark gray. The transformation can be inverted anytime to show the original image. This case study was implemented in a software application that is under test by the Centro de Imagen e Investigación (Medimagen) of Chihuahua, México [13].

4 Results

This section exposes the results of the proposed CNN. All the models were trained with 100 epochs. Fig. 3 shows (a) one patch with MCCs that undergoes prediction, (b) the six feature maps F_0 yielded by the first CL, and (c) the sixteen feature maps F_1 at the output of the second CL.

Fig. 4(a) shows the convolution process of the input patch with MCs with the third trained filter of the $\mathrm{CL}\,1$. The brightest pixels represent the parts of the spectrum with the highest magnitude. Fig. 4(b) shows the magnitude response of this filter. Observe the limited frequency support.

Fig. 5 shows two plots of the element-wise average output of the sixteen components of the vector F_2, the upper graph, represented by the dotted black line, is the average of the prediction of one hundred patches classified as presence. The light gray dotted line is the element-wise average of the prediction of one hundred patches classified as absent.

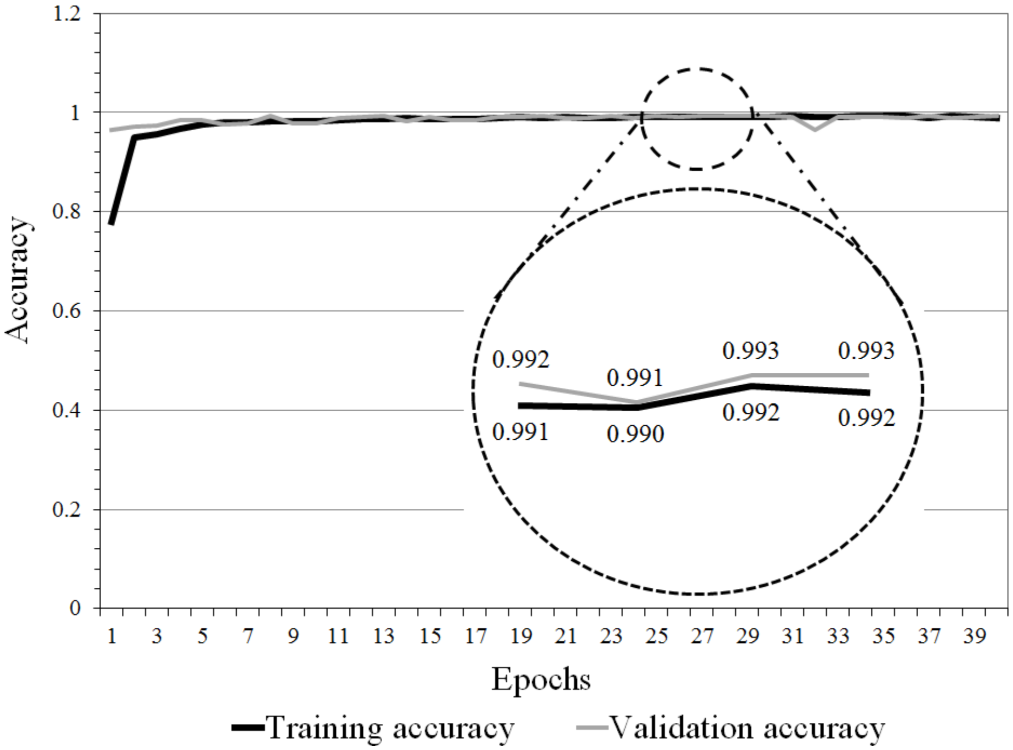


Fig. 6. Proposed model performance accuracy for (black line) training and (gray line) validation

In other words:

$$F_{2\,\mathrm{av}} = (F_{2,1} \cdot + F_{2,2} \cdot + \cdots \cdot + F_{2,100}) \cdot /100, \quad (4)$$

where $\cdot +$ and $\cdot /$ are the element-wise sum and division operations and $F_{2,i}$ the ith vector after each prediction. Table 3 presents a comparison of both accuracy and the number of trainable parameters among the proposed model and the MobileNetV2 and LeNet-5 networks.

In [19], MobileNetV2 demonstrated the highest precision in detecting MCCs, while the LeNet-5 network exhibited the fewest number of trainable parameters. Observe that both, the MobileNetV2 and the LeNet-5, were trained from scratch using the same datasets as in the proposed model was trained. Fig. 6 shows the accuracy performance throughout the configured epochs for both the training and the validation processes.

It is important to mention that an expert radiologist corroborated the testing results by using the software application developed.

5 Discussion

In Fig. 3 (b) we notice that, in the first, second, fourth, and sixth maps (from left to right), the MCs locations appear in a pitch black with a rounded feature. Smaller MCs locations are more noticeable in the first and second maps. However, larger MCs locations are detected on the second, fourth, and sixth maps.

These maps separate the MCs leaving only the information of the surrounding tissue. The third and fourth maps highlight the features of the MCs being more prominent on the third map. Besides, the surrounding tissue is attenuated leaving only the MCs features.

Furthermore, Fig. 3 (c) shows a higher level of features. However, we can still see that, from left to right and top to bottom, the third, fifth, eighth, eleventh, twelfth, and thirteenth maps carry the tissue features, and the remaining maps are the MCs features.

The proposed CNN identifies and separates in the feature maps the various characteristics in a patch. To save parameters, a GMPL is added to the output of the second layer. Fig. 5 shows two plots $F_{2\,\mathrm{av}}$ corresponding to the averaged elements of each output F_2 as explained in the previous section.

Notice how the two plots do not overlap each other, this means that on average, there is no overfitting in the network. It is important to observe that ten feature maps yield results close to zero when MCCs are absent and results greater than 0.5 when MCCs are present. Here, the third feature map yields a result greater than 0.5 when MCCs are absent.

However, the same map yields a value close to one when MCCs are present. Additionally, feature maps 7 and 8 give results close to the overlap. Nevertheless, on average, the results are separated. Fig. 6 shows that training and validation performance are not separated from each other.

In fact, they maintain the same tendency. This suggests that there is no overfitting. Table 3 shows that our network achieves comparable accuracy to LeNet-5 CNN with the notable advantage of being 268 times smaller. Moreover, observe that the MobileNetV2 CNN yields an accuracy that is only 0.5% higher than the proposed network. However, the proposed network is 8,167 times smaller.

The MCs range from 0.1 to 1 mm [4] and the scanner used to collect the INbreast database has a resolution of 70 microns per pixel in both directions (horizontal and vertical) [22].

Therefore, an MC varies in size from approximately 2 to 14 pixels which indicates a limited frequency support (from $|1/14|$ to $|1/2|$) as shown in Fig. 4(b) where the bandpass region is delimited by the size of the MC, which clearly indicates that this filter is trained to capture the support.

Moreover, within this region of MCs support, there are other signals that are not MCs as shown in the output features map of Fig. 4(a). Nevertheless, these extra features will be discriminated by the $\mathrm{CL}\,2$.

6 Conclusions

In this paper, we propose a lightweight CNN for detecting MCCs in digital mammograms. The input layer has 6 filters of size 9×9 with ReLU activation functions to have a 6-dimensional feature maps. The second layer performs a nonlinear mapping using 16 filters of size 9×9 with ReLU function.

No PL was added to reduce the dimensionality of the CLs. A GMPL is added to reduce the number of parameters and transform the last 16-dimensional feature maps into a 1D vector. For binary classification, the last layer is a sigmoid function. The resulting model comprises 8,301 parameters making it easily implementable across various frameworks. The achieved accuracy aligns with results from the LeNet-5 and the even more intricate MobileNetV2.

The application developed for our model is under test by the Centro de Imagen e Investigación (Medimagen) of Chihuahua, México. A noteworthy discovery by the expert radiologist, while using the application, was that the model can identify MCCs that initially were not labeled in the INbreast database. This is because the unmarked MCCs were challenging to observe without the support of the application, and the almost imperceptible MCCs often turn out to be malignant.

The ongoing aspect of this research involves developing a faster residual CNN with enhanced performance. Then, the proposed model in this research serves as a foundation for the new CNN. In addition, other types of layers such as the depthwise separable convolutional layers are also being tested. Because of the simplicity of our CNN, we are developing a framework to include explainability in the model. In addition, we are collecting a database of Mexican mammograms, labeled by expert radiologists with several types of lesions that can be used to train new models of DL to work in hospitals and clinics of the country.

Acknowledgments

We thank the UACJ for the support provided, the CONAHCYT for the scholarship granted, and the radiologist Dra. Karina Núñez for her valuable support in carrying out this work.

References

1. **Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., et al. (2015).** TensorFlow: Large-scale machine learning on heterogeneous systems. DOI: 10.48550/arXiv.1603.04467.

2. **Abdelaziz-Ismael, S. A., Mohammed, A., Hefny, H. (2020).** An enhanced deep learning approach for brain cancer mri images classification using residual networks. Artificial Intelligence in Medicine, Vol. 102, pp. 101779. DOI: 10.1016/j.artmed.2019.101779.

3. **American Cancer Society (2023).** Tasas de supervivencia del cáncer de seno.

4. **Basile, T. M. A., Fanizzi, A., Losurdo, L., Bellotti, R., Bottigli, U., Dentamaro, R., Didonna, V., Fausto, A., Massafra, R., Moschetta, M., Tamborra, P., Tangaro, S., La-Forgia, D. (2019).** Microcalcification detection in full-field digital mammograms: A fully automated computer-aided system. Physica Medica, Vol. 64, pp. 1–9. DOI: 10.1016/j.ejmp.2019.05.022.

5. **Bisong, E. (2019).** Building machine learning and deep learning models on Google Cloud Platform: A comprehensive guide for beginners. Apress Berkeley, CA. DOI: 10.1007/978-1-4842-4470-8.

6. **Cronin, K., Scott, S., Firth, A., Sung, H., Henley, S. J., Sherman, R. L., Siegel, R., Anderson, R., Kohler, B., Benard, V., Negoitia, S., Wiggins, C., Cance, W., Jemal, A. (2018).** Annual report to the nation on the status of cancer, part i: National cancer statistics. Cancer, Vol. 128, No. 24, pp. 4251–4284. DOI: 10.1002/cncr.34479.

7. **Echeverry-Correa, J. D., Orozco-Gutiérrez, A. A., Cárdenas-Peña, D. A., Marín-Mejía, S. (2023).** Recuperación de información por contenido orientada a la clasificación de grupos de microcalcificaciones en mamografías - Protocam. Universidad Tecnológica de Pereira. DOI: 10.22517/9789587225174.

8. **Google (2023).** The web development framework for building the future.

9. **Henriksen, E. L., Carlsen, J. F., Vejborg, I. M. M., Nielsen, M. B., Lauridsen, C. A. (2018).** The efficacy of using computer-aided detection (CAD) for detection of breast cancer in mammography screening: a systematic review. Acta Radiologica, Vol. 60, No. 1, pp. 13–18. DOI: 10.1177/0284185118770917.

10. **Hernández-Gómez, K. A., Echeverry-Correa, J. D., Orozco-Gutiérrez, A. A. (2021).** Automatic pectoral muscle removal and microcalcification localization in digital mammograms. Healthcare Informatics Research, Vol. 27, No. 3, pp. 222–230. DOI: 10.4258/hir.2021.27.3.222.

11. **Hsieh, Y. C., Chin, C. L., Wei, C. S., Chen, I. M., Yeh, P. Y., Tseng, R. J. (2020).** Combining VGG16, Mask R-CNN and inception V3 to identify the benign and malignant of breast microcalcification clusters. IEEE International Conference on Fuzzy Theory and Its Applications (iFUZZY), pp. 1–4. DOI: 10.1109/iFUZZY50310.2020.9297809.

12. **IBM (2023).** DREAM challenge results: Can machine learning help improve accuracy in breast cancer screening?

13. **ImagenologIA (2023).** Microcalcification clusters detection model in real time.

14. **International Agency for Research on Cancer (2023).** Breast.

15. **Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012).** Imagenet classification with deep convolutional neural networks. 26th Annual Conference on Neural Information Processing Systems. Advances in Neural Information Processing Systems, pp. 1106–1114.

16. **Lecun, Y., Bottou, L., Bengio, Y., Haffner, P. (1998).** Gradient-based learning applied

to document recognition. Proceedings of the IEEE, Vol. 86, No. 11, pp. 2278–2324. DOI: 10.1109/5.726791.

17. **Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A. (2017).** Hyperband: A novel bandit-based approach to hyperparameter optimization. The Journal of Machine Learning Research, Vol. 18, No. 1, pp. 6765–6816.

18. **Liu, H., Chen, Y., Zhang, Y., Wang, L., Luo, R., Wu, H., Wu, C., Zhang, H., Tan, W., Yin, H., Wang, D. (2021).** A deep learning model integrating mammography and clinical factors facilitates the malignancy prediction of BI-RADS 4 microcalcifications in breast cancer screening. European Radiology, Vol. 31, No. 8, pp. 5902–5912. DOI: 10.1007/s00330-020-07659-y.

19. **Luna-Lozoya, R. S., Ochoa-Domínguez, H. J., Sossa-Azuela, J. H., Cruz-Sánchez, V. G., Vergara-Villegas, O. O. (2023).** Comparison of deep learning architectures in classification of microcalcifications clusters in digital mammograms. Mexican Conference on Pattern Recognition, pp. 231–241. DOI: 10.1007/978-3-031-33783-3_22.

20. **Mahardi, Wang, I. H., Lee, K. C., Chang, S. L. (2020).** Images classification of dogs and cats using fine-tuned VGG models. IEEE Eurasia Conference on IOT, Communication and Engineering, pp. 230–233.

21. **Miotto, R., Wang, F., Wang, S., Jiang, X., Dudley, J. T. (2018).** Deep learning for healthcare: review, opportunities and challenges. Briefings in Bioinformatics, Vol. 19, No. 6, pp. 1236–1246. DOI: 10.1093/bib/bbx044.

22. **Moreira, I. C., Amaral, I., Domingues, I., Cardoso, A., Cardoso, M. J., Cardoso, J. S. (2012).** INbreast: Toward a full-field digital mammographic database. Academic Radiology, Vol. 19, No. 2, pp. 236–248. DOI: 10.1016/j.acra.2011.09.014.

23. **Mota, A. M., Clarkson, M. J., Almeida, P., Matela, N. (2022).** Automatic classification of simulated breast tomosynthesis whole images for the presence of microcalcification clusters using deep CNNs. Journal of Imaging, Vol. 8, No. 9, pp. 231. DOI: 10.3390/jimaging8090231.

24. **Rasool, E., Anwar, M. J., Shaker, B., Hashmi, M. H., Rehman, K. U., Seed, Y. (2023).** Breast microcalcification detection in digital mammograms using deep transfer learning approaches. Proceedings of the 9th International Conference on Computing and Data Engineering, pp. 58–65.

25. **Rehman, K. U., Li, J., Pei, Y., Yasin, A., Ali, S., Mahmood, T. (2021).** Computer vision-based microcalcification detection in digital mammograms using fully connected depthwise separable convolutional neural network. Sensors, Vol. 21, No. 14, pp. 4854. DOI: 10.3390/s21144854.

26. **Sabani, A., Landsmann, A., Hejduk, P., Schmidt, C., Marcon, M., Borkowski, K., Rossi, C., Ciritsis, A., Boss, A. (2022).** BI-RADS-Based classification of mammographic soft tissue opacities using a deep convolutional neural network. Diagnostics, Vol. 12, No. 7, pp. 1564. DOI: 10.3390/diagnostics12071564.

27. **Sawyer-Lee, R., Gimenez, F., Hoogi, A., Rubin, D. (2016).** Curated breast imaging subset of digital database for screening mammography (CBIS-DDSM) [data set]. The cancer imaging archive.

28. **Shiri Kahnouei, M., Giti, M., Akhaee, M. A., Ameri, A. (2022).** Microcalcification detection in mammograms using deep learning. Iranian Journal of Radiology, Vol. 19, No. 1. DOI: 10.5812/iranjradiol-120758.

29. **Sickles, E., D'Orsi, C., Bassett, L., et al. (2013).** ACR BI-RADS® mammography. In: ACR BI-RADS® atlas, breast imaging reporting and data system. American College of Radiology.

30. **Software in the Public Interest (2023).** The universal operating system.

31. **Suckling, J. (2023).** The mini-MIAS database of mammograms.

32. **The Apache Software Foundation (2023).** HTTP server project.

33. **The Bootstrap Team (2023).** The most popular HTML, CSS, and JS library in the world.

34. **The PHP Group (2023).** PHP: Hypertext preprocessor. www.php.net/.

35. **Valvano, G., Santini, G., Martini, N., Ripoli, A., Iacconi, C., Chiappino, D., Della Latta, D. (2019).** Convolutional neural networks for the segmentation of microcalcification in mammography imaging. Journal of Healthcare Engineering, Vol. 2019, pp. 1–9. DOI: 10.1155/2019/9360941.

36. **Wang, J., Nishikawa, R. M., Yang, Y. (2017).** Global detection approach for clustered microcalcifications in mammograms using a deep learning network. Journal of Medical Imaging, Vol. 4, pp. 024501. DOI: 10.1117/1.JMI.4.2.024501.

37. **Wang, J., Yang, Y. (2018).** A context-sensitive deep learning approach for microcalcification detection in mammograms. Pattern Recognition, Vol. 78, pp. 12–22. DOI: 10.1016/j.patcog.2018.01.009.

38. **Yang, Y., Yang, Y., Liu, Z., Guo, L., Li, S., Sun, X., Shao, Z., Ji, M. (2021).** Microcalcification-based tumor malignancy evaluation in fresh breast biopsies with hyperspectral stimulated raman scattering. Analytical Chemistry, Vol. 93, No. 15, pp. 6223–6231. DOI: 10.1021/acs.analchem.1c00522.

Article received on 04/07/2023; accepted on 13/10/2023.
** Corresponding author is Humberto de Jesús Ochoa-Domínguez.*

ISSN 2007-9737

Classifying Roads with Multi-Step Graph Embeddings

Mohale E. Molefe*, Jules R. Tapamo

University of KwaZulu Natal,
School of Engineering,
South Africa

molefemohale@gmail.com, tapamoj@ukzn.ac.za

Abstract. Machine learning-based road-type classification is pivotal in intelligent road network systems, where accurate network modelling is crucial. Graph embedding methods have emerged as the leading paradigm for capturing the intricate relationships within road networks. However, their effectiveness hinges on the quality of input features. This paper introduces a novel two-stage graph embedding approach used to classify road-type. The first stage employs Deep Autoencoders to produce compact representation of road segments. This compactified representation is then used, in the second stage, by graph embedding methods to generate an embedded vectors, leveraging the features of neighbouring segments. Results achieved, with experiments on realistic city road network datasets, show that the proposed method outperforms existing approaches with respect to classification accuracy.

Keywords. Road type classification, road networks intelligent systems, graph embedding methods, deep autoencoder.

1 Introduction

Cities worldwide face growing traffic issues, such as congestion, accidents, and rising fuel costs. These problems are caused by increased population, vehicles in traffics, and the overall number of people using the roads.

Designing and developing smart cities is essential for better managing and reducing these traffic problems [11]. Smart city initiatives leverage information and communication infrastructure (ICT) to optimize urban living by tackling historical urban challenges through data-driven solutions and interconnected systems.

Urban transportation systems within smart cities encompass diverse applications, aiming to optimize traffic flow and minimize congestion by designing intelligent systems that rely on data captured by sensors strategically placed throughout road infrastructure. This data can be leveraged to build and train machine learning models for various transportation applications, including real-time arrival estimations and prediction of traffic flows.

Notably, the potential of machine learning for the design of intelligent transport systems extends beyond these well-established applications, with one promising yet underexplored area being the automated classification of road types. Integrating models to classify road-type within interactive maps offers valuable traffic information to users, enabling them to avoid congested routes, accident-prone areas, and intersections with high frequency.

However, leveraging machine learning for road-type classification on road network graphs presents a challenge because of the scarcity of established hand crafted based methodologies for generating feature vectors from road segments. To address this, recent research has explored graph embedding techniques, that use deep learning models to capture spatial relationships between road segments.

Features are automatically extracted within the graph network structure. Feature vectors are generated with graph embedding using their neighbouring road segments. This study represents an extension of the research initially presented at the MCPR conference [9], introducing a new multi-stage graph embedding approach

doi: 10.13053/CyS-28-1-4891

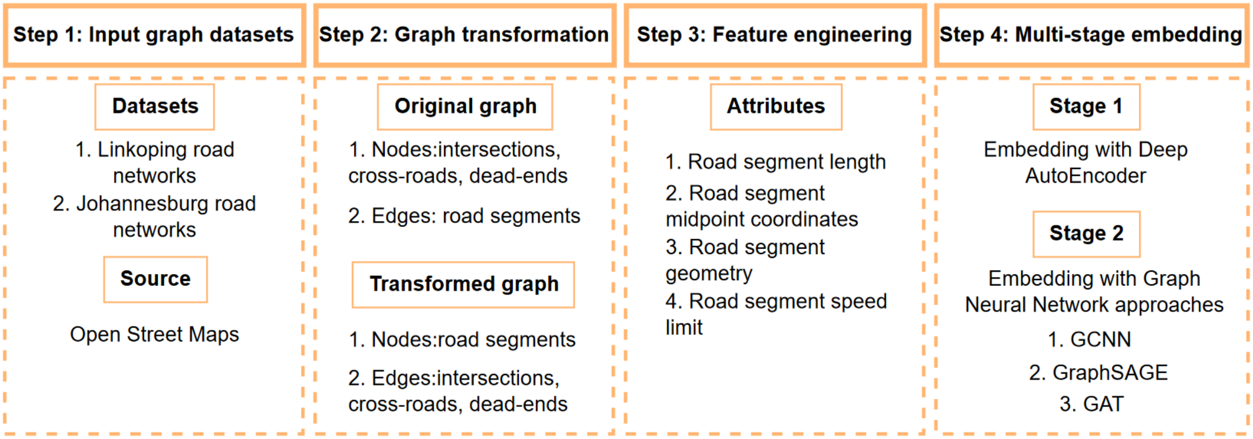

Step 1: Input graph datasets
Datasets
1. Linkoping road networks
2. Johannesburg road networks
Source
Open Street Maps
Step 2: Graph transformation
Original graph
1. Nodes:intersections, cross-roads, dead-ends
2. Edges: road segments
Transformed graph
1. Nodes:road segments
2. Edges:intersections, cross-roads, dead-ends
Step 3: Feature engineering
Attributes
1. Road segment length
2. Road segment midpoint coordinates
3. Road segment geometry
4. Road segment speed limit
Step 4: Multi-stage embedding
Stage 1
Embedding with Deep AutoEncoder
Stage 2
Embedding with Graph Neural Network approaches
1. GCNN
2. GraphSAGE
3. GAT

Fig. 1. System diagram of the proposed method

for classifying road types in real-world urban environments [10]. The original research addressed two key challenges associated with graph embedding methodologies for road networks modelling: a) the inability of graph embedding methods to conduct embedding raw road segments' feature vectors of , and b) the frequent assumption within graph embedding methods that road segment features are consistently robust and accurate.

To overcome these challenges, the initial stage of the original research utilized a Deep AutoEncoder (DAE) model to embed the raw road segments' feature vectors into significantly smaller dimensions while preserving essential features.

The resulting features from this first stage were then utilized as input for the second stage, where Graph Convolutional Neural Networks (GCNN) were employed to obtain the embedded features of a given road segment based on the features of its neighbouring road segments. The classification of road types was then accomplished using a MultiLayer Perceptron (MLP) classifier.

Building upon the original work's multi-stage graph embedding method, this study conducts a comparative analysis of various graph embedding techniques for road network modelling. The proposed approach is evaluated across multiple diverse road network datasets to ensure generalizability.

The rest of paper is organised as follows: Section 2 reviews recent advances in road-type classification, highlighting their techniques, models, and results achieved. Section 3 discusses the technical details of the proposed multi-stage graph embedding approach, outlining its components and implementation.

Section 4 presents the experimental setup, evaluation metrics, and obtained results, offering a comparative analysis with existing methods. Finally, Section 5 summarizes the essential findings of the study and outlines potential directions for future research.

2 Background and Related Work

Researchers can effectively model road networks using graph theory. This approach captures the complete topological structure of any road network, regardless of its size or complexity. Notably, graphs can represent not only spatial road networks but also diverse transportation systems, including highways, public transit networks, air routes, and waterways.

Furthermore, graph-based models can readily incorporate various network attributes such as speed limits, travel times, lane numbers, and traffic flow patterns. Essentially, graphs depict the topological structure of a network through nodes and edges.

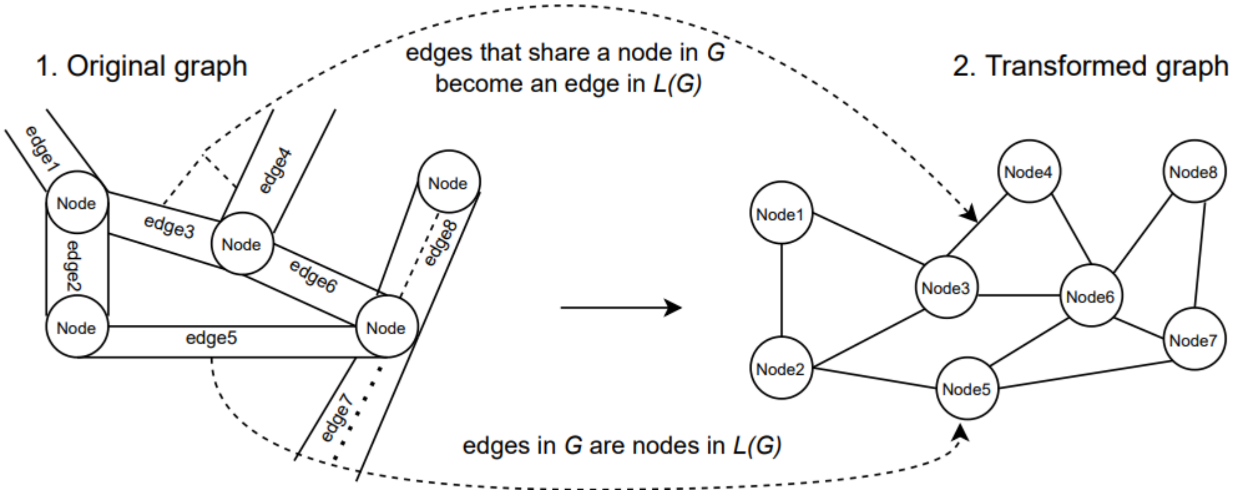


Fig. 2. Transformation of original graph to line graph

Nodes, represented by points, correspond to key locations such as intersections, dead-ends, and points of interest along the roads. Edges, represented by lines, connect these nodes and indicate the road segments between them.

Applications like traffic forecasting [1, 18, 14], speed limit optimization [12, 16, 7], and the estimation of travel time [6, 10] have witnessed successful implementations using machine learning techniques.

However, representing road networks as feature vectors for machine learning models poses a significant challenge due to the limited availability of suitable feature extraction methods in existing literature.

While recent advancements in deep learning offer an attractive avenue for automatically learning network structure and representing individual road segments based on their spatial connections to neighbouring segments, applying such techniques to graph-structured data presents unique difficulties.

Unlike commonly used data types like images and text, which are Euclidean and have fixed dimensions, the underlying connectivity patterns within graph-structured data (such as road networks) are inherently complex and non-Euclidean, posing challenges for directly applying existing deep learning methods.

Recent efforts in applying deep learning to complex, non-Euclidean graph data often rely on a fundamental approach: embedding high-dimensional graph features into a lower-dimensional Euclidean space using graph embedding techniques.

This process reduces the complexity of the data while preserving its essential relationships. By capturing these relationships in a simplified form, the model can tackle various graph-related tasks, such as predicting node attributes or connections between nodes.

Ultimately, graph embedding aims to represent each node (e.g., a road segment) with a lower-dimensional vector. This vector retains the node's similarity to the node in the original, allowing researchers to leverage standard metrics for similarity comparisons in the embedded space. Several studies have explored various graph embedding techniques for modelling road network tasks like traffic flow prediction.

While some, such as the Hybrid Graph Convolution Neural Network (HGCN) proposed in [17], capture the spatial connectivity of the network by representing toll stations as nodes and road segments as edges, however, the authors neglected to incorporate node-specific features beyond location, time, and weather.

Algorithm 1 Road segment feature extraction

```
Input: $G$ and $L(G)$.
Output: Road segment feature vector.
 1: for $s \in L(G)$ do
 2:     From $G$ obtain $l_s$.
 3:     From $s$ obtain $(x_s,\ y_s)$.
 4:     From $s$ obtain segment geometry.
 5:     if geometry exist then
 6:         Obtain 20 equally distanced points of $l_s$:
            $(lx_i,\ ly_i)_{i=1,\ 2,\ ...,\ 20}$ by divid $l_s$.
 7:         for $i = 1$ to $20$ do
 8:             Subtract $(lx_i,\ ly_i)$ from $(x_s,\ y_s)$.
 9:         end for
10:     else
11:         Convert to line geometry.
12:         Repeat steps 6 to 8.
13:     end if
14:     Obtain $S$ of the speed limits with $m$
        standard values.
15:     Concatenate features generated from
        steps 2 to 14.
16: end for
```

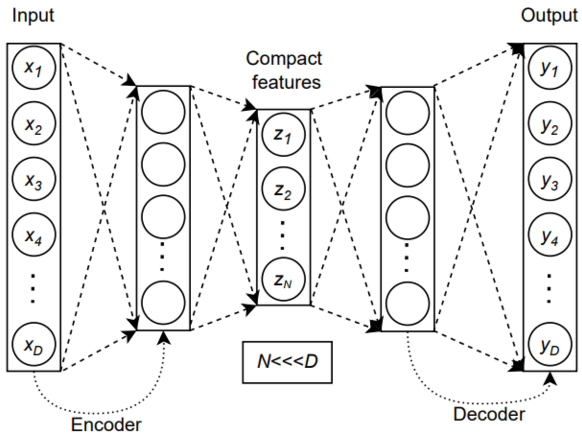


Fig. 3. Stage 1 embedding: Deep AutoEncoder

This gap is addressed in Relational Fusion Networks (RFN) introduced in [4] for speed limit classification and estimation. RFN leverages a novel graph convolution operator to effectively integrate edge information (e.g., road type, speed limits) into the representation learning process. This leads to a more comprehensive understanding of the network dynamics.

The study presented in [2] investigates the classification of different road types within realistic cities using a graph dataset extracted from Open Street Maps (OSM). Inspired by the Relational Fusion Network (RFN), the authors incorporate edge features into the learning process by transforming the initial graph into a line graph.

The further proposes a novel method for generating road segment features based on readily available attributes, such as road segment length, speed limit, and geometric characteristics. The authors then compare the performance of various embedding methods, including Graph Convolutional Neural Networks (GCCNs) [5], GraphSAGE [3], Graph Attention Networks (GATs) [13], and Graph Isomorphism Networks (GINs) [15], across different learning settings.

This comprehensive evaluation aims to identify the most effective approach for classifying road types within complex urban environments. A method to classify road types using Deep AutoEncoder (DAE) method is proposed in [2]. Similar to the work in [8], the authors generated road segment features using road attributes.

DAE was applied to reduce the dimensionality of input features while preserving the most essential features. Classification of road types was achieved using the MultiLayer Perceptron (MLP) classifier. The study proposed in [9] is an improvement to the studies presented in [2] and [8], where road segment features obtained by DAE are fed as input to GCNN before classifying road types with MLP classifier.

3 Materials and Methods

This study builds upon a novel, multi-stage graph embedding method for road-type classification tasks, originally proposed in [9]. As shown in Figure 1, the proposed method extracts road network graphs from Linkoping and Johannesburg cities using OSMnx, where nodes and edges are intersections and road segments, respectively.

The initial graph is then converted into a line graph where road segments are represented as nodes. Next, a feature extraction process is employed following the construction of both the original and transformed road network graphs.

Algorithm 2 Graph embedding with Deep Auto Encoder

```
Require: Original road features: $\mathrm{RSEGFS} \subset \mathbb{R}^N$
Outputs: Embedded road features: $\mathrm{RSES} \subset \mathbb{R}^M$
 1: Parameter definition: DAE encoder and decoder.
 2: Model definition: DAE model (encoder, decoder).
 3: for $X \in \mathrm{RSGFS}$ do
 4:     Fit input feature vectors ($X$) to DAE model.
 5:     Randomly initialise weights.
 6:     while no convergence in error difference do
 7:         Produce feature vectors on the decoder ($Y$).
 8:         Find the MSE between $X$ and $Y$.
 9:         Update weights.
10:     end while
11:     Obtain the embedding features vector ($Z$).
12:     $\mathrm{RSEGFS} \leftarrow \mathrm{RSEGFS} \cup \{Z\}$.
13: end for
14: Return features from the embedded space $\mathrm{RSEGFS}$
```

Table 1. Datasets description

Dataset	Nodes	Edges	Classes
Linkoping	6,799	13,022	5
Johannesburg	17,431	39,980	5

This process leverages the structural information encoded in both graphs to extract road properties relevant for road type classification. The core innovation: a multi-stage graph embedding approach, is introduced in step 4. Similar to the original study, the first stage leverages Deep AutoEncoder (DAE) to compress high-dimensional feature vectors into lower-dimensional representations.

However, this study goes beyond the original work by investigating alternative methods for the second embedding stage. Instead of Graph Convolution Neural Networks (GCNN), it explores the use of GraphSAGE and Graph Attention Networks (GAT) to model road types based on the features obtained from stage 1. Finally, a MultiLayer Perceptron (MLP) classifier categorises the road types.

3.1 Input Graph Datasets and Transformed Graph

This study utilizes undirected road network graphs of Linkoping and Johannesburg cities extracted from OSMnx for experimentation.

The input graph is represented as $G = (V, E)$, where V represents nodes, and E represents edges. Nodes correspond to intersections, junctions, and crossroads, while edges represent road segments connecting these nodes.

This section will only refer to the Linkoping City road networks dataset for simplicity and clarity to explain how each algorithm was applied to input graph datasets.

Graph embedding techniques aim to embed node features. However, nodes in the original graphs (crossroads, junctions, and intersections) lack crucial information for road-type classification.

Therefore, transforming the original graph into a line graph is necessary to represent road segments as nodes, thus facilitating graph embedding. Fig. 2 depicts the process of converting original graph G into its corresponding line graph $L(G)$.

This transformation involves mapping each edge (road segment) in G to a distinct node in $L(G)$. Subsequently, edges in G that share a common node (e.g., junction) are transformed into connecting edges within $L(G)$.

3.2 Labelling Road Type Classes

OSMnx represents road segments with corresponding road type labels, enabling the use of supervised learning for modelling road networks.

Unfortunately, the data suffers from imbalanced classes, with certain road types rarely appearing. To address this, similar to the original work, certain road types are merged and assigned new labels as follows:

- Class 1: Highway, yes, primary, secondary, motorway-link, trunk-link, primary-link, secondary-link.
- Class 2: Tertiary-link, tertiary.
- Class 3: Unclassified, planned, road.
- Class 4: Residential.
- Class 5: Living street.

Algorithm 3 Graph embedding with Graph Convolution Neural Networks

```
Require: Road segment embedded graph features space: $\mathrm{RSEGFS} \subset \mathbb{R}^N$.
Outputs: Road segment embeddings: $\mathrm{RSES} \subset \mathbb{R}^M$.
Define $k$ number of hops.
Define input and output layer dimensions at each $k$.
for $Z_v \in \mathrm{RSFS}$ do
    Construct computational graph.
    Initialise $W_k$.
    Set $h_v^0$ as $Z_v$.
    for $i = 1 : k$ do
        Using Eq 2, find $h_v^i$.
    end for
    Obtain embedded vector $E_v = h_v^k$ .
    $\mathrm{RSES} \leftarrow \mathrm{RSES} \cup \{E_v\}$.
end for
Return $\mathrm{RSES}$.
```

3.3 Feature Engineering

This study employs a feature engineering technique similar to the one used in [2] to ensure a fair comparison of results. In this technique, four key attributes from G and $L(G)$ are extracted to create a 58-dimensional raw feature vector for each road segment. These attributes are:

- Road segment length: Represented by a single dimension.
- Midpoint coordinates: Represented by two dimensions, one for longitude and one for latitude.
- Distances to nearby points: The midpoint is surrounded by 20 points spaced at equal distances, and the subtraction of these distances creates 20 dimensions.
- This categorical feature is represented using 15 dimensions, with each dimension corresponding to a possible speed limit.

For a given road segment s with length l_s, midpoint coordinates (x_s, y_s), and a one-hot encoded speed limit vector $S = \{s_1, s_2, s_3, \cdots, s_m\}$ (where m represents the number of possible speed limits), each road segment vector can be obtained using Algorithm 1.

3.4 Graph Embedding: The Multi-Stage Approach

The proposed multi-stage graph embedding method for classifying road types is described in this section. As previously discussed, the method employs two distinct embedding approaches. The Deep AutoEncoder (DAE) model is used in the first stage.

This model acts as a dimensionality reduction technique, compressing the high-dimensional feature vectors associated with each node (representing road segments) into a lower-dimensional, compact representation.

This compressed representation captures the essential characteristics of the road segments while discarding redundant information. In the second stage, the approach leverages graph embedding techniques to extract contextual information for each road segment.

This is achieved by incorporating the feature vectors of its neighbouring segments, previously obtained in Stage 1. Through this process, the method captures the influence and relationships between individual roads within the broader network structure.

3.4.1 Stage 1: Deep AutoEncoder Embedding:

The presented method utilizes a Deep AutoEncoder (DAE) to embed road segment features from a high-dimensional space (D) into a lower-dimensional space (N), where N is significantly smaller than D ($D >>> N$).

This dimensionality reduction process aims to achieve “compact” representations of the road segments. To understand the meaning of “compact” features, it’s crucial to grasp the DAE architecture. As depicted in Figure 3, the DAE comprises three key components:

- Encoder: This component receives a feature vector ($X_i = \{x_{i,1}, x_{i,2}, x_{i,3}, \cdots, x_{i,D}\}$) containing D features and processes it through several hidden layers with progressively decreasing dimensions.

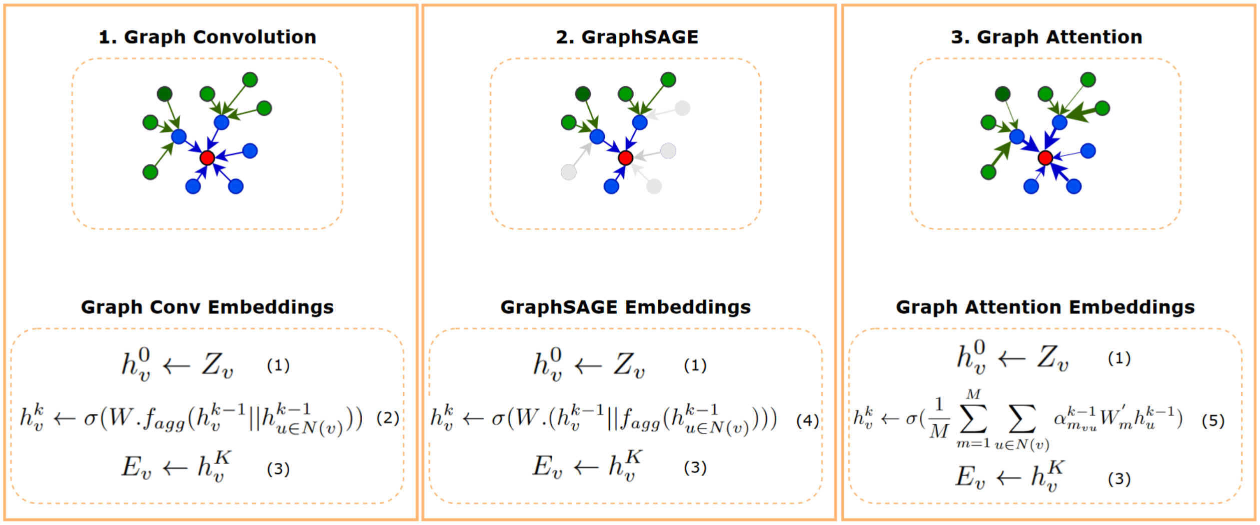

1. Graph Convolution
2. GraphSAGE
3. Graph Attention
Graph Conv Embeddings
$h_v^0 \leftarrow Z_v$ (1)
$h_v^k \leftarrow \sigma(W.f_{agg}(h_v^{k-1} || h_{u \in N(v)}^{k-1}))$ (2)
$E_v \leftarrow h_v^K$ (3)
GraphSAGE Embeddings
$h_v^0 \leftarrow Z_v$ (1)
$h_v^k \leftarrow \sigma(W.(h_v^{k-1} || f_{agg}(h_{u \in N(v)}^{k-1})))$ (4)
$E_v \leftarrow h_v^K$ (3)
Graph Attention Embeddings
$h_v^0 \leftarrow Z_v$ (1)
$h_v^k \leftarrow \sigma(\frac{1}{M} \sum_{m=1}^{M} \sum_{u \in N(v)} \alpha_{m_{vu}}^{k-1} W_m^{'} h_u^{k-1})$ (5)
$E_v \leftarrow h_v^K$ (3)

Fig. 4. Graph Neural Network approaches. 1) GCN: Target node (red) receives update from its direct neighbours (blue nodes) and neighbours of the neighbours (green nodes). 2) GraphSAGE: Target node (red) receives update only from k sampled nodes of its neighbours (blue) and m sampled nodes of neighbours of the neighbours (green). 3) GAT: Learns a scoring to weigh the influence of neighbouring nodes on the target node

– Compact Features Layer: This layer represents the heart of the dimensionality reduction. It compresses the encoded feature vector (X_i) into a lower-dimensional vector $(Z_i = \{z_{i,1}, z_{i,2}, z_{i,3}, \cdots, z_{i,N}\})$ with only N features $(N < D)$.

– Decoder: This component takes the compact feature vector (Z_i) and utilizes it to reconstruct an approximation of the original feature vector $(Y_i = \{y_{i,1}, y_{i,2}, y_{i,3}, \cdots, y_{i,D}\})$ through several dense layers with increasing dimensions.

The "compactness" of the features in the compact features layer is defined by the error difference between the original feature vector (X_i) and its reconstructed counterpart (Y_i).

If this error difference is minimal; then, the compact layer features are considered "compact" as they effectively capture the essential information of the original features in a reduced dimension.

– Number and size of hidden layers: This parameter affects the capacity of the model in learning complex patterns.

– Learning rate: This parameter controls how quickly the model updates its weights during training.

– Dimensionality of compact features: This parameter determines the compression level achieved by the embedding.

DAE utilizes a fully connected neural network architecture comprising input, output, and dedicated "compact features" layers. The encoder and decoder have input and output layers, respectively, and they share similar numbers and sizes of hidden layers.

The process begins with normalizing the input feature vectors. These normalized vectors are then fed into the encoder. ReLU activation, defined as $f(x) = \max(0, x)$, is applied to introduce non-linearities. On the decoder's output layer, the reconstructed values are normalized between 0 and 1 using the sigmoid function, defined as:

$$g(y) = \frac{1}{1 + e^{-y}}. \qquad (1)$$

Leveraging the Adam optimization algorithm, the encoder's weight parameters are iteratively

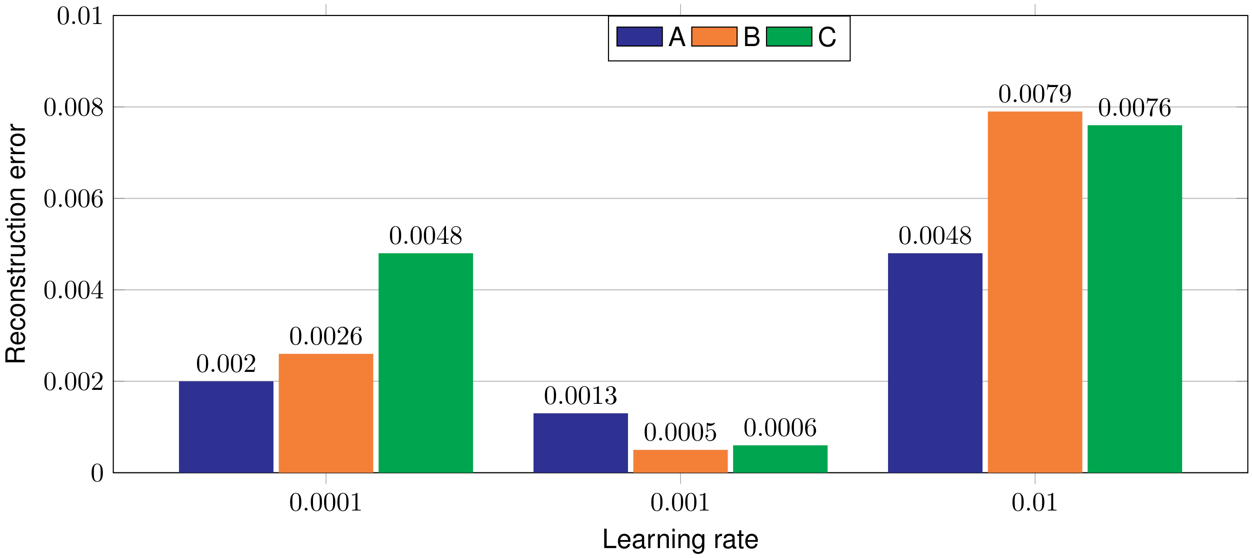


Fig. 5. Stage 1 embedding: Error difference at various DAE models and learning rates for Linkoping road network

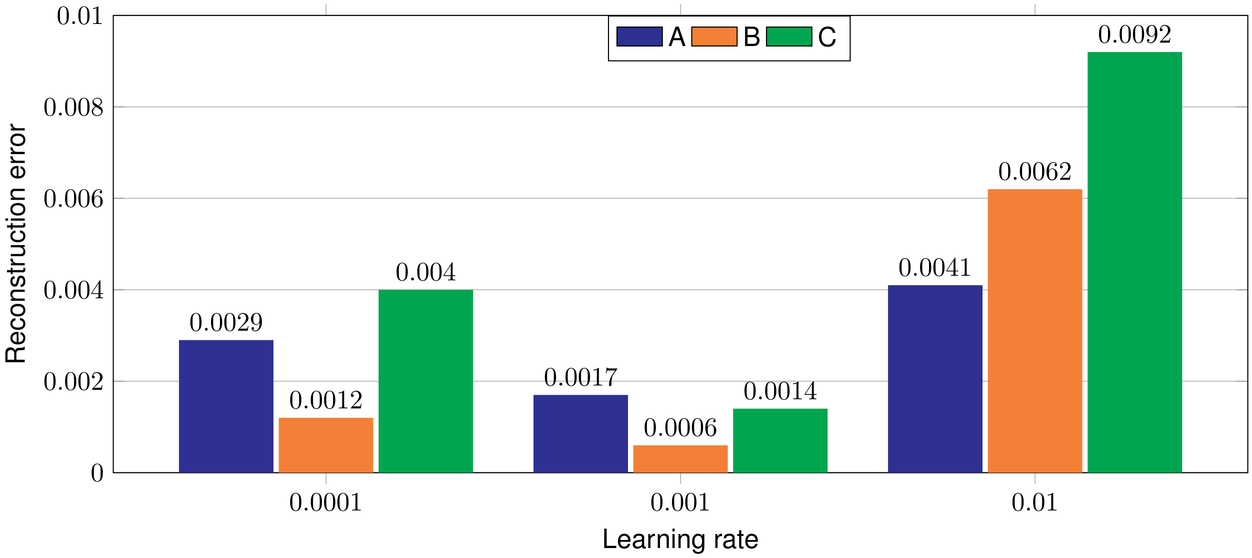


Fig. 6. Stage 1 embedding: Error difference at various DAE models and learning rates for Johannesburg road network

adjusted to minimize the reconstruction error, defined as the discrepancy between the input data and its encoded representation. Finally, the original D-dimensional road segment features are replaced with the N-dimensional compact features obtained from the model. Algorithm 2 describes the DAE model.

3.4.2 Stage 2: Embedding with Graph Neural Network Approaches:

Stage 2 leverages graph neural network (GNN) approaches to exploit the intrinsic topological and spatial relationships embedded within the graph-structured road network data.

Table 2. Description of various DAE models

Model	Hidden Layers	Sizes	N
A	5	$\{58, 49, 40, 31, 22, 13\}$	4
B	4	$\{58, 48, 38, 28, 18\}$	8
C	3	$\{58, 46, 34, 22\}$	10

This enables the extraction of richer feature representations for each road segment by incorporating information from its neighbouring segments, leading to a more comprehensive understanding of the road network's spatial context and connectivity.

As highlighted earlier, the ultimate goal of any GNN approach is to generate the embedded vector h_v^k of the sampled road segment v for each hop layer k by aggregating information from its direct neighbours $u \in N(v)$. Several GNN methods are available in the literature for generating such an embedded vector.

In this work, the comparison between GCNN, GraphSAGE and GAT is conducted. It is worth noting that inputs to each GNN are the road segment feature vectors $Z \subset \mathbb{R}^N$(from stage 1), and the outputs are the embedded vector $E \subset \mathbb{R}^M$.

3.4.3 Graph Convolution Neural Networks

A two-hop GCNN architecture [5] generates the embedded vector of a given road segment by aggregating features from its direct neighbours as well as the neighbours of the neighbours. As indicated in Equation 2 of Figure 4, GCN generates embedded vector h of target road segment v at any hop k by concatenating the embedded vectors h_v^{k-1} and $h_{u \in N(u)}^{k-1}$ of the target and neighbouring road segments, respectively at previous hop $k-1$.

It then uses some aggregator function f_{agg} to obtain the contribution of neighbouring road segments to the target road segment before applying the Sigmoid function σ. W represents the set of weights associated with the target and neighbour road segments. The experimental section of the study will investigate the performance of three GCNN aggregator functions namely, GCNN-Mean, GCNN-Max, and GCNN-Sum.

3.4.4 GraphSAGE

A two-hop GraphSAGE architecture [3] generates the embedded vector of a given road segment by aggregating information from only a set of sampled neighbouring road segments. As indicated in Equation 4 of Figure 4, GraphSAGE generates embedded vector h of target road segment v at any hop k by first applying the aggregator function f_{agg} to the embedded vector $h_{u \in N(u)}^{k-1}$ of neighbouring road segments, at previous hop $k-1$.

The aggregated vector is then concatenated to the embedded vector h_v^{k-1} before applying the Sigmoid function σ. The experimental section of the study will investigate the performance of three GraphSAGE aggregator functions: GSAGE-Mean, GSAGE-Max, and GSAGE-Sum.

3.4.5 Graph Attention Networks

Similar to GCNN, a two-hop GAT architecture [13] generates the embedded vector of a given road segment by aggregating features from its direct neighbours as well as the neighbours of the neighbours. However, GAT further learns the attention weights that describe the influence of each neighbouring road segment towards the target road segment.

As indicated in Equation 5 of Figure 4, GAT generates the average weighted embedded vector h of target road segment v at any hop k over multiple heads by applying attention weights α_{vu}^m to the corresponding neighbours shown in Algorithm 3 is the the pseudo-code for achieving the embedding task using GCNN. GraphSage and GAT follow the same pseudo-code with the only exception being the generation h_v^k.

3.5 Classifying Road Types With MLP Classifier

The study employs a Multilayer Perceptron (MLP) classifier, characterized by its non-linear activation functions and multiple hidden layers, for road type classification.

The MLP is trained, validated, and tested on feature vectors generated using a multi-stage graph embedding method. A concise summary of MLP parameters is provided, instead of

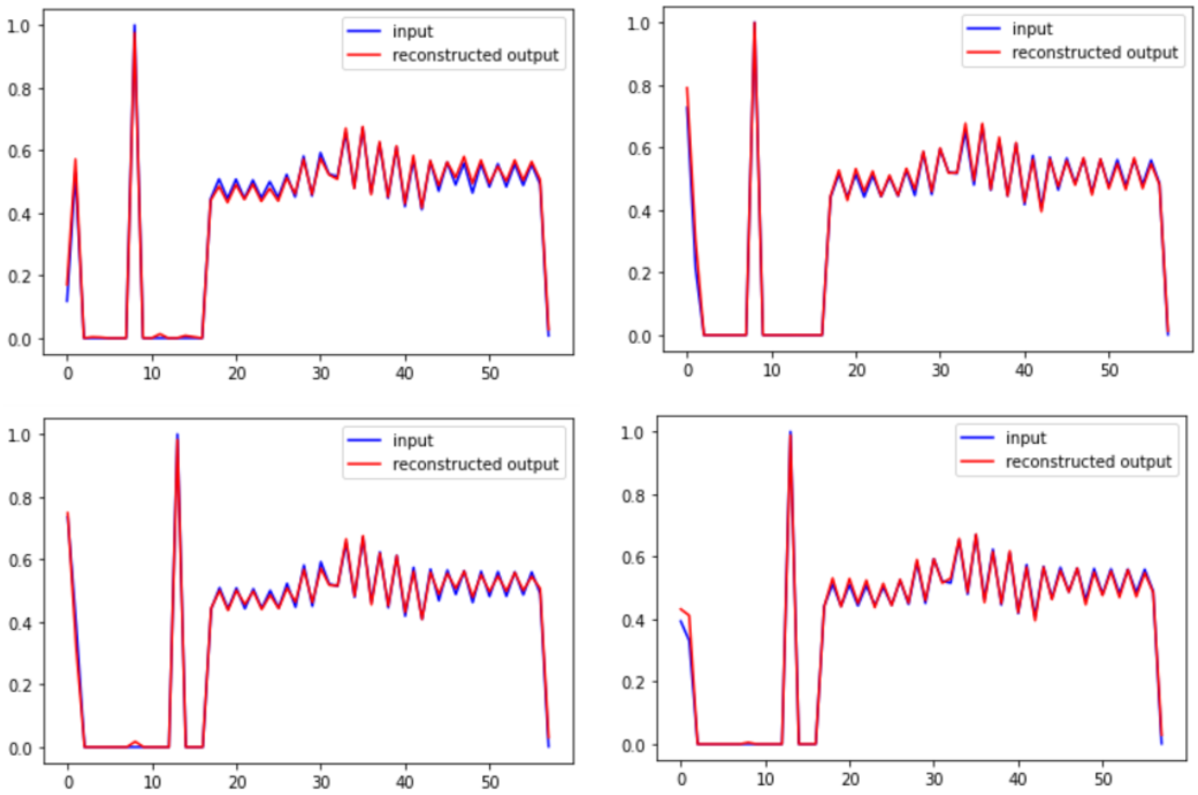


Fig. 7. Stage 1 embedding: Examples of actual and reconstructed road segment feature vectors for Johannesburg road network based on the test dataset

delving into the intricate mathematical framework of the MLP, which isn't crucial to this method's originality. To ensure a fair comparison with the approach presented in [9], a single hidden layer MLP classifier equipped with the Adam optimizer is employed.

The input layer size matches the dimensionality of the input road segment, while the output layer size aligns with the number of road type labels (five classes). Road segment feature vectors are fed through the input and hidden layers of the MLP classifier. The output layer leverages the softmax activation function to generate probability values for each road-type class.

The cross-entropy loss function measures the discrepancy between predicted and true class labels. The Adam optimizer then utilizes this calculated loss to update the MLP's weight parameters.

4 Experimental Results

Datasets of road network of Linkoping and Johannesburg cities, with 6761 and 17431 road segments (nodes), respectively, are used to carried out the experiments. Algorithm 1 is used to generate 58-dimensional feature vector from each road segment.

4.1 Stage 1: Graph Embedding with Deep Autoencoder

The aim of this stage is to embed road segment features from D dimensional space ($D = 58$) into N dimensional space with compact road segment features. Section 3.4 describes how to produce compact features. Road segments data was split into 50% for the training set, 20% for the validation set used to obtain the optimal DAE parameters and 30% for the testing set.

Table 3. Parameter settings required for the experiments

Parameters	
Learning rate	$\{0.001, 0.01, 0.1\}$
Output dimension	$\{16, 32, 64\}$
Epochs	1000
Batch size	1024
Dropout	0.2

Table 4. Prediction results for Linkoping road network datasets: Results for different graph embedding methods are shown in terms of micro F1-Score. Training time for every 50 epochs is also shown

Approach	**Training Time (s)**	**Val. F1**	**Test F1**
Raw features	04	62	59
DAE	02	66	64
GCN-Sum	26	77	72
GCN-Mean	31	76	70
GCN-Max	32	79	75
GSAGE-Sum	20	78	77
GSAGE-Mean	23	77	76
GSAGE-Max	29	79	78
GAT	29	78	76

Table 2 describes the parameters of several DAE models at varying numbers and sizes of hidden layers, respectively. For instance, model A has 5 hidden layers of size $\{49, 40, 31, 22, 13\}$ on the encoder and decoder component, while the compact layer size (N) is 4.

Figure 5 shows the error difference obtained by each DAE model at varying learning rate parameters for the Linkoping road network based on the test set.

Figure 6 shows the error difference obtained by each DAE model at varying learning rate parameters for the Johannesburg road network based on the test set. It can be seen that the lowest possible error difference is achieved with model B at 0.001 learning rate parameter.

This shows that DAE successfully performs the embedding of road segments 58-dimensional feature space into 8-dimensional space for both Linkoping and Johannesburg road network datasets. Figure 7 depicts the examples of actual and reconstructed road segment feature vectors obtained by the DAE model for the Johannesburg road network using a test dataset.

4.2 Stage 2. Embedding with Graph Convolution Neural Network

In this stage, the aggregation of information from neighbouring road segments is used to generate road segment embedded vectors using methods discussed in section 3.4.2. As in [9], the dataset was split into 70% for training, 15% for validation and 15% for testing. Each embedding model comprises a definition of a two-hop layer, in which inputs of the first layer are the 8-dimensional road segment feature vectors generated from stage 1. The M-dimensional output of the second layer is fed into the MLP classifier. The experiments are designed mainly to investigate the performances of various graph embedding methods for modelling both Linkoping and Johannesburg road networks.

4.2.1 Hyperparemeter Settings

As discussed in section 3.4.2, experiments are conducted using 7 different graph embedding approaches namely GCNN-Mean, GCNN-Max, GCNN-Sum, GSAGE-Mean, GSAGE-Max, GSAGE-Sum and GAT.

For each approach, the model that achieves the lowest micro F1 score during the validation process is selected as the best-performing model, and it is further tested on the test set. Furthermore, various learning rates (0.001, 0.01 and 0.1) and output dimension M parameters (16, 32, and 64) are investigated on each approach to obtain optimal parameters. Batch normalization is applied after each layer as a regularizer.

Combining different graph embedding approaches, output dimensions and learning rates yield 126 models for both Linkoping and Johannesburg road network datasets. Table 3 illustrates the parameters required to conduct the experiments.

Table 5. Prediction results for Johannesburg road network datasets: Results for different graph embedding methods are shown in terms of micro F1-Score. Training time for every 50 epochs is also shown

Approach	Training Time (s)	Val. F1	Test F1
Raw features	08	67	64
DAE	03	74	70
GCN-Sum	91	78	74
GCN-Mean	88	78	73
GCN-Max	116	79	73
GSAGE-Sum	65	84	81
GSAGE-Mean	81	86	84
GSAGE-Max	70	85	82
GAT	67	79	76

Table 6. Comparison of impact of method used on classification performance: The proposed method uses DAE features as input to graph embedding methods, while the method proposed in [2] uses raw features as input. The results compare the micro F1 score of GCNN, GraphSAGE and GAT using DAE features and raw features

Approach	Proposed method F1	Other method F1
GCN-Mean	70	58
GSAGE-Mean	76	62
GAT	76	76

4.2.2 Linkoping Road Networks

Table 4 shows the micro F1 score achieved by the best model on each approach based on the test set for road type classification on the Linkoping road network graph dataset. The training time after 50 epochs is also presented.

The results obtained by each embedding method are compared with the performance of DAE features and only when raw features are given to the classifier. Raw features refer to the 58-dimensional road segment features generated by Algorithm 1.

DAE features are embedded features obtained in the first stage of the proposed method, where Algorithm 2 generates 8-dimensional road segment features. As indicated, using only raw features yields the micro F1 score of 59%.

Also, using DAE features slightly improves the micro F1 score to 64%. This slight improvement is understandable given that DAE features are much more robust and accurate compared to raw features. It can be observed that all 7 graph embedding methods in the second stage of the proposed method outperform both raw features and DAE features.

However, GSAGE-Max outperforms the rest of the methods with 22% improvement compared to the performance of raw features. It can further be observed that GAT and GraphSAGE approaches have much shorter training time compared to GCNN approaches, this observation is reasonable given that GCNN uses all the neighbouring road segments to generate an embedded vector, whereas GraphASAGE and GAT only take a sample of neighbours.

4.2.3 Johannesburg Road Networks

Table 5 shows the micro F1 score achieved by the best model on each approach based on the test set for road type classification on the Johannesburg road network graph dataset.

The training time after 50 epochs is also presented. The results obtained by each embedding method are compared with the performance of DAE features and only when raw features are given to the classifier.

Raw features refer to the 58-dimensional road segment features generated by Algorithm 1. DAE features are embedded features obtained in the first stage of the proposed method, where Algorithm 2 generates 8-dimensional road segment features. As indicated, using only raw features yields the micro F1 score of 64%.

Also, using DAE features slightly improves the F1 score to 70%. It is further observed that all 7 graph embedding methods in the second stage of the proposed method outperform both raw features and DAE features.

However, GSAGE-Mean outperforms the rest of the methods with 20% improvement compared to the performance of raw features.

4.3 Comparison with Existing Works

Performance of the proposed method was compared to the model presented in [2] for classification of road types of Linkoping City. There are similarities between and GCNN methods: (1) they use the same road network graph dataset. (2) They use similar classifier parameters.

They differ on the embedding approach. In fact, Graph embedding methods proposed in [2] apply embedding using raw data as opposed to the proposed method that will initially use DAE to generate the compact version of road segments before graph embedding methods is applied.

Table 6 shows the comparison of the methods in terms of micro f1 score. It can be observed the method proposed in this study outperforms the method proposed in [2] for road type classification when GCNN-Mean and GSAGE-Mean are used as graph embedding methods.

The results achieved for GAT are similar for both studies. However, it can be observed that the use DAE embedding approach significantly improves the performance of graph embedding methods for road-type classification tasks.

5 Conclusion

A multi-stage graph embedding method for the classification of road has been presented. Experiments are conducted using the Linkoping and Johannesburg road networks dataset extracted from OpenStreetMaps. Similar to [2], road attributes such as length, mid-point coordinates, geometry and speed limit are used to generate raw features for each road segment.

Embedded road segment feature vectors are produced from raw features using a two state graph embedding method. GCNN (Sum, Mean and Max), GraphSAGE (Sum, Mean, and Max) and GAT methods were used in this study to investigate their performance for road type classification on the obtained road network graph datasets.

The results indicated that all seven methods outperform both raw and DAE features. Furthermore, GraphSAGE-Sum and GraphSAGE-Mean outperform other methods for classifying road types in Linkoping and Johannesburg cities, respectively. The results obtained by GCNN-Mean, GraphSAGE-Mean and GAT on the Linkoping road dataset were compared to the methods proposed in [2], where a similar dataset was used to solve the same tasks when only raw features were input to the graph embedding methods.

Results further indicate that the use DAE embedding method to generate compact road segment features significantly improves the performance of graph embedding methods for modelling road types. Future work of the study will generate more road segment features using attributes such as lane count.

Furthermore, replacing the one-hot encoding method with deep neural network embedding for representing categorical features is worth attempting. Additionally, the F1-score metrics used in this study have been found to exhibit a bias influenced by the imbalanced degree of the imbalanced dataset. Therefore, future work will utilise metrics such as the Matthews Correlation Coefficient (MCC).

References

1. **Deekshetha, H. R., Madhav, A., Tyagi, A. (2022).** Traffic prediction using machine learning. Evolutionary Computing and Mobile Sustainable Networks Lecture Notes on Data Engineering and Communications Technologies, pp. 969–983. DOI: 10.1007/978-981-16-9605-3_68.

2. **Gharaee, Z., Kowshik, S., Stromann, O., Felsberg, M. (2021).** Graph representation learning for road type classification. Pattern Recognition, Vol. 120, pp. 108174. DOI: 10. 1016/j.patcog.2021.108174.

3. **Hamilton, W. L., Ying, R., Leskovec, J. (2017).** Inductive representation learning on large graphs. Proceedings of the 31st

Conference on Neural Information Processing Systems, pp. 1–11.

4. **Jepsen, T. S., Jensen, C. S., Nielsen, T. D. (2022).** Relational fusion networks: Graph convolutional networks for road networks. IEEE Transactions on Intelligent Transportation Systems, Vol. 23, No. 1, pp. 418–429. DOI: 10.1109/TITS.2020.3011799.

5. **Kipf, T. N., Welling, M. (2017).** Semi-supervised classification with graph convolutional networks. Proceedings of the 5th International Conference on Learning Representations, pp. 1–14.

6. **Masiero, L., Casanova, M., Tilio, M. (2011).** Travel time prediction using machine learning. Proceedings of the 4th ACM Special Interest Group on Spatial Information and International Workshop on Computational Transportation Science, Vol. 11509. DOI: 10.1145/2068984.2068991.

7. **Modi, S., Bhattacharya, J., Basak, P. (2022).** Multistep traffic speed prediction: A deep learning based approach using latent space mapping considering spatio-temporal dependencies. Expert Systems with Applications, Vol. 189, pp. 116140. DOI: 10.1016/j.eswa.2021.116140.

8. **Molefe, M., Tapamo, J. R. (2023).** Road-type classification with deep autoencoder. Computational Intelligence and Neuroscience, Vol. 2023, pp. 1–14. DOI: 10.1155/2023/1456971.

9. **Molefe, M. E., Tapamo, J. R. (2023).** A new approach for road type classification using multi-stage graph embedding method. Proceedings of the Mexican Conference on Pattern Recognition, Vol. 13902, pp. 23–35. DOI: 10.1007/978-3-031-33783-3_3.

10. **OpenStreetMap Contributors (2022).** Planet osm. planet.osm.org.

11. **Sahoo, J., Rath, M. (2017).** Study and analysis of smart applications in smart city context. International Conference on Information Technology (ICIT), pp. 225–228. DOI: 10.1109/ICIT.2017.38.

12. **Szwed, P. (2019).** Speed limits can be determined from geospatial data with machine learning methods. International Conference on Artificial Intelligence and Soft Computing, pp. 431–442. DOI: 10.1007/978-3-030-20915-5_39.

13. **Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y. (2018).** Graph attention networks. Proceedings of the 6th International Conference on Learning Representations, pp. 1–12. DOI: 10.48550/ARXIV.1710.10903.

14. **Vázquez, J. J., Arjona, J., Linares, M., Casanovas-Garcia, J. (2020).** A comparison of deep learning methods for urban traffic forecasting using floating car data. Transportation Research Procedia, Vol. 47, pp. 195–202. DOI: 10.1016/j.trpro.2020.03.079.

15. **Xu, K., Hu, W., Leskovec, J., Jegelka, S. (2018).** How powerful are graph neural networks? Proceedings of the 6th International Conference on Learning Representations, pp. 1–17.

16. **Yan, M., Li, M., He, H., Peng, J. (2018).** Deep learning for vehicle speed prediction. Energy Procedia, Vol. 152, pp. 618–623. DOI: 10.1016/j.egypro.2018.09.220.

17. **Yang, F., Zhang, H., Tao, S. (2022).** Hybrid deep graph convolutional networks. International Journal of Machine Learning and Cybernetics, Vol. 13, No. 8, pp. 2239–2255. DOI: 10.1007/s13042-022-01520-y.

18. **Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., Yin, B. (2022).** Deep learning on traffic prediction: methods, analysis, and future directions. IEEE Transactions on Intelligent Transportation Systems, Vol. 23, No. 6, pp. 4927–4943. DOI: 10.1109/tits.2021.3054840.

Article received on 04/07/2023; accepted on 13/10/2023.
** Corresponding author is Mohale E. Molefe.*

ISSN 2007-9737

Corn/Weed Plants Detection Under Authentic Fields based on Patching Segmentation and Classification Networks

Francisco Garibaldi-Márquez[1,2], Gerardo Flores[1],
Luis M. Valentín-Coronado[*,1,3]

[1] Centro de Investigaciones en Óptica A. C., Guanajuato,
Mexico

[2] Instituto Nacional de Investigaciones Forestales,
Agrícolas y Pecuarias–Campo Experimental Pabellón,
Pabellón de Arteaga,
Mexico

[3] Consejo Nacional de Humanidades, Ciencias y Tecnologías,
Mexico

{franciscogm, gflores, luismvc}@cio.mx,
garibaldi.francisco@inifap.gob.mx

Abstract. Effective weed control in crop fields at an early stage is a crucial aspect of modern agriculture. Nonetheless, detecting and identifying these plants in environments with unpredictable conditions remain a challenging task for the agricultural industry. Thus, a two-stage deep learning-based methodology to effectively address the issue is proposed in this work. In the first stage, multi-plant image segmentation is performed, whereas regions of interest (ROIs) are classified in the second stage. In the segmentation stage, a Deep learning model, specifically a UNet-like architecture, has been used to segment the plants within an image following two approaches: resizing the image or dividing the image into patches. In the classification stage, four architectures, including ResNet101, VGG16, Xception, and MobileNetV2, have been implemented to classify different types of plants, including corn and weed plants. A large image dataset was used for training the models. After resizing the images, the segmentation network achieved a Dice Similarity Coefficient (DSC) of around 84% and a mean Intersection over Union (mIoU) of around 74%. On the other hand, when the images were divided into patches, the segmentation network achieved a mean DSC of 87.48% and a mIoU of 78.17%. Regarding the classification, the best performance was achieved by the Xception network with a 97.43% Accuracy. Then, According to the results, the proposed approach is a beneficial alternative for farmers as it offers a method for detecting crops and weeds under natural field conditions.

Keywords. Deep learning, weed detection, segmentation and classification, corn field variabilities.

1 Introduction

Corn holds great gastronomical and economic significance for many countries across the globe. In Mexico, for instance, the sown area has kept steadily in the last decade (2010 – 2020), with an average sown surface of 8 million hectares annually.

Nonetheless, the demand for this cereal increased by 136% in the same period [3], which has been compensated with importations. In this sense, factors such as land tenure, weather change, and crop management could avoid the self-sufficiency of this cereal for the country.

Among management practices, weeds elimination is one of the most important tasks in agriculture because these unwanted herbs compete with crop plants for nutrients, sunlight,

doi: 10.13053/CyS-28-1-4890

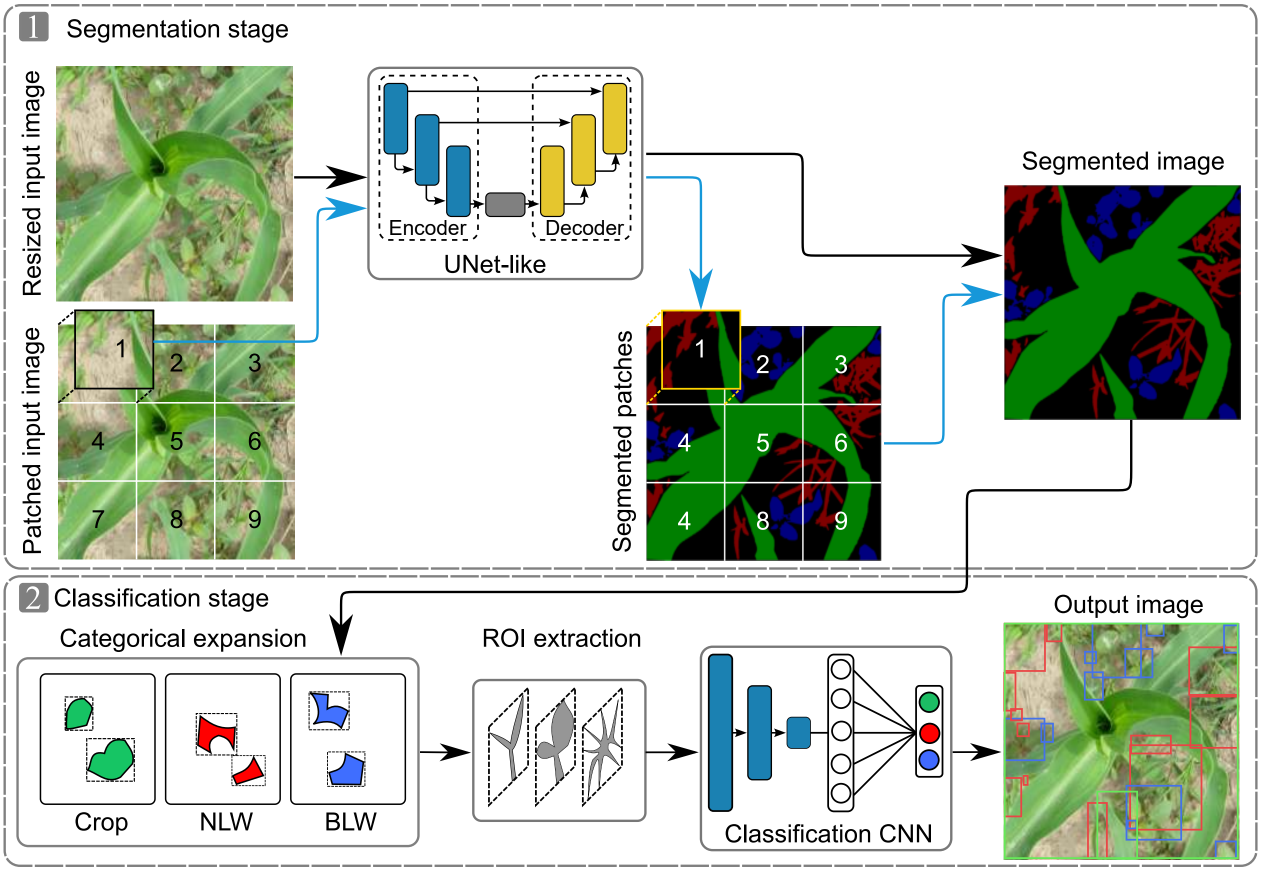


Fig. 1. Propose a method for detecting crop and weed plants in authentic corn fields, utilizing segmentation and classification networks.The resulting image output comprises of green, red, and blue boxes, each representing the Crop, NLW, and BLW classes, respectively

and water [19] and could lead to 90% of kernel yield reduction if not controlled in time [13]. The most commonly employed control strategy to eradicate weeds from cornfields is through the application of herbicides.

However, the excessive use of herbicides has resulted in environmental pollution [9]. This is predominantly due to the uniform application of significant volumes of these chemicals throughout the entire field, even in regions where weeds are absent [11]. Consequently, to address the environmental impact of herbicides while sustaining crop yield, researchers have developed a sophisticated technique termed site-specific weed management (SSWM). This method involves the targeted application of chemicals exclusively in areas where weeds are present, thereby minimizing environmental pollution.

Operating systems that can effectively distribute adequate herbicides on individual weed plants or patches of them in the fields is plausible [14]. Nevertheless, detecting (localizing and classifying) these plants in natural crop environments has been reported to be a demanding and intricate task [6]. This challenge is primarily attributed to diverse parameters, such as the intensity of sunlight, the density of plants, foliage occlusions, and the variety of plant species.

The implementation of Convolutional Neural Networks (CNNs) for identifying crop and weed plants has gained significant traction in recent times. YOLO [11] and Faster-RCNN [10] are

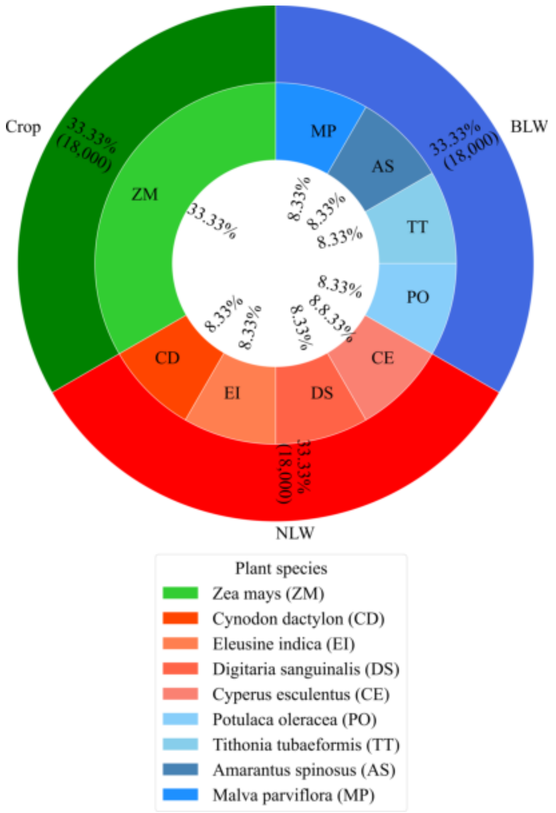


Fig. 2. Visualization of the experimental dataset showcasing plant species grouped into distinct classes, with corresponding labels meticulously traced per individual plant species

among the popular architectures employed for this purpose. However, their efficacy is limited when detecting plants in densely populated fields.

A promising alternative is the technique of semantic segmentation, which separates the plants from the background, although it requires additional algorithms for classification.

While established architectures exist for the semantic segmentation of objects within images, there has been limited research on weed segmentation in corn fields, mainly due to the unavailability of a large and diverse corn/weed dataset.

Here, we use deep learning models to segment and classify corn and weed plants under authentic environments and high plant density.

1.1 Related Works

The segmentation of plants in natural conditions poses a significant challenge due to the complexity of the variables involved. These variables include the plant species, density, foliage occlusion, morphological changes across growth stages, soil appearance, and sunlight intensities.

The presence of these variables makes it challenging to extract and classify the unique features of plants. Few works in the literature have been conducted on the segmentation of weeds in corn crops. However, Fawakherji et al. [5] recently proposed a method for segmenting a multispectral dataset.

The images were captured using an unmanned aerial vehicle (UAV) within a natural cornfield environment and classified into soil and green plants. A VGG-UNet model was then trained using four sub-dataset images derived from Red, NIR, synthetic images from the Normalized Difference Vegetation Index (NDVI), and RED+NIR+NDVI.

Results showed a mean accuracy of 73%, 85%, 92%, and 88%, respectively. It is worth noting that multispectral channels offer better segmentation performance compared to the visible spectrum [2, 12]. However, the associated cost of infrared sensors would present a challenge for autonomous weed control systems.

Visible spectrum cameras have been utilized in discriminating between corn and weeds in real fields. For instance, in the work of Quan et al. [16], the segregation of weeds under complex cornfield environments was explored using the BlendMask network. An extensive dataset of 5,700 images was formed, which included two broadleaf weeds and one narrowleaf weed.

Results indicated that a ResNet101 backbone yielded a higher mIoU of 60.7% compared to 50.2% with ResNet50. More recently, Picon et al. [15] employed the PSPNet network in segmenting a corn/weed dataset in natural fields, resulting in a Dice Similarity Coefficient (DSC) of 25.32%.

This dataset consisted of corn, narrowleaf weed (three species), and broadleaf weed (three species). However, the authors acknowledged that

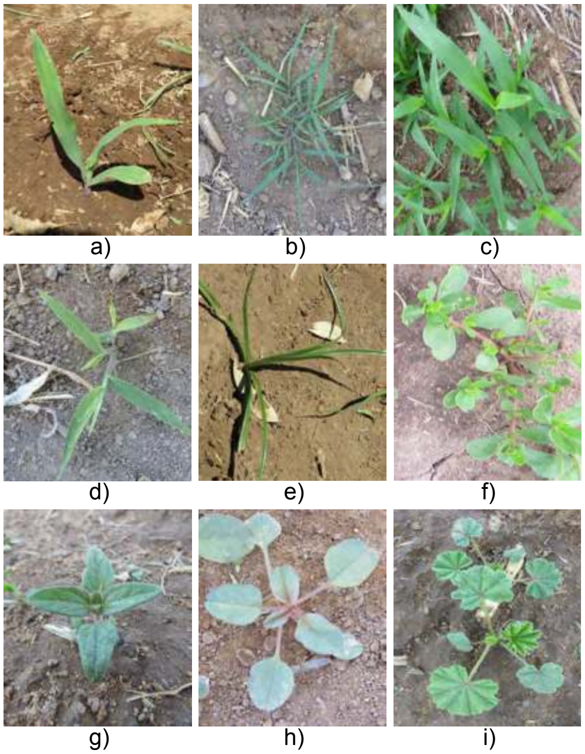


Fig. 3. Sample of the plant species of the experimental dataset. a) Zea mays, b) Cynodon dactylon, c) Eleusine indica, d) Digitaria sanguinalis, e) Cyperus esculentus, f) Portulaca oleracea, g) Tithonia tubeoformis, h) Amarantus spinosus and i) Malva parviflora

the narrowleaf class was not correctly classified due to its visual similarity to the crop class.

In this work, we present a large dataset of corn and weed images that were captured in authentic natural corn fields. This dataset includes four monocotyledon plant species and four dicotyledon plant species as weeds, as well as corn plants as the crop.

To detect the Crop, narrowleaf weeds (NLW), and broadleaf weeds (BLW), we propose a deep learning-based approach. Each weed class, NLW and BLW, groups the four plant species of weeds, respectively. The proposed approach performs well despite the challenging conditions presented in the acquired images.

The rest of the document is structured as follows: Section 2 contains the dataset description as well as the implementation details of the segmentation approaches. Section 3 presents the primary results of the experiments, and Section 4 provides the conclusions of the work.

2 Materials and Methods

According to the results obtained from our previous work [7], the UNet-like model [17], whose encoder layer was the network ResNet101, performs the segmentation of plants adequately. However, it has been observed that the model often misclassifies the pixels of the isolated Regions of Interest (ROIs).

This evidenced the necessity of developing a vision system with the ability to detect corn plants, narrowleaf weeds, and broadleaf weeds under authentic corn fields, giving the excessive field variabilities. This gap is covered by proposing a detection method based on deep learning segmentation and classification networks, as shown in Figure 1.

The algorithm comprises two main stages: a segmentation stage and a subsequent classification stage. In the segmentation stage, an image with multiple plants is segmented using a UNet-like architecture. The segmentation process has been carried out under two approaches.

In the first approach, the input images are segmented in a simple step by simply resizing them, whereas in the second approach, the input images are divided first into patches to avoid the loss of significant features, and then each patch is segmented. Subsequently, in the classification stage, the pixels belonging to each class (Crop, NLW, or BLW), from the segmented image are first separated into single-class images, and then each image is transformed into binary masks for the easy extraction of the ROIs under scenarios of high density of plants.

These ROIs are extracted using the well-known connected component analysis (CCA) [8]. Then, in the final stage, an image is obtained within the detected plants that have been detected. To perform this task, the networks ResNet101, VGG16, Xception, and MobileNetV2 have been implemented and evaluated. The implementation details of the segmentation

Table 1. Metrics adopted for evaluating the UNet-Like model

Name	Acronym	Definition
Dice Similarity Coefficient	DSC	$\dfrac{2\ \mathrm{TP}}{2\ \mathrm{TP}+\mathrm{FP}+\mathrm{FN}}$
Intersection over Union	IoU	$\dfrac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}$
Mean Intersection over Union	mIoU	$\dfrac{1}{N}\sum_{j=1}^{N}\mathrm{IoU}_j$

and classification networks are covered in Sections 2.2.1 and 2.2.2, respectively.

2.1 Dataset Description and Image Pre-Processing

The dataset consisted of 12,000 visible spectrum images captured from five corn fields in Aguascalientes, Mexico. Three corn fields were established during the spring-summer agricultural cycle of the year 2020, and two additional corn fields in the same cycle of the year 2021.

The dataset images have varying dimensions, including 4,608 × 3,456 pixels, 2,460 × 1,080 pixels and 1,600 × 720 pixels. During the process of capturing images, the camera was positioned at a distance between 0.4 m and 1.5 m above the soil surface. Consequently, a significant number of images were captured from a top-down perspective, while a limited number had a side view.

Furthermore, it is noteworthy that most top-down view images were captured from a distance greater than 1 m to avert dust accumulation on the camera lens, which can be caused by agricultural tractors traveling through crop fields.

It is, therefore, recommended that during the tentative instrumentation, the camera should be positioned at a height of more than 1 m from the ground to avoid such issues. The dataset contains various factors that introduce variability.

The plants' variability is determined by the number of species, instances in a single image, and occlusion and foliage overlap. Changes in zoom and side views also affect the scale and perspective of the plants.

Furthermore, the dataset includes plants in different growth stages, starting from two true leaves to seven true leaves, captured every five days. Soil status is another parameter that affects the dataset, including humidity conditions, organic matter content, and changes in its appearance, such as color and texture. The images were captured in different sunlight intensities, including morning, noon, and evening, as well as on sunny and cloudy days.

After integrating the dataset, meticulous manual annotation of each image at a pixel level was conducted. The aim of this process was to precisely quantify not only the crop species (Zea mays L.), but also eight different weed plant species: four narrow-leaf weeds (NLW) and four broadleaf weeds (BLW). Figure 2 summarized the plant species and the labels traced per each of them. Noticed that they have been grouped into the classes Crop, NLW, and BLW. Furthermore, Figure 3 shows a sample of the plant species of the dataset.

To develop an effective detection strategy, a Convolutional Neural Network (CNN) was trained using a sub-dataset comprising of individual-plant images that were extracted from the original experimental dataset's multi-plant images. This approach ensured that the dataset used for training the classification networks was well-balanced, with 18,000 images per class.

2.2 Training of the Architectures

The detection approach proposed involves two stages, as previously mentioned. The first stage employs a UNet-like network for the segmentation process. The second stage involves implementing and evaluating ResNet101, VGG16, Xception, and MobileNetV2 networks for the classification process. The CNN architectures were trained on a desktop computer that boasted a Core i7 processor, 32 GB of RAM, and an NVIDIA GeForce RTX 3070Ti GPU with 8GB of memory.

The implementation was carried out in Python 3.8, utilizing the Keras framework with Tensorflow 2.5.0 as the backend.

Table 2. Metrics adopted for evaluating the classification models

Name	Definition
Accuracy	$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$
Precision	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$
Recall	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$
F_1-score	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

2.2.1 UNet-Like

First of all, to train the UNet-like proposed model, a fine-tuning of the hyper-parameter was performed. Therefore, a few training steps were implemented before figuring out the better configuration of UNet-like network. In a first trial, the encoder and decoder blocks were trained, and their weights were randomly initialized.

Then, in a second trial, a transfer learning strategy was implemented to the network, i.e., the weights of the convolutional layers of ResNet101 (encoder) were imported from that when it was trained in the ImageNet dataset [4], and then they were frozen. The learning rate, the optimizer, and the number of epochs also were changed.

In the segmentation approach where all input images are segmented in a single step, it was only necessary to resize the image and train the network. On the other hand, an image padding pre-processed was implemented in the approach in which the input images were divided into patches.

Thus, the original size of input images remains unchanged, and pixels of value 0 were added on two sides of them to obtain fixed-size patches. The loss function always was the dice loss, since it is very strict for segmentation tasks because it penalizes those predominant pixels of certain classes.

The computation of dice loss is as follows:

$$L_{\text{Dice}} = 1 - \frac{2\,y\,y^{*} + 1}{y + y^{*} + 1}, \tag{1}$$

where y and y^{*} refer to the ground truth and the predicted model value, respectively.

2.2.2 Classification Networks

In all the cases, the convolutional layers of the classification networks were the original from the architectures, but the Fully Connected (FC) layers were proposed. Then, we have established the parameters and hyper-parameters of these architectures, following a similar approach to that of the segmentation network. Firstly, the weights of the convolutional and FC layers were initialized randomly and trained. Secondly, the convolutional layers were initialized with weights obtained from the ImageNet dataset and subsequently retrained with our own dataset.

In this step, only the FC layers were trained. Furthermore, we have changed the FC layers from two to three. Thus, the neural network architecture employed in our study consisted of variable numbers of neurons, ranging from 512 to 4,096, with increments of 512 for the first and second layers. The ReLu activation function was used for the first two layers, while the output, which was the third layer, comprised three neurons with a softmax activation function.

This choice of activation function was motivated by the three specific classes of our dataset, namely Crop, NLW, and BLW. To optimize the neural network's performance, we employed a fine-tuning process that involved varying the optimizer, learning rate, loss function, and number of epochs. This approach allowed us to achieve superior results and ensure the accuracy of our model.

2.3 Evaluation Metrics

The proposed approach was evaluated in two stages. The initial stage involved the assessment of the segmentation process, followed by an evaluation of the classification stage.

The chosen evaluation metrics for the UNet-like segmentation network are Dice Similarity Coefficient (DSC), Intersection over Union (IoU), and mean Intersection over Union (mIoU). These metrics have been selected to assess the

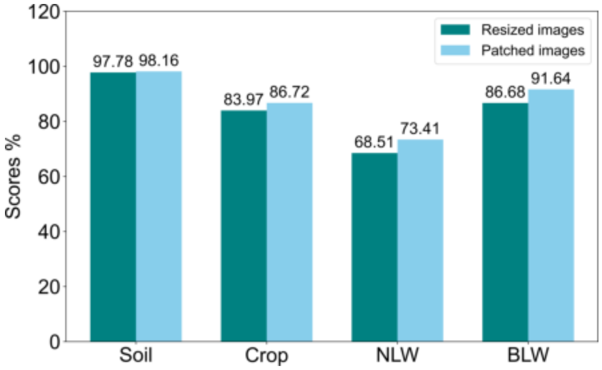


Fig. 4. DSC achieved by the UNet-like model when the input images were resized and divided into patches

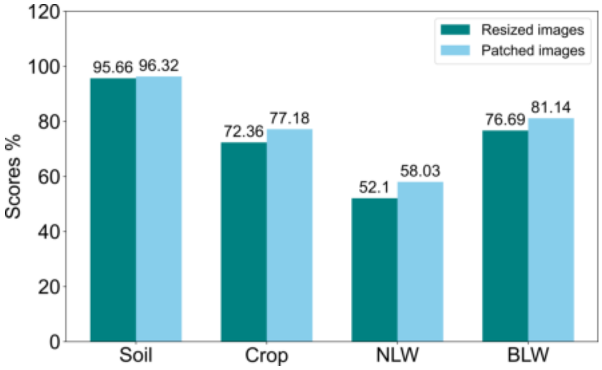


Fig. 5. IoU reached by the UNet-like model when the input images were resized and divided into patches

network's performance and provide an accurate representation of its effectiveness.

DSC pixel-wise compares the similarity between the ground truth and the predicted mask, reflecting their size and localization agreement as perceptual quality [1]. IoU is employed to calculate the percentage of overlap and align concerning the desired outcome.

The metrics utilized to evaluate the networks' performance are presented in Table 1. Noticed that the $mIoU$ is computed considering the total number of classes (N) of the dataset.

The performance assessment of our classification models was conducted using the established metrics of Accuracy, Precision, Recall, and F_1-score. Table 2 offers an insightful overview of these metrics. In Table 1 and 2, the TP (true positive), TN (true negative), FP (false positive), and FN (false negative) values are directly estimated from the confusion matrix.

3 Results and Discussion

This section provides an overview of the results obtained from the segmentation network UNet, as well as the classification networks' performance. Furthermore, we will undertake a comprehensive analysis of the achievements of each task. A set of representative images showcasing the accurate detection of crop and weed plants is also presented to understand the system output better.

3.1 Performance of the Unet-Like Model

The segmentation stage has been carried out under two approaches. The first one consists of segmenting the resized input images, whereas in the second approach, the input images are divided first into patches, and then each patch is segmented.

In either case, the best results were obtained when the transfer learning technique was implemented to train the UNet-like model. Regardless of the approach, the network input image size was 512 × 512. In addition, and according to the experimentation, the Adam optimizer with a learning rate of 0.0001 was observed to fit better into our dataset. The number of epochs used to train the model was 100.

The performance of the DSC metric of the trained UNet-like model, when images were resized and divided into patches is depicted in Figure 4. It is observed that the four classes of the dataset were better segmented by the UNet-like model when the images were divided into patches since the DSC of the four classes is superior under this scenario. Specifically, the BLW class was found to be better segmented by the network, followed by the Corn class, and finally, the NLW class, when focusing solely on the plant classes.

A narrow analysis indicates that the classes Crop, NLW, and BLW were 2.75%, 4.90% and 4.96% better segmented respectively when images were divided into patches, in contrast when they were resized and segmented in a step.

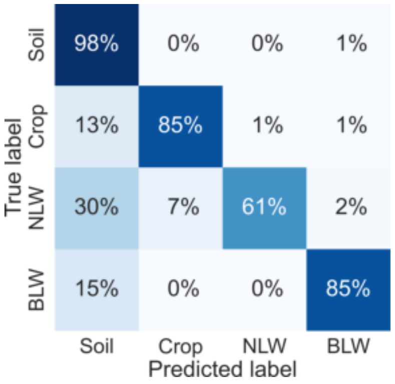


Fig. 6. Confusion matrix obtained when the images were solely resized for segmenting

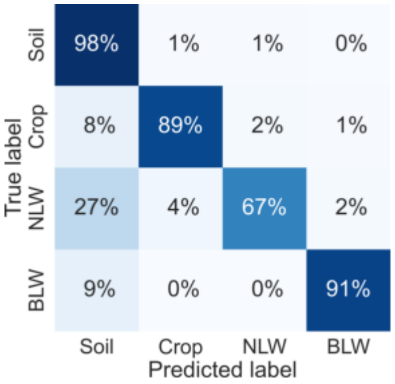


Fig. 7. Confusion matrix obtained when the images were divided in patches for segmenting

The same behavior of the UNet-like model is observed for the metric IoU under the two segmentation scenarios, as Figure 5 shows. That is, the UNet-like model performs better for all the classes when images are divided into patches. The IoU reaffirms that the BLW was the best-segmented class, then the class Crop and the worst was the class NLW.

Segmenting the patches increased 4.82%, 5.93%, and 4.45% the IoU metric for classes Crop, NLW, and BLW, respectively, concerning the IoU obtained where images were resized. Segmenting the patches obtained from the input images, without modifying the original size, may help to preserve significant features of the classes, then, the performance of the UNet-like model, under this scenario, is superior.

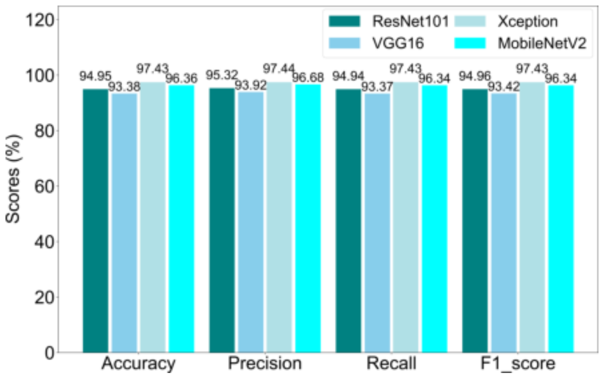


Fig. 8. Performance classification networks

In summary, when the images were resized during segmentation, the UNet-like model achieved a mean DSC of 84.27% and mIoU of 74.21%. In the other condition, when the images have been divided into patches, the UNet-like model achieved a mean DSC of 87.48% and a mIoU of 78.17%.

It is important to note that the magnitude values of the metrics used in our study are deemed acceptable as they surpass the performance of similar works reported in the literature. These works encompassed the segmentation of corn and weed plants in natural environments, as exemplified by the works of Quan et al. [16] and Picon et al. [15].

Additionally, our trained model can potentially segment other monocotyledon and dicotyledon plant species, given that the classes NLW and BLW, for which the architecture was trained, contain four species of each group with distinct growth stages. Moreover, the field variability was varied enough, making our trained model useful for segmenting a range of plant species.

In Figure 6 and Figure 7, we present two confusion matrices in which the performance of the UNet-like model can be appreciated. These matrices showcase the percentage of correctly and incorrectly classified pixels. In particular, Figure 6 shows the confusion matrix for the scenario where the input images were resized. In contrast, Figure 7 shows the confusion matrix for the scenario where the input images were divided into patches.

Under the two segmentation approaches, the classes Crop and BLW were better segmented

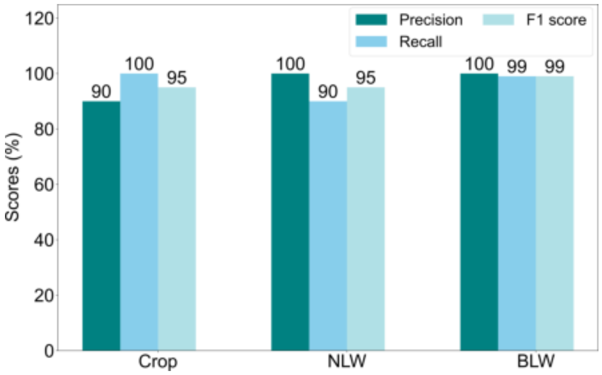


Fig. 9. MobileNetV2 classification performance

than the class NLW. In the first approach, the model was able to classify the pixels belonging to the Crop and BLW classes with a high degree of similarity, achieving an accuracy of 85%. However, the model's ability to classify the pixels of the NLW class was relatively lower, with an accuracy of only 61%.

In contrast, when the images were divided into patches, the pixels of the class BLW were better classified as such (91.0%), next the pixels of the class Crop (89.0%) and the worse classified were the pixels belonging to the class NLW.

In all the cases, the UNet-like model classified better the pixels belonging to the classes of plants into their corresponding class when the images were divided into patches, compared to that when they were solely resized.

It is also observed that the UNet-like model confused in more magnitude the pixels belonging to the classes of plants as if they were soil, under the two scenarios.

3.2 Performance of the Classification Networks

The implementation of transfer learning resulted in a notable improvement in classification performance. Specifically, the fully connected (FC) layers were tuned to our dataset to achieve this. The FC block comprised three layers, and it was observed that the classification accuracy was enhanced when the first two layers had 2,048 neurons.

Furthermore, the Adam optimizer with a learning rate of 0.0001 was utilized, and the categorical cross-entropy loss function was employed to minimize the error. The model was trained for 50 epochs on the complete dataset, with input images sized at 224 × 224 pixels for networks.

The macro performance of the networks ResNet101, VGG16, Xception, and MobileNetV2 on classifying the ROIs extracted from the segmented images are shown in Figure 8. It is worth mentioning that these metrics have been estimated under the segmentation scenario when the images were divided into patches.

As it is appreciated, Xception performed better, then MobiNetV2, and subsequently ResNet101, and the worse performance was depicted by VGG16, as the metrics Accuracy, Precision, Recall, and F_1-score indicate. In real-field applications, the inference time is crucial.

In this way, from the studied classification networks, the computation cost of MobileNetV2 network could be 8 to 9 times smaller than the rest of the architectures since it implements depthwise separable convolution (depthwise convolutions and pointwise convolutions), instead of conventional convolutions. Depthwise separable convolutions reduce trainable parameters [18].

For this reason, it was decided to present in Figure 9 the fine performance of MobileNetV2 model on classifying plants that belong to the classes Crop, NLW, and BLW. Analyzing first the metric Recall, it indicates that 100% of the images belonging to the class Crop were classified as such, 90% of the images of the class NLW were classified as such and 99% of the images from the class BLW were correctly classified by the MobileNetV2 model. Since the precision of the class Crop is 90%, it indicated that the model is misclassifying 10% of the plants of the class NLW as if they were corn, because the precision of this class, NLW, is 100%. Therefore, the metrics Precision, Recall, and F1-score make us realize that the better classified class was BLW. Finally, the mean classification performance among classes was 95%, 95% and 99%, for Crop, NLW, and BLW, accordingly, which is indicated by the F_1-score.

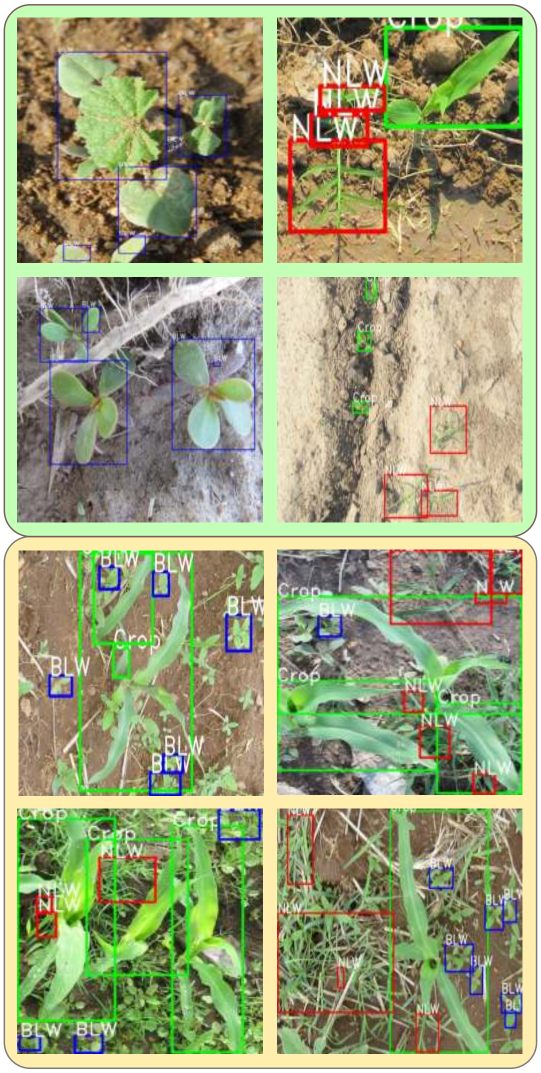


Fig. 10. Examples of the output images generated through the implemented detection method, utilizing both the segmentation and classification networks. The initial two rows exhibit images with low plant density, while the subsequent two rows showcase images featuring high plant density. Across all samples, the visual annotations include a green box denoting the crop, a red box indicating non-leaf weeds (NLW), and a blue box highlighting broadleaf weeds (BLW)

3.3 Detection Approach Visualization

Detecting objects in an image involves identifying the location and class of every object within the image. Figure 10 shows a sample of images in which the plant classes have been detected by applying our proposal. The first two rows of Figure 10 contain images that have a low density of plants, and occlusion of the foliage does not exist.

On the contrary, the images in the third and fourth rows show a high density of plants, and the foliage is partially covered in both cases. It's worth noting that the green boxes in all samples represent the Crop class, the red boxes represent the NLW class, and the blue boxes represent the BLW class. A visual inspection of the images with a low plant density indicates that almost all the green regions have been detected.

Nonetheless, since the localization of the plants is slightly related to the region provided by the segmentation model, more than one bounding box often appears in a simple image.

When high-density plant images are analyzed, it has been observed that most plant classes are accurately detected.

Nevertheless, due to the segmentation model's region extraction, it is common for multiple plants of the same classification to share a bounding box due to the density of foliage.

It is also appreciated that certain high-density plant images were not detected by the segmentation model due to the confusion of the pixels that belong to the plant classes with those of the soil. Although, in some cases, the detection covers part of the foliage of the plants, the implementation of this vision system for spraying herbicides under real corn fields is still adequate.

It is because the systemic herbicides are absorbed by the plants and gradually propagated throughout their vascular system, killing all their organs. Therefore, it has been observed that applying herbicides on a targeted section of plant foliage is adequate to eliminate them.

When the trained segmentation model considers multiple plants in a region, it could be tackled by subdividing the bounding box for spraying less area of the foliage.

4 Conclusion and Future Work

In this work, we present a method for detecting corn plants, as well as four narrowleaf (NLW) and four broadleaf (BLW) weed species in authentic corn fields. The proposed methodology comprises two distinct stages, namely segmentation and classification. A UNet-like architecture is employed from two different perspectives during the segmentation stage. The first consider segmenting the images entirely by resizing them, and the second approach consists of dividing the images into patches and then segmenting them.

In the classification stage, the four architectures ResNet101, VGG16, Xception, and MobileNetV2 have been evaluated on classifying the ROIs from the segmented images. Upon resizing the input images, the UNet-like model was able to attain a DSC of 84.27% and a mIoU of 74.21%. In the other scenario, when the images were divided into patches, the UNet-like model achieved a mean DSC of 87.48% and a mIoU of 78.17%. Regarding the classification networks, Xception performed better than MobiNetV2 and ResNet101. VGG16 showed the worst performance.

The segmentation model exhibited some limitations in accurately identifying the three classes of plants and the soil class. A significant proportion of pixels was frequently misclassified between these categories. Moreover, the models performed better in classifying the BLW class, but struggled with the NLW class, both in segmentation and classification. Notably, the models frequently mislabeled NLW as Crop.

In general, the models perform well despite the complexity of the dataset. In future work, we aim to enhance the segmentation performance of networks operating under high-density plants and develop a robust model capable of adapting to various field variabilities. Then, the dataset will be enlarged with more plant species and blur images captured with cameras mounted over moving agricultural tractors.

Acknowledgments

Francisco Garibaldi Márquez thanks to CONAHCYT for the scholarship granted.

References

1. **Bertels, J., Eelbode, T., Berman, M., Vandermeulen, D., Maes, F., Bisschops, R., Blaschko, M. B. (2019).** Optimizing the dice score and jaccard index for medical image segmentation: Theory and practice. Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 92–100. DOI: 10.1007/978-3-030-32245-8_11.

2. **Das, M., Bais, A. (2021).** DeepVeg: Deep learning model for segmentation of weed, canola, and canola flea beetle damage. IEEE Access, Vol. 9, pp. 119367–119380. DOI: 10.1109/access.2021.3108003.

3. **de Informacion Agroalimentaria y Pesquera, S. (2023).** Anuario estadístico de la producción agrícola. nube.siap.gob.mx/cierreagricola/.

4. **Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L. (2009).** ImageNet: A large-scale hierarchical image database. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. DOI: 10.1109/cvpr.2009.5206848.

5. **Fawakherji, M. (2020).** Crop and weed classication using pixel-wise segmentation on ground and aerial images. International Journal of Robotic Computing, Vol. 2, No. 1, pp. 39–57. DOI: 10.35708/rc1869-126258.

6. **Gao, J., French, A. P., Pound, M. P., He, Y., Pridmore, T. P., Pieters, J. G. (2020).** Deep convolutional neural networks for image-based convolvulus sepium detection in sugar beet fields. Plant Methods, Vol. 16, No. 1. DOI: 10.1186/s13007-020-00570-z.

7. **Garibaldi-Márquez, F., Flores, G., Valentín-Coronado, L. M. (2023).** Segmentation and classification networks for corn/weed detection under excessive field variabilities. Proceedings of the Mexican Conference on Pattern Recognition, Vol. 13902, pp. 125–138. DOI: 10.1007/978-3-031-33783-3_12.

8. **Haralick, R. M., Shapiro, L. G. (1992).** Computer and robot vision. Addison-Wesley Publishing Company, Inc.

9. **Hashemi-Beni, L., Gebrehiwot, A., Karimoddini, A., Shahbazi, A., Dorbu, F. (2022).** Deep convolutional neural networks for weeds and crops discrimination from UAS imagery. Frontiers in Remote Sensing, Vol. 3, pp. 755939. DOI: 10.3389/frsen.2022.755939.

10. **Hu, C., Sapkota, B. B., Thomasson, J. A., Bagavathiannan, M. V. (2021).** Influence of image quality and light consistency on the performance of convolutional neural networks for weed mapping. Remote Sensing, Vol. 13, No. 11, pp. 2140. DOI: 10.3390/rs13112140.

11. **Hussain, N., Farooque, A., Schumann, A., McKenzie-Gopsill, A., Esau, T., Abbas, F., Acharya, B., Zaman, Q. (2020).** Design and development of a smart variable rate sprayer using deep learning. Remote Sensing, Vol. 12, No. 24, pp. 4091. DOI: 10.3390/rs12244091.

12. **Milioto, A., Lottes, P., Stachniss, C. (2018).** Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs. Proceedings IEEE International Conference on Robotics and Automation, pp. 2229–2235. DOI: 10.1109/icra.2018.8460962.

13. **Nedeljković, D., Knežević, S., Božić, D., Vrbničanin, S. (2021).** Critical time for weed removal in corn as influenced by planting pattern and PRE herbicides. Agriculture, Vol. 11, No. 7, pp. 587. DOI: 10.3390/agriculture11070587.

14. **Partel, V., Charan-Kakarla, S., Ampatzidis, Y. (2019).** Development and evaluation of a low-cost and smart technology for precision weed management utilizing artificial intelligence. Computers and Electronics in Agriculture, Vol. 157, pp. 339–350. DOI: 10.1016/j.compag.2018.12.048.

15. **Picon, A., San-Emeterio, M. G., Bereciartua-Perez, A., Klukas, C., Eggers, T., Navarra-Mestre, R. (2022).** Deep learning-based segmentation of multiple species of weeds and corn crop using synthetic and real image datasets. Computers and Electronics in Agriculture, Vol. 194, pp. 106719. DOI: 10.1016/j.compag.2022.106719.

16. **Quan, L., Wu, B., Mao, S., Yang, C., Li, H. (2021).** An instance segmentation-based method to obtain the leaf age and plant centre of weeds in complex field environments. Sensors, Vol. 21, No. 10, pp. 3389. DOI: 10.3390/s21103389.

17. **Ronneberger, O., Fischer, P., Brox, T. (2015).** U-Net: Convolutional networks for biomedical image segmentation. Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

18. **Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L. C. (2018).** MobileNetV2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510–4520. DOI: 10.1109/cvpr.2018.00474.

19. **Wang, A., Zhang, W., Wei, X. (2019).** A review on weed detection using ground-based machine vision and image preprocessing techniques. Computers and Electronics in Agriculture, Vol. 158, pp. 226–240. DOI: 10.1016/j.compag.2019.02.005.

Article received on 04/07/2023; accepted on 13/10/2023.
** Corresponding author is Luis M. Valentín-Coronado.*

ISSN 2007-9737

Identificación automática de contenido misógino en redes sociales: Un enfoque basado en transferencia de conocimiento proveniente de canciones

Ricardo Calderón-Suarez[1,3], Rosa María Ortega-Mendoza[1],
Marco Antonio Márquez-Vera[2], Félix Agustín Castro-Espinoza*,[1]

[1] Universidad Autónoma del Estado de Hidalgo,
Hidalgo,
México

[2] Universidad Politécnica de Pachuca,
Hidalgo,
México

[3] Universidad Politécnica de Tulancingo,
Hidalgo,
México

{ricardo_calderon, rosa.ortega, fcastro}@uaeh.edu.mx
marquez@upp.edu.mx,

Resumen. Este artículo de investigación presenta un resumen de la tesis “Detección Automática de Contenido Misógino en Redes Sociales mediante Transferencia de Conocimiento proveniente de Canciones”, donde la idea principal es aprovechar el conocimiento existente en algunas canciones para transferir patrones lingüísticos que ayuden a identificar manifestaciones de misoginia en las redes sociales. En particular, se analizaron varias técnicas de transferencia de aprendizaje. Además, se presenta una metodología para construir, automáticamente, una colección de canciones y otra de frases, ambas con instancias etiquetadas de acuerdo con la presencia o ausencia de contenido misógino. La mayor contribución de esta investigación es un método de aumentación de datos que incrementa la capacidad de generalización de los modelos de detección de misoginia mediante la transferencia de la riqueza semántica contenida en las letras de las canciones. El enfoque propuesto fue evaluado en colecciones de referencia que contienen textos en español e Inglés, obteniendo resultados alentadores. En comparación con enfoques robustos del estado del arte, el enfoque propuesto obtuvo resultados competitivos en el idioma Inglés y ganancias importantes en el idioma Español. Esta investigación confirmó la existencia de conocimiento lingüístico valioso en las canciones, el cual puede ser transferido para detectar contenido misógino en redes sociales.

Palabras clave. Transferencia de aprendizaje, aumentación de datos, detección de misoginia, redes sociales.

Automatic Identification of Misogynistic Content on Social Networks: An Approach based on Knowledge Transfer from Songs

Abstract. This research paper presents a summary of the thesis “Automatic Detection of Misogynistic Content in Social Networks through Knowledge Transfer from Songs”, where the main idea is to leverage the existing knowledge of some songs to transfer linguistic patterns that help to identify manifestations of misogyny in social media. In particular, several learning transfer techniques were analyzed. In addition, a methodology is presented to build, automatically, a collection of songs and another of phrases, both with instances labeled according to the presence or absence of misogynistic content. The major contribution of this research is a data augmentation method that increases the generalization capability of the misogyny detection models by transferring the semantic richness contained in song lyrics. The proposed approach was evaluated in benchmark collections containing texts in Spanish and English, obtaining encouraging results. Compared to robust state-of-the-art approaches, the proposed approach obtained competitive results in English and significant gains in Spanish. This research confirmed the existence of valuable linguistic knowledge in songs, which can be transferred to detect misogynistic content in social media.

Keywords. Transfer learning, data augmentation, mysogyny detection, social media.

1. Introducción

La misoginia ha lastimado seriamente el bienestar de la sociedad y en casos severos, ha conducido a feminicidios [26]. Este concepto abarca ideas culturales que sugieren la inferioridad de las mujeres y se manifiesta a través de diversas formas como el menosprecio, la discriminación de género, acoso sexual, objetivación sexual, violencia verbal y física contra las mujeres [12]. Este comportamiento ha estado presente en la sociedad y ha evolucionado con el tiempo de acuerdo con el contexto social, cultural y religioso de los países o regiones.

Hoy en día, las manifestaciones de misoginia se observan en distintos niveles y en diversos medios de comunicación, tales como la música y las plataformas sociales (e.g., Facebook y Reddit) [21, 26, 33]. La misoginia se ha estudiado desde diversas áreas como la sociología y la psicología [18, 12]. En particular, se ha establecido una relación entre la misoginia y el lenguaje [32].

En este contexto, en el año 2016 se presentó un estudio sobre el uso del lenguaje misógino en redes sociales, encontrando hallazgos interesantes como la escritura frecuente del título de algunas canciones populares [22]. Más tarde, se publicó un trabajo pionero sobre la detección y clasificación automática de misoginia en tweets [3].

A la fecha, se han realizado varios esfuerzos para identificar automáticamente contenido ofensivo contra las mujeres en publicaciones provenientes de diversas plataformas sociales [16, 18, 24]. Recientemente, se han creado foros internacionales para evaluar métodos automáticos que abordan la tarea de identificación automática de misoginia en plataformas sociales, la cual es conocida como AMI (por sus siglas en Inglés, Automatic Misogyny Identification).

En el año 2018, dentro de los foros Ibereval [18] y Evalita [17], se lanzó una tarea compartida para identificar y clasificar mensajes misóginos escritos en Inglés, Español e Italiano de la red social Twitter[1]. En general, los enfoques participantes exploraron representaciones basadas en n-gramas y embeddings así como diversas características lingüísticas, tales como léxicas y estilísticas.

Actualmente, la tarea se ha extendido hacia la detección de contenido misógino en escenarios multimodales, implicando el procesamiento de texto e imágenes. En este contexto, en el año 2022, dentro del foro SemEval, se estableció la tarea denominada MAMI (por sus siglas en Inglés, Multimedia Automatic Mysogyny Identification) [16], la cual fue dirigida hacia la detección de misoginia en memes.

En esta competencia, el uso de modelos pre-entrenados para tratar texto e imágenes fue un factor común. La tarea AMI ha sido comúnmente abordada desde un marco de clasificación supervisada de textos.

Por lo tanto, el desempeño de los clasificadores depende en gran medida del tamaño, así como de la calidad de los conjuntos de datos de entrenamiento. Sin embargo, actualmente esta tarea se enfrenta a la escasez de colecciones de datos de entrenamiento etiquetados.

Además, resulta complicado encontrar conjuntos que contengan todo el vocabulario que exprese actitudes misóginas, tanto de forma implícita como explícita. Por ejemplo, la detección del lenguaje abusivo explícito se enfrenta a diferentes desafíos, como la identificación de vocabulario informal o jerga, así como la diversidad de significados de algunos

[1] Actualmente, esta plataforma ha cambiado su nombre a X.

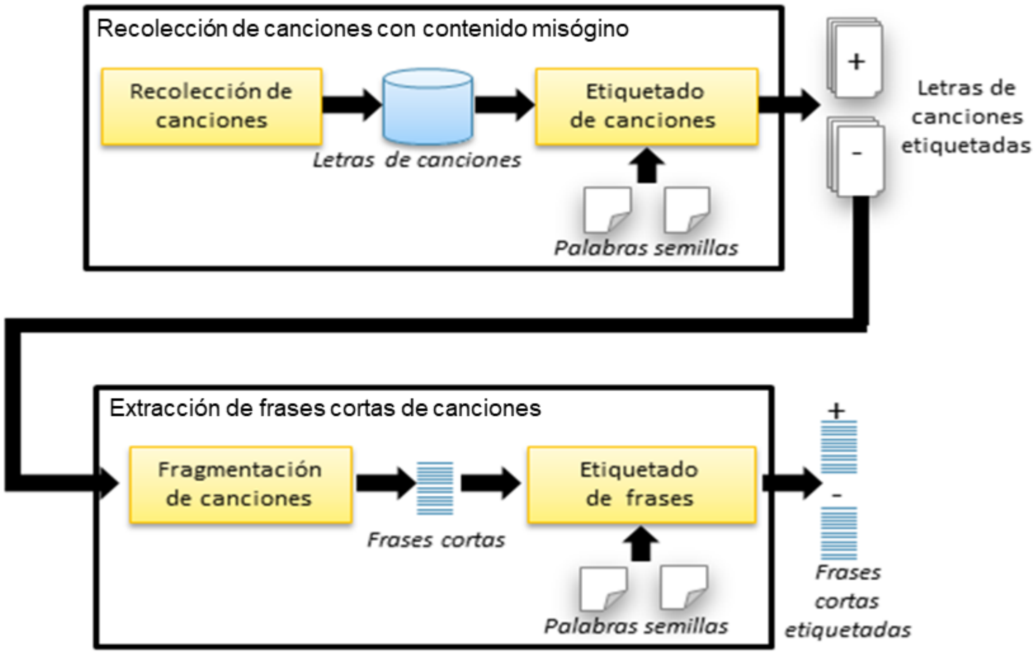

Recolección de canciones con contenido misógino
Recolección de canciones
Letras de canciones
Etiquetado de canciones
Palabras semillas
+
-
Letras de canciones etiquetadas
Extracción de frases cortas de canciones
Fragmentación de canciones
Frases cortas
Etiquetado de frases
Palabras semillas
+
-
Frases cortas etiquetadas

Fig. 1. Metodología propuesta para la construcción de colecciones de canciones y frases etiquetadas según la presencia de contenido misógino

términos relevantes (e.g., ofensas y blasfemias) dependiendo del contexto donde son usados. Por otro lado, identificar el lenguaje abusivo implícito también presenta dificultades, ya que las agresiones pueden estar ocultas o disfrazadas a través de chistes, bromas o comentarios sarcásticos [30, 34]. Tratando de enfrentar los desafíos mencionados, en este trabajo de investigación se propone enriquecer la capacidad de generalización de los modelos de detección de misoginia.

En particular, se desarrolló un enfoque para identificar manifestaciones de misoginia en redes sociales mediante la transferencia de conocimiento proveniente de otro dominio, específicamente de las letras de canciones.

El objetivo es agregar nueva información para enriquecer la diversidad de los patrones lingüísticos encontrados durante el entrenamiento. Las siguientes preguntas de investigación motivaron el trabajo: ¿Las canciones contienen patrones lingüísticos que pueden ser explotados por modelos de detección de misoginia en el ámbito de las redes sociales?, ¿Las frases de canciones pueden ser aprovechadas para aumentar los datos de entrenamiento? y ¿El enfoque propuesto puede ser adaptado para detectar misoginia en contextos multimodales?

Con el propósito de investigar las respuestas, se diseñó un enfoque que automáticamente detecta contenido misógino en redes sociales, empleando transferencia de conocimiento derivado de letras de canciones. El resto del manuscrito resume la investigación de la tesis [9] y las publicaciones derivadas [10].

2. Construcción automática de los conjuntos de datos: etiquetando canciones

La música se ha considerado como un medio de comunicación masiva en el que se transmiten ideas, sentimientos y emociones [7, 14, 19]. En este sentido, se ha encontrado que la música está vinculada al contexto donde se produce, lo que motivado diferentes investigaciones.

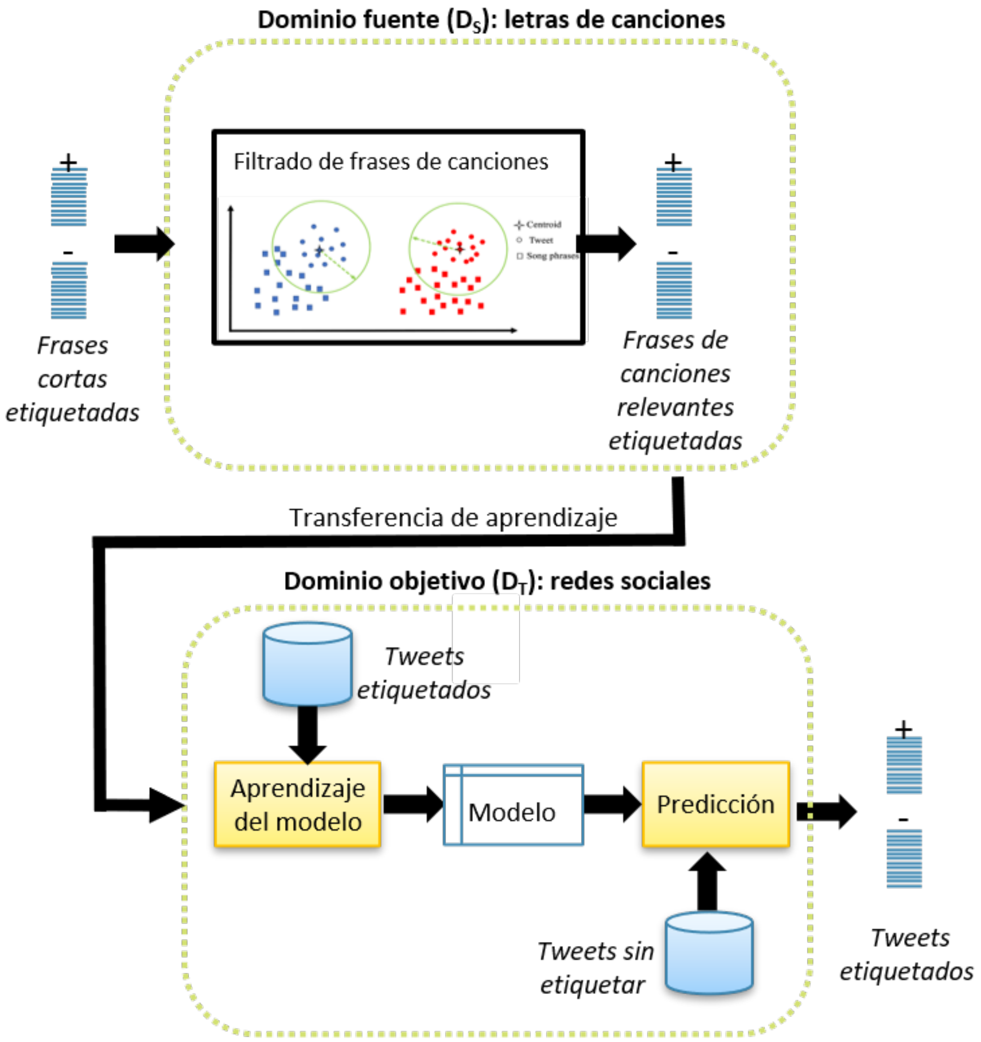

Dominio fuente (D_S): letras de canciones
+
-
Frases cortas etiquetadas
Filtrado de frases de canciones
Centroid
Tweet
Song phrases
+
-
Frases de canciones relevantes etiquetadas
Transferencia de aprendizaje
Dominio objetivo (D_T): redes sociales
Tweets etiquetados
Aprendizaje del modelo
Modelo
Predicción
Tweets sin etiquetar
+
-
Tweets etiquetados

Fig. 2. Esquema del enfoque propuesto para aumentación de datos

Por ejemplo, algunos hallazgos indican que existen letras de canciones populares que expresan una representación desfavorable de las mujeres [4, 6], mostrando algunos fenómenos arraigados en la sociedad, tales como: objetivación sexual de las mujeres [33], inferioridad femenina e incluso violencia contra las mujeres [1, 8].

Por lo tanto, el contenido de las canciones y el uso del lenguaje en tal dominio pueden constituir una fuente valiosa de conocimiento para detectar, de manera automática, comportamiento agresivo contra las mujeres.

Para explorar y aprovechar esta base de conocimientos, en esta investigación se diseñó una metodología que recopila y etiqueta de manera automática dos conjuntos de datos: uno compuesto por canciones completas y otro por las frases que contienen expresiones de misoginia extraídas de dichas canciones.

El proceso completo para generar las colecciones mencionadas en el párrafo anterior se muestra en la Figura 1. Como se observa, tanto las canciones como las frases son etiquetadas mediante las categorías: misógina y no misógina.

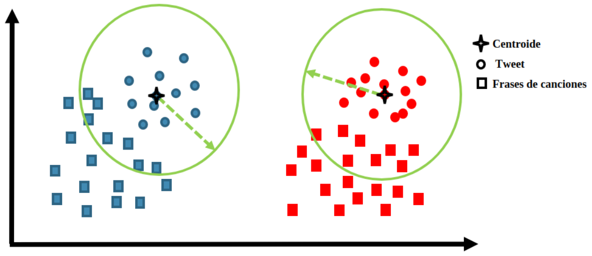


Fig. 3. Ilustración del mecanismo de filtrado propuesto. La calidad de las instancias está asociada con la distancia hacia el correspondiente centroide. Los tweets positivos y negativos se encuentran representados con círculos rojos y azules, respectivamente. Las frases de canciones están ilustradas mediante pequeños cuadrados

Tabla 1. Distribución de las colecciones generadas. Se muestran las estadísticas del conjunto de canciones (Ca) y Frases (Fr). Las etiquetas fueron asignadas automáticamente y corresponden a la categoría misógina (M) o no misógina (N)

Idioma	**Conjunto**	**M**	**N**	**Total**
Español	Ca	4228	4228	8456
	Fr	1411	1411	2822
Inglés	Ca	11086	11086	22172
	Fr	2120	2120	4240

En las siguientes secciones se describen las etapas de la metodología propuesta.

2.1. Recolección de canciones con contenido misógino

En esta etapa, se recopilan y etiquetan automáticamente canciones. Durante la recolección, se incluyeron canciones que han sido señaladas como misóginas por activistas en diversos foros web. También se añadieron otras seleccionadas aleatoriamente.

Además, con el fin de asegurar un amplio vocabulario de términos, se recolectaron composiciones de diversos autores y estilos musicales, las cuales fueron obtenidas de distintas plataformas en línea[2].

Un paso importante dentro de la metodología es etiquetar automáticamente las letras de canciones (i.e., asignar una etiqueta a cada instancia). Para ello, se diseñó un proceso automático que considera la presencia de palabras clave, a las cuales se les denomina semillas.

Específicamente, se emplearon dos tipos de semillas: palabras misóginas y palabras relacionadas con el término mujer. Las palabras misóginas son aquellas asociadas con la manifestación de abuso verbal contra la mujer.

Para esta investigación se emplearon dos léxicos de términos vinculados con misoginia, los cuales fueron presentados en [15, 27], para el idioma Inglés y Español, respectivamente. Por otro lado, las semillas relacionadas con el término mujer se utilizaron para garantizar que el contenido de las letras haga referencia a las mujeres.

En este contexto, se consideró una lista de palabras clave comúnmente relacionadas[3], tales como niña, novia y esposa. Considerando estas semillas, los criterios diseñados para definir las etiquetas de cada canción son los siguientes:

Misógina. Se asigna esta etiqueta cuando una canción contiene palabras de ambos tipos de semilla: una relacionada con mujer y al menos dos palabras vinculadas con misoginia.

No misógina. Se asigna esta etiqueta a las canciones que no contienen palabras misóginas.

2.2. Extracción de frases cortas de canciones

La segunda etapa de la metodología propuesta está orientada a extraer frases cortas de canciones y etiquetarlas con las categorías misógina o no misógina. El proceso inicial consiste en fragmentar las letras de las canciones en segmentos, cada uno de los cuales tiene una extensión máxima de 280 caracteres.

Cabe señalar que se eligió un tamaño de longitud similar a aquella de las publicaciones en la plataforma Twitter, ya que el método se enfoca en detectar misoginia en tweets.

[2] Por ejemplo: www.lyrics.com/ y www.letras.com/

[3] Fueron obtenidas consultando los siguientes portales web: relatedwords.org para la configuración en el idioma Inglés y www.ideasafines.com.ar/do-buscar.php para el idioma Español.

Tabla 2. Algunas estadísticas de los conjuntos de frases filtradas utilizando las técnicas propuestas basadas en similitud Coseno (F-Coseno) y el algoritmo de Roccio (F-Roccio)

Datos	Conjunto	Misógino	No Misógino	Total
Español	Frases	1411	1411	2822
	F-Coseno	282	282	564
	F-Roccio	290	1411	1701
Inglés	Frases	1783	1783	3566
	F-Coseno	357	357	714
	F-Roccio	1085	1776	2861

Los criterios de etiquetado de las frases siguen un procedimiento similar al etiquetado de las canciones completas. La etiqueta misógina (clase positiva) se otorga a aquellas frases provenientes de canciones etiquetadas con esta categoría y que, además, contienen dos palabras semillas vinculadas con la misoginia y una relacionada con el término mujer. En contraste, la etiqueta No misógina (clase negativa) se otorga a frases cortas elegidas aleatoriamente del conjunto de canciones no misóginas.

2.3. Resultados de la construcción de las colecciones

Las estadísticas de las colecciones resultantes del proceso de etiquetado de canciones y la extracción de frases se muestran en la Tabla 1. Como se observa, se crearon colecciones de acuerdo con el idioma en el que fueron escritas las canciones (Español o Inglés).

Como parte del análisis de las colecciones construidas, se realizó una exploración de su vocabulario. En general, se observó que los términos más frecuentes incluyen, además de los términos semilla, palabras despectivas u ofensivas contra las mujeres.

También, se observaron referencias a diversas partes del cuerpo, la cuales son comúnmente relacionadas con el concepto de cosificación sexual. En general, el análisis indica que los fragmentos de canciones que contienen palabras semilla muestran manifestaciones de misoginia.

Este conocimiento lingüístico puede ser relevante para diversas tareas basadas en clasificación automática, por ejemplo, la detección de misoginia.

3. Un nuevo enfoque de aumentación de datos usando frases de canciones

La tarea AMI suele abordarse bajo un esquema de clasificación de textos. En este enfoque, la calidad de los clasificadores se relaciona con su capacidad para generalizar, la cual, a su vez, depende de la cantidad de datos de entrenamiento. Sin embargo, esta tarea se ha enfrentado a la poca disponibilidad de datos etiquetados para entrenar los modelos computacionales.

Actualmente, una de las soluciones al problema de escasez de datos de entrenamiento etiquetados contempla el uso de técnicas de aumentación de datos [23]. En este trabajo de investigación se propone un enfoque de aumentación de datos que aprovecha el conocimiento y patrones lingüísticos provenientes de las canciones. La Figura 2 muestra una visión general del método propuesto.

Su objetivo es utilizar frases de alta calidad de las canciones para aumentar los conjuntos de entrenamiento relacionados con la tarea. La idea clave es incrementar la capacidad de aprendizaje de los modelos diversificando las instancias de entrenamiento con ejemplos de expresiones socioculturales contenidas en la música.

3.1. Transfiriendo conocimiento proveniente de las canciones

El enfoque propuesto se sitúa en el dominio de los enfoques de transferencia de aprendizaje, ya que aprovecha el conocimiento existente en las canciones para utilizarlo en una tarea fuera del dominio (out-domain). En concordancia con las notaciones en [2], el concepto de transferencia de aprendizaje se define enseguida. Sean D_S datos del dominio fuente, D_T datos del dominio objetivo o destino, T_S la tarea de aprendizaje del dominio fuente y T_T representa la tarea de aprendizaje en el dominio objetivo.

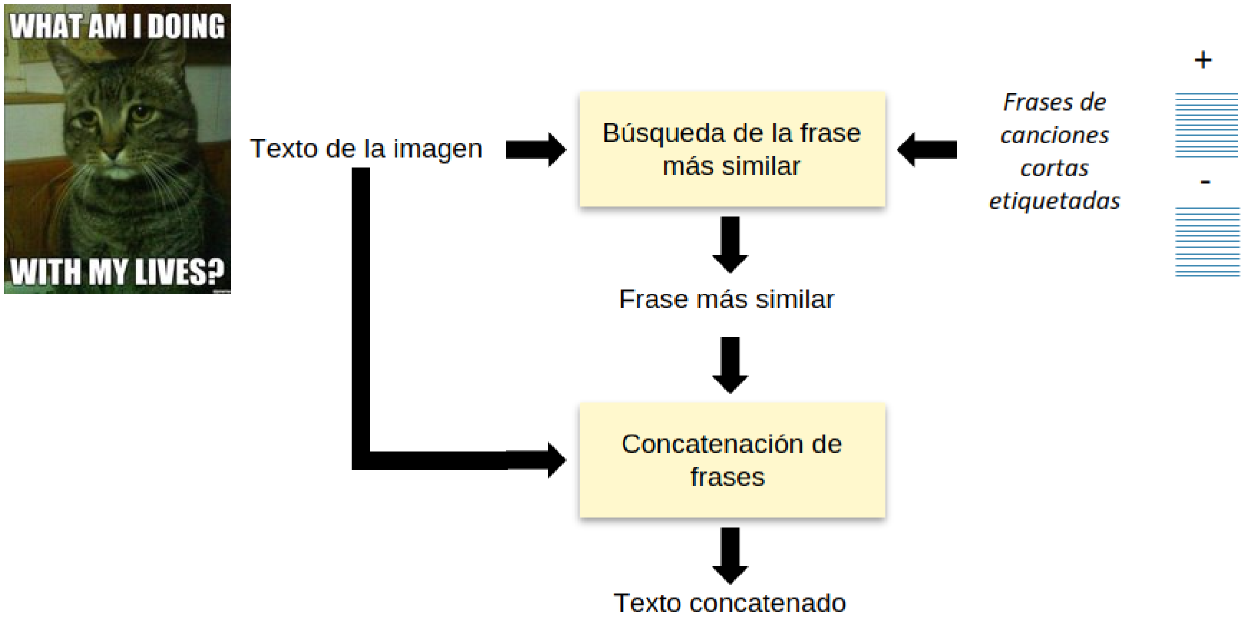


Fig. 4. Representación de la técnica propuesta para expandir el texto contenido en los memes. La imagen fue extraída del conjunto de entrenamiento del concurso SemEval 2022 (tarea MAMI)

La meta del aprendizaje por transferencia es emplear el conocimiento del dominio fuente y su tarea asociada en el proceso de aprendizaje para la tarea del dominio objetivo, donde $D_S \neq D_T$ o $T_s \neq T_T$.

Para este trabajo de investigación, las letras de canciones y las redes sociales son consideradas los dominios D_S y D_T, respectivamente, mientras $T_S = T_T$ (i.e., detección de misoginia).

El método propuesto aumenta los datos de entrenamiento del D_T siguiendo un enfoque de dominio cruzado. El objetivo es agregar únicamente frases de calidad que aporten en la mejora del desempeño del clasificador. Para lograr esto, se diseñó un mecanismo de filtrado, el cual es descrito en la siguiente sección.

3.2. Mecanismo de filtrado

La variedad de temas presentes en las canciones puede generar oraciones que no contribuyen al proceso de generalización, añadiendo ruido y afectando el rendimiento del clasificador.

Para solventar este problema, se propone un mecanismo que evalúa la calidad de las frases y selecciona solamente aquellas más pertinentes para la tarea (es decir, las de mayor calidad). En particular, se diseñó un filtro basado en la similitud de las frases de canciones con el conjunto de entrenamiento formado por los tweets.

La Figura 3 esquematiza el filtro propuesto. Su propósito consiste en seleccionar únicamente las instancias del dominio de fuente que estén más próximas a los centroides de cada clase existente en el dominio destino.

Para cuantificar la proximidad existente entre los centroides de los tweets y las frases de las canciones, se empleó la similitud Coseno. En este sentido, se sugiere retener únicamente un porcentaje (θ) de las frases con mayor similitud.

Cabe señalar que, también se empleó el clasificador Roccio[4] como una estrategia alternativa al uso de la similitud Coseno.

[4] scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestCentroid.html

Tabla 3. Desempeño de una BoW aplicando diversas configuraciones de adaptación de dominio. Los resultados se reportan en términos de exactitud (Exac) y F_1. El método de referencia corresponde a un clasificador entrenado únicamente con los tweets (Tw). La información de las letras se consideró en el entrenamiento a partir de las dos colecciones construidas: letras de canciones completas (LetraC) y frases de canciones (Frases)

Datos		**MVS**		**XG**		**RL**	
		Exac	**F1**	**Exac**	**F1**	**Exac**	**F1**
Iber-Es	Tw	**0.819**	**0.819**	0.781	0.780	0.813	0.813
	LetraC	0.659	0.646	0.656	0.648	0.651	0.629
	Frases	0.665	0.656	0.657	0.649	0.649	0.641
Iber-In	Tw	**0.806**	0.793	0.798	0.772	0.762	0.723
	LetraC	0.665	0.591	0.675	0.574	0.678	0.601
	Frases	0.686	0.633	0.669	0.574	0.697	0.654
Eval-In	Tw	0.597	0.597	0.567	0.562	0.606	0.602
	LetraC	0.638	0.634	0.642	0.627	**0.644**	0.636
	Frases	0.615	0.614	0.641	0.633	0.623	0.622

3.3. Colecciones filtradas: Resultados del proceso de filtrado

Los conjuntos conformados por las instancias filtradas se muestran en la Tabla 2. La colección generada usando el filtro basado en similitud Coseno fue intencionalmente balanceada con respecto al número de instancias de la clase positiva. Los conjuntos de frases filtradas serán utilizadas para aumentar los datos del entrenamiento.

4. Adaptación para detectar misoginia en ambientes multimodales

Como parte de la investigación, el enfoque propuesto fue adaptado para detectar misoginia en ambientes multimodales. La adaptación propuesta fue evaluada con datos la tarea MAMI del foro Semeval 2022 [16], cuyo objetivo fue detectar memes con contenido misógino.

La idea principal de la adaptación del enfoque es transferir el conocimiento de las canciones hacia la clasificación de memes, los cuales pueden incluir texto. Considerando que el texto de los memes presenta, comúnmente, una longitud corta, se propone expandirlo añadiendo frases muy similares provenientes de las canciones.

De esta manera, se agrega nueva información que puede enriquecer los patrones lingüísticos discriminativos para la tarea. El procedimiento de expansión se encuentra representado en la Figura 4.

Como se observa, antes de ingresar los memes al clasificador, pasan por el proceso de expansión. Posteriormente, las instancias son clasificadas usando una arquitectura multimodal. Particularmente, para los experimentos, se utilizó el modelo presentado en [31].

5. Configuración experimental

En las siguientes secciones se muestra el marco de trabajo experimental: conjuntos de datos usados, representaciones textuales y clasificadores.

5.1. Conjuntos de datos

Las ideas de este trabajo de investigación fueron evaluadas usando los conjuntos de datos provenientes de los foros: Ibereval [18] y Evalita [17]. Particularmente, se realizaron experimentos con los datos en Español e Inglés del primer conjunto, mientras que de Evalita se utilizaron los datos en Español.

Tabla 4. Comparación de desempeño (exactitud) de diferentes modelos de clasificación empleando los embedings generales (Gen) y especializados (Esp). Los vectores de palabras fueron evaluados a través de dos representaciones de texto: vector promedio (AWE) y como capa de entrada de un modelo GRU

Conjunto	**Tipo**	**AWE**			**GRU**
		MVS	**XG**	**RL**	**Prom±DE**
Iber-Es	Gen	0.776	0.781	0.762	0.779 ± 0.012
	Esp	**0.788**	0.771	0.777	**0.792±0.018**
Iber-In	Gen	0.751	0.731	0.729	0.729 ± 0.029
	Esp	**0.792**	0.758	0.791	0.742 ± 0.022
Eval-In	Gen	0.641	0.625	**0.665**	0.576 ± 0.018
	Esp	0.615	0.617	0.618	0.588 ± 0.026

En las siguientes secciones se hará referencia a ellos a través de la siguiente notación: Iber-Es, Iber-In y Eval-In, respectivamente. Para el escenario multimodal se utilizó el conjunto de datos proveniente de la tarea Mami en el foro SemEval 2022 [16].

5.2. Representaciones textuales

Pre-procesamiento. El texto de los tweets fue procesado como sigue: conversión a minúsculas, eliminación de palabras vacías, así como caracteres especiales, emojis, URL's y las menciones a usuarios. Además, el texto fue separado en unigramas de palabras y para crear las representaciones textuales se usaron los 10,000 términos más frecuentes.

Bolsa de palabras BoW. Como método de referencia, se utilizó una tradicional bolsa de palabras (BoW por sus siglas en Inglés, Bag of Words). El esquema de pesado de términos utilizado corresponde a la frecuencia normalizada.

Vectores de palabras (word embeddings). Para explorar el uso de word embeddings, se empleó una representación basada en el promedio de vectores de palabras (AWE, por sus siglas en Inglés, Average Word Embeddings). Específicamente, cada tweet es representado por el promedio de los vectores de sus palabras. Para este propósito, se entrenaron word embeddings de 300 dimensiones usando la colección de letras misóginas a través del modelo Skip-gram.

En los experimentos se hace referencia a ellos como word embeddings especializados, ya que fueron generados específicamente para la tarea. Para fines de comparación, también se usaron embeddings generales previamente entrenados[5].

También se empleó una Unidad recurrente cerrada, GRU (por sus siglas en Inglés, (Gated Recurrent Unit) con una capa de atención. En este caso, los embeddings (especializados o generales) se utilizaron como la primera capa en el modelo.

Modelos del lenguaje pre-entrenados. Para los experimentos en Inglés y Español, se emplearon los modelos pre-entrenados distilbert-base-uncased [29] y BETO [11], respectivamente. Para la tarea multimodal, la representación textual de las frases se obtuvo aplicando Sentence-BERT [28] mediante el modelo all-MiniLM-L12-v1.

Todos los modelos fueron obtenidos de la librería de hugging-transformers[6]. También, se consideró un tamaño de lote (batch size) de 16 y la estrategia early stopping.

5.3. Clasificación y evaluación

Durante el proceso experimental, se utilizaron diferentes algoritmos de aprendizaje automático: Máquina de Vectores de Soporte (MVS), XGBoost (XG) [13] y Regresión Logística (RL). Como medidas de desempeño se reportaron la exactitud

[5] fasttext.cc/docs/en/pretrained-vectors.html

[6] huggingface.co/docs/transformers/index

Tabla 5. Evaluación de diferentes configuraciones en el proceso de filtrado: sin aumentación de datos (No), con aumentación de frases de canciones positivas y negativas, denotada como Frases, con frases provenientes de las técnicas de filtrado con instancias positivas (+), así como con instancias positivas y negativas (+)(-). Se reportan los valores de exactitud promedio, mínimo y máximo

Datos	Aumentación	Prom ± DE	Min	Max
Iber-Es	No	0.829 ± 0.015	0.810	0.854
	Frases	0.841 ± 0.011	0.826	0.859
	Coseno (+)	**0.851** ±0.005	0.845	**0.859**
	Coseno (+) (-)	0.839 ± 0.010	0.828	0.857
	Roccio (+)	0.846 ± 0.004	0.846	0.856
	Roccio (+) (-)	0.844 ± 0.005	0.835	0.852
Iber-In	No	0.836 ± 0.010	0.822	0.853
	Frases	0.860 ± 0.016	0.844	0.886
	Coseno (+)	0.825 ± 0.013	0.810	0.842
	Coseno (+) (-)	0.861 ± 0.019	0.835	0.888
	Roccio (+)	0.843 ± 0.030	0.803	0.868
	Roccio (+) (-)	**0.892** ± 0.009	0.883	**0.906**
Eval-In	No	0.644 ± 0.018	0.617	0.671
	Frases	0.684 ± 0.010	0.666	0.694
	Coseno (+)	0.682 ± 0.016	0.658	0.705
	Coseno (+) (-)	**0.686** ± 0.016	0.663	**0.705**
	Roccio (+)	0.652 ± 0.004	0.645	0.656
	Roccio (+) (-)	0.666 ± 0.006	0.659	0.677

(Exac) y los valores F1. Los experimentos basados en el uso de redes neuronales y modelos pre-entrenados se realizaron cinco veces y se reportó el resultado promedio en la partición de prueba, así como la desviación estándar $(\mathrm{Prom} \pm \mathrm{DE})$.

6. Experimentos y resultados

En esta sección se reportan los experimentos y resultados que validan las ideas del método propuesto.

6.1. Evaluación de técnicas tradicionales

Como parte del trabajo experimental, se evaluaron dos técnicas comúnmente utilizadas para transferir conocimiento entre dominios: Adaptación de dominio y el uso de embeddings especializados (generados a partir de las letras de canciones etiquetadas como misóginas).

6.1.1. Adaptación de dominio

En términos generales, la adaptación de dominio (DA) busca entrenar un clasificador en un dominio y probarlo en otro que tiene una distribución de datos diferente [2].

En este sentido, el objetivo de este experimento es determinar la pertinencia de transferir el conocimiento lingüístico de las canciones como marcador de misoginia en el dominio de las redes sociales.

Para alcanzar este objetivo, se entrenaron clasificadores utilizando el contenido de las canciones, es decir, el dominio fuente, para clasificar instancias de conjuntos de datos de redes sociales, es decir, el dominio objetivo.

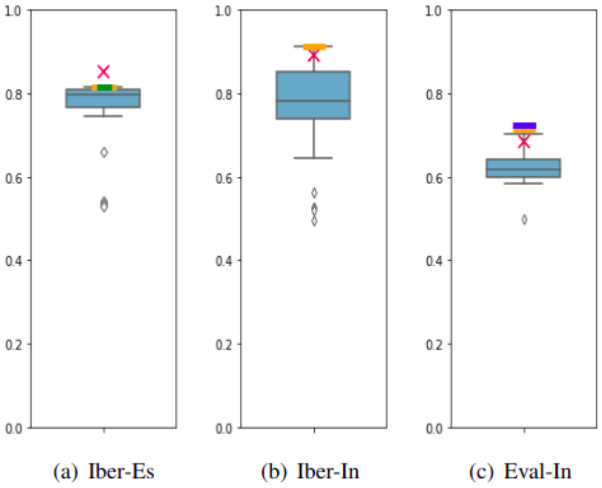


Fig. 5. Resultados en las respectivas tareas compartidas. Se ilustra la distribución de los valores de exactitud obtenidos en los conjuntos a) IberEval Español, b) IberEval Inglés, y c) Evalita Inglés. Las marcas de cruz en color rojo muestran el rendimiento alcanzado por la técnica propuesta

Particularmente, se entrenaron tres clasificadores mediante una BoW tradicional construida con las colecciones de datos: las letras completas de las canciones (denotada como LetraC) o con las frases de canciones etiquetadas (denotada como Frases). Como método de referencia se presentan los resultados de entrenar los clasificadores usando tweets en el entrenamiento y prueba (Tw).

La Tabla 3 presenta los resultados de este experimento. Se observa que el rendimiento de los clasificadores entrenados con las canciones completas o con las frases no superaron los resultados del método de referencia en las primeras dos colecciones.

Sin embargo, sí obtuvieron un mejor desempeño que un clasificador aleatorio en una tarea binaria, donde los resultados promedio se acercan al 50 %. Estos resultados indican la presencia de un subconjunto común de características en ambos dominios que tienen un valor relevante para detectar misoginia.

Además, de forma general, se observó que usando únicamente las frases de canciones se obtuvieron mejores resultados que empleando todas las canciones para el entrenamiento.

Este desempeño es notable en las colecciones Iber-Es e Iber-In. Estos resultados sugieren que los patrones lingüísticos que detectan misoginia están concentrados en algunas frases de las canciones y no en toda la letra. En general, los hallazgos encontrados pueden ser aprovechados para enriquecer enfoques y representaciones textuales más robustas.

6.1.2. Evaluación de embeddings

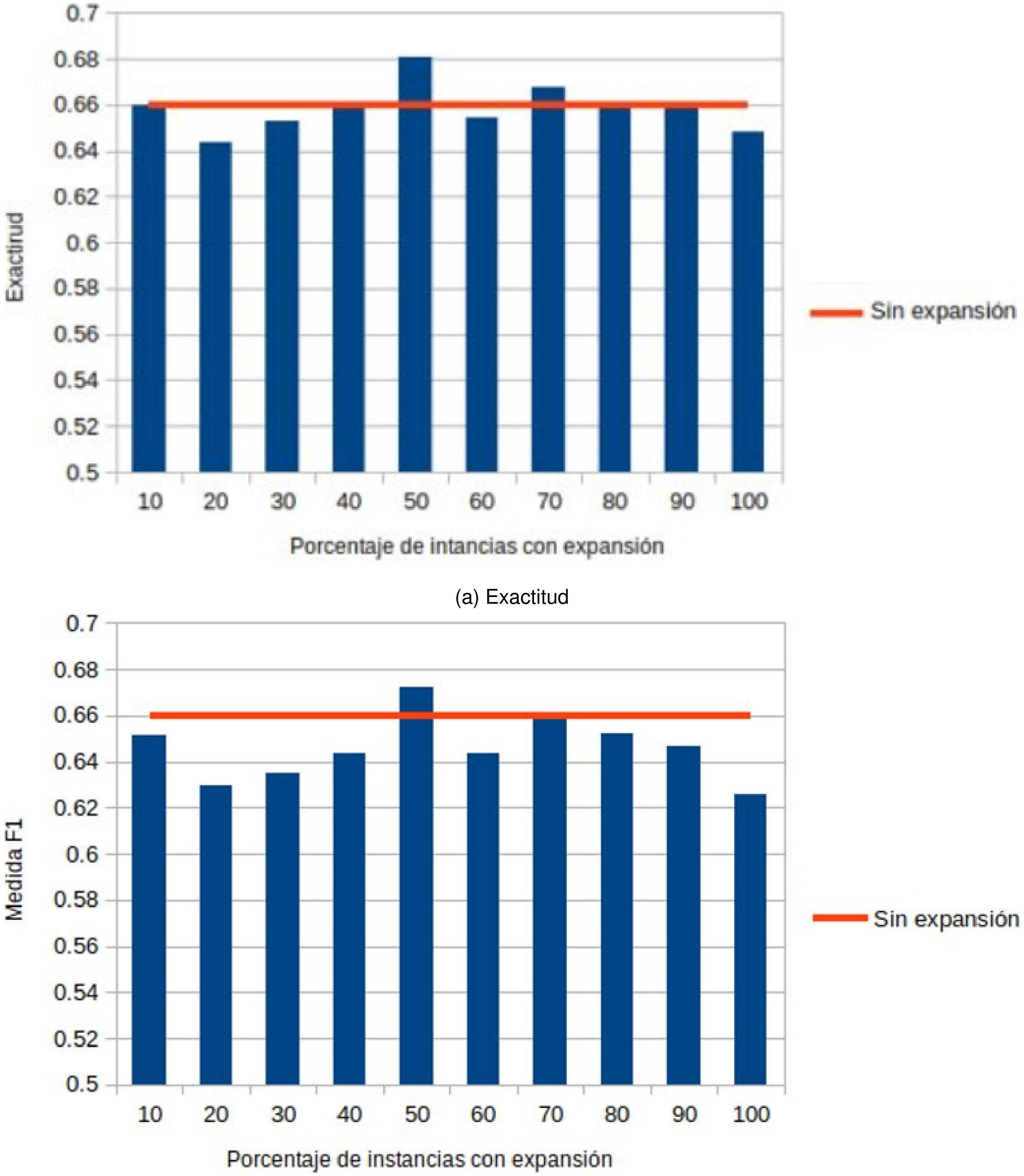
El objetivo de este experimento es evaluar métodos basados en word embeddings para transferir conocimiento de las letras de canciones hacia la tarea AMI. El interés es comparar el uso de embeddings especializados contra el uso de embeddings generales.

Los word embeddings especializados fueron aprendidos de las letras de las canciones que contienen contenido misógino explícito, por lo tanto, se generaron a partir de las canciones etiquetadas como misóginas. Mientras, los vectores generales corresponden a embeddings pre-entrenados de FastText.

Para llevar a cabo la comparación, se evaluaron dos representaciones basadas: i) el promedio de los embeddings individuales (AVE) clasificados por diferentes algoritmos de aprendizaje automático y ii) una arquitectura GRU alimentada por embeddings generales o particulares. La tabla 4 muestra los resultados de la comparación.

Los resultados mostraron que en la mayoría de los casos, los embeddings especializados se desempeñaron mejor que los embeddings generales, independientemente de la representación utilizada. No obstante, los resultados no superan el desempeño del BoW mostrada en el experimento anterior.

Este comportamiento indica que la representación textual es muy importante para aprovechar el conocimiento existente en las canciones y lograr transferirlo a la tarea AMI. Por lo tanto, es importante explorar nuevas representaciones que aprovechen el conocimiento contenido en las canciones.



(a) Exactitud

(b) Valores F1

Fig. 6. Evaluación del método de expansión en la tarea de clasificación de memes. Se muestran los resultados obtenidos al aplicar la expansión del texto a diferentes porcentajes de instancias dentro de la colección de datos. Los resultados sin expansión son mostrados como método de referencia (línea roja)

6.2. Evaluación del enfoque propuesto para aumentación de datos

Los experimentos previos mostraron que el conocimiento de las canciones puede ayudar a detectar mensajes misóginos en las redes sociales. Esta sección está enfocada a evaluar el método de aumentación de datos propuesto, el cual tiene como objetivo diversificar los datos de entrenamiento en el dominio de destino (tweets) añadiendo ejemplos del dominio fuente (frases de canciones).

Dado que los modelos lingüísticos pre-entrenados (e.g., BERT) han mostrado resultados significativos en diferentes tareas de clasificación de textos, en este experimento se evaluaron modelos basados en BERT.

Particularmente, se utilizaron DistilBERT y Beto para los experimentos en Inglés y Español, respectivamente. Para fines de comparación, se evaluaron los mismos modelos con diferentes configuraciones en el entrenamiento, como se describe a continuación: sin aumentación de datos (No), con aumentación de frases originadas de canciones positivas y negativas (Frases), con frases provenientes de las técnicas de filtrado (filtrado basado en Similitud Coseno o basado el método Roccio) con instancias positivas (+) o con instancias positivas y negativas (+)(-). La Tabla 5 muestra los resultados del enfoque con las diversas configuraciones.

En general, los resultados muestran que todas las configuraciones de aumentación de datos obtuvieron mejor desempeño en comparación con los casos en los que no se aplicó aumentación. Por lo tanto, se concluye que las frases de las canciones son útiles para aumentar los datos de entrenamiento.

Además, es importante notar que el mejor resultado en cada conjunto de datos siempre fue obtenido con la técnica de aumentación de datos propuesta involucrando alguno de los mecanismos de filtrado (Coseno o Roccio).

Esto demuestra la utilidad de las frases de las canciones, pero sugiere una ventaja aún mayor cuando se utilizan únicamente aquellas frases de mayor calidad para la tarea.

Comparación con el estado del arte. En la Figura 5, el enfoque propuesto fue comparado con métodos del estado del arte para evaluar su competitividad. Primero, se contrastó con la distribución de resultados oficiales alcanzados en las tareas compartidas donde se han utilizado los conjuntos de datos empleados en este estudio.

Para fines de comparación, el desempeño del método de aumentación de datos propuesto se representa con cruces rojas y corresponde a las configuraciones que obtuvieron resultados los mejores resultados en cada conjunto de datos. Se pueden distinguir resultados competitivos en relación con los equipos participantes.

Es importante señalar que en el conjunto de datos Español, el método propuesto superó el desempeño del equipo ganador de la tarea compartida IberEval [18].

Sin embargo, en el conjunto de datos en Inglés, el rendimiento estuvo por debajo del ganador, lo cual ubicaría al método propuesto en el cuarto lugar de la competencia.

Por otro lado, en Evalita [17], los resultados obtenidos estuvieron ligeramente por debajo del primer lugar. El método propuesto también fue comparado con enfoques recientes y robustos del estado del arte que han usado los mismos conjuntos de datos.

En específico, en la figura, las marcas de color naranja corresponden a los resultados de un enfoque de clasificación de dominios cruzados que utiliza conjuntos de datos de diversas tareas relacionadas con el lenguaje abusivo, como discursos de odio y sexismo [25], empleando elementos del texto como emojis y clasificadores como MVS, GRU y BERT.

También se comparó con el rendimiento de un método de transferencia Bayesiano basado en una LSTM, el cual fue denotado con un guion azul [5]. Finalmente, se contrastó con resultados obtenidos por un método que usa una combinación de incrustaciones de palabras y características lingüísticas, el cual está representado con un guion verde [20]. En general, se demostró la competitividad del método frente a estos enfoques del estado del arte, resaltando su desempeño sobresaliente en el idioma Español.

6.3. Evaluación de la técnica de expansión de texto: detección de misoginia en memes

Este experimento está enfocado a abordar la tarea MAMI, una tarea multimodal enfocada a clasificar memes según la presencia de contenido misógino. En específico, el objetivo de este experimento es evaluar el impacto de la técnica de expansión del texto, la cual fue descrita en la Sección 4.

La idea de la técnica es expandir el texto de los memes con frases similares de canciones. En los experimentos, el texto fue expandido con la frase más similar dentro de la colección de frases etiquetadas generada en este trabajo de investigación. Una vez que el texto de las instancias es expandido, se entrena un clasificador multimodal.

Los resultados de este clasificador se muestran en las Figuras 6.a y 6.b, las cuales reportan los valores de exactitud y F1, respectivamente. Además, como método de referencia se muestra el desempeño obtenido por el modelo sin expansión del texto (línea roja en la figura).

Por otro lado, para analizar el comportamiento de acuerdo con el número de instancias en las cuales el texto ha sido expandido, se presentan los resultados obtenidos al aplicar la expansión en diferentes porcentajes de las instancias de entrenamiento. La figura muestra que el rendimiento del clasificador es mejorado cuando la expansión se realiza en el 50 % de las instancias del entrenamiento.

Estos resultados sugieren que el lenguaje de las canciones puede aumentar la diversidad lingüística de las expresiones textuales existentes en los memes. Sin embargo, es importante profundizar en el diseño de arquitecturas que tomen ventaja de estos hallazgos.

7. Conclusiones y trabajo futuro

En esta investigación se examinó la relevancia de las letras de las canciones para modelar manifestaciones de misoginia y transferir el conocimiento hacia la tarea de identificar de misoginia en redes sociales.

La idea que impulsó la investigación es la difusión de la ideología de género expuesta en una variedad de canciones, reflejando creencias socioculturales.

En particular, en este trabajo se propuso un método de aumentación de datos que aumenta la capacidad de generalización de los modelos de aprendizaje a través del uso de conocimiento proveniente de las canciones.

El enfoque fue evaluado en colecciones compuestas de publicaciones de redes sociales en el idioma Español e Inglés.

Los resultados experimentales mostraron que algunas canciones contienen patrones lingüísticos que reflejan manifestaciones misóginas y que este conocimiento puede ser transferido para detectar misoginia en contenido publicado en redes sociales.

Sin embargo, la riqueza de las canciones para este propósito se concentra únicamente en algunos fragmentos y no en toda la letra. Particularmente, los fragmentos relevantes pueden ser aprovechados para aumentar los datos de entrenamiento de la tarea a través de una evaluación de su calidad.

En este contexto, el enfoque propuesto superó los resultados del Estado del Arte en el idioma Español. El método puede ser adaptado para trabajar en escenarios de detección de misoginia multimodal mediante la expansión de los textos cortos con frases similares, brindando mayor información a los modelos computacionales.

Los resultados de esta investigación han motivado el interés de adaptar el método para trabajar con otros idiomas, por ejemplo, Italiano. Además, se planea diseñar arquitecturas y estrategias multimodales robustas que aprovechen el conocimiento de las letras de canciones.

Agradecimientos

Los autores agradecen al Consejo Nacional de Humanidades, Ciencias y Tecnologías de México (CONAHCYT) por financiar esta investigación mediante una beca para estudios de posgrado (CVU-714747).

Referencias

1. **Adams, T. M., Fuller, D. B. (2006).** The words have changed but the ideology remains the same: Misogynistic lyrics in rap music. Journal of Black Studies, Vol. 36, No. 6, pp. 938–957. DOI: 10.1177/0021934704274072.

2. **Alyafeai, Z., AlShaibani, M. S., Ahmad, I. (2019).** A survey on transfer learning in natural language processing. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials, pp. 15–18. DOI: 10.18653/v1/N19-5004.

3. **Anzovino, M., Fersini, E., Rosso, P. (2018).** Automatic identification and classification of misogynistic language on twitter. Proceedings of the International Conference on Applications of Natural Language Processing and Information Systems, Vol. 10859, pp. 57–64. DOI: 10.1007/978-3-319-91947-8_6.

4. **Barton, G. (2018).** The relationship between music, culture, and society: Meaning in music. Music Learning and Teaching in Culturally and Socially Diverse Contexts: Implications for Classroom Practice, pp. 23–41. DOI: 10.1007/978-3-319-95408-0_2.

5. **Bashar, M. A., Nayak, R., Suzor, N. (2020).** Regularising LSTM classifier by transfer learning for detecting misogynistic tweets with small training set. Knowledge and Information Systems, Vol. 62, No. 10, pp. 4029–4054. DOI: 10.1007/s10115-020-01481-0.

6. **Behm-Morawitz, E., Frisby, C. M. (2019).** Undressing the words: Prevalence of profanity, misogyny, violence, and gender role references in popular music from 2006-2016. Journal of Communication: Media Watch, Vol. 10, No. 1, pp. 5–21. DOI: 10.17613/7Y4W-JM88.

7. **Bicknell, J. (2002).** Can music convey semantic content? a Kantian approach. The Journal of Aesthetics and Art Criticism, Vol. 60, No. 3, pp. 253–261.

8. **Brook, B., Schindler-Zimmerman, T., Banning, J. H. (2008).** A feminist analysis of popular music. Journal of Feminist Family Therapy, Vol. 4, No. 18, pp. 29–51. DOI: 10.1300/J086v18n04_02.

9. **Calderón-Suarez, R. (2023).** Detección automática de contenido misógino en redes sociales mediante transferencia de conocimiento proveniente de canciones. Ph.D. thesis, Universidad Politécnica de Tulancingo.

10. **Calderón-Suarez, R., Ortega-Mendoza, R. M., Montes-Y-Gómez, M., Toxqui-Quitl, C., Márquez-Vera, M. A. (2023).** Enhancing the detection of misogynistic content in social media by transferring knowledge from song phrases. IEEE Access, Vol. 11, pp. 13179–13190. DOI: 10.1109/ACCESS.2023.3242965.

11. **Canete, J., Chaperon, G., Fuentes, R., Ho, J. H., Kang, H., Pérez, J. (2020).** Spanish pre-trained BERT model and evaluation data. Vol. 2020, pp. 1–10. DOI: 10.48550/arXiv.2308.02976.

12. **Chaudhury, S., Srivastava, K., Bhat, P., Sahu, S. (2017).** Misogyny, feminism, and sexual harassment. Industrial Psychiatry Journal, Vol. 26, No. 2, pp. 111. DOI: 10.4103/ipj.ipj_32_18.

13. **Chen, T., Guestrin, C. (2016).** XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM Special Interest Group on Knowledge Discovery and Data Mining International Conference on Knowledge Discovery and Data Mining, pp. 785–794. DOI: 10.1145/2939672.2939785.

14. **Cooke, D. (1959).** The language of music. Oxford University Press.

15. **Farrell, T., Fernandez, M., Novotny, J., Alani, H. (2019).** Exploring misogyny across the manosphere in reddit. Proceedings of the 10th ACM Conference on Web Science, pp. 87–96. DOI: 10.1145/3292522.3326045.

16. **Fersini, E., Gasparini, F., Rizzi, G., Saibene, A., Chulvi, B., Rosso,**

P., Lees, A., Sorensen, J. (2022). SemEval-2022 task 5: Multimedia automatic misogyny identification. Proceedings of the 16th International Workshop on Semantic Evaluation, pp. 533–549. DOI: 10.18653/v1/2022.semeval-1.74.

17. **Fersini, E., Nozza, D., Rosso, P. (2018).** Overview of the Evalita 2018 task on automatic misogyny identification (AMI). Proceedings of the 6th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, pp. 59–66.

18. **Fersini, E., Rosso, P., Anzovino, M. (2018).** Overview of the task on automatic misogyny identification at IberEval 2018. Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages, IberEval, Vol. 2150, pp. 214–228.

19. **Fischer, G. R. (1985).** How music communicates. Semiotica, Vol. 53, No. 1–3. DOI: 10.1515/semi.1985.53.1-3.131.

20. **García-Díaz, J. A., Cánovas-García, M., Colomo-Palacios, R., Valencia-García, R. (2021).** Detecting misogyny in spanish tweets. An approach based on linguistics features and word embeddings. Future Generation Computer Systems, Vol. 114, pp. 506–518. DOI: 10.1016/j.future.2020.08.032.

21. **Gourdine, R. M., Lemmons, B. P. (2011).** Perceptions of misogyny in hip hop and rap: What do the youths think?. Journal of Human Behavior in the Social Environment, Vol. 21, No. 1, pp. 57–72. DOI: 10.1080/10911359.2011.533576.

22. **Hewitt, S., Tiropanis, T., Bokhove, C. (2016).** The problem of identifying misogynist language on twitter (and other online social spaces). Proceedings of the 8th ACM Conference on Web Science, pp. 333–335. DOI: 10.1145/2908131.2908183.

23. **Li, B., Hou, Y., Che, W. (2022).** Data augmentation approaches in natural language processing: A survey. AI Open, Vol. 3, pp. 71–90. DOI: 10.1016/j.aiopen.2022.03.001.

24. **Lynn, T., Endo, P. T., Rosati, P., Silva, I., Santos, G. L., Ging, D. (2019).** A comparison of machine learning approaches for detecting misogynistic speech in urban dictionary. Proceedings of the International Conference on Cyber Situational Awareness, Data Analytics And Assessment, pp. 1–8. DOI: 10.1109/CyberSA.2019.8899669.

25. **Pamungkas, E. W., Basile, V., Patti, V. (2020).** Misogyny detection in Twitter: A multilingual and cross-domain study. Information Processing and Management, Vol. 57, No. 6, pp. 102360. DOI: 10.1016/j.ipm.2020.102360.

26. **Peza-Casares, M. D. C. (2009).** Discurso de odio y feminicidios en méxico. Tram[p]as de la Comunicación y la Cultura, Vol. 66, pp. 29–35.

27. **Plaza-Del-Arco, F. M., Molina-González, M. D., Ureña-López, L. A., Martín-Valdivia, M. T. (2020).** Detecting misogyny and xenophobia in spanish tweets using language technologies. ACM Transactions on Internet Technology, Vol. 20, No. 2, pp. 1–19. DOI: 10.1145/3369869.

28. **Reimers, N., Gurevych, I. (2019).** Sentence-BERT: Sentence embeddings using siamese BERT-networks. Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 3982–3992. DOI: 10.18653/v1/D19-1410.

29. **Sanh, V., Debut, L., Chaumond, J., Wolf, T. (2019).** DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. Proceedings of the 5th EMC2 - Energy Efficient Machine Learning and Cognitive Computing Colocated with the 33rd Conference on Neural Information Processing Systems, pp. 1–5. DOI: 10.48550/arXiv.1910.01108.

30. **Shushkevich, E., Cardiff, J. (2019).** Automatic misogyny detection in social media: A survey. Computación y Sistemas, Vol. 23, No. 4, pp. 1159–1164. DOI: 10.13053/CyS-23-4-3299.

31. **Tomás, D., Ortega-Bueno, R., Zhang, G., Rosso, P., Schifanella, R. (2022).** Transformer-based models for multimodal irony detection. Journal of Ambient Intelligence and Humanized Computing, Vol. 14, No. 6, pp. 7399–7410. DOI: 10.1007/s12652-022-04447-y.

32. **Tsokalidou, R. (1989).** Linguistic misogyny - a language universal: Observations, questions and ideas. Selected Papers on Theoretical and Applied Linguistics, Vol. 3, pp. 363–381. DOI: 10.26262/istal.v3i0.7182.

33. **Weitzer, R., Kubrin, C. E. (2009).** Misogyny in rap music: A content analysis of prevalence and meanings. Men and Masculinities, Vol. 12, No. 1, pp. 3–29. DOI: 10.1177/1097184x0832 7696.

34. **Zeinert, P., Inie, N., Derczynski, L. (2021).** Annotating online misogyny. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Vol. 1, pp. 3181–3197. DOI: 10.18653/v1/2021.acl-long.247.

Article received on 25/10/2023; accepted on 14/01/2024.
** Corresponding author is Félix Agustín Castro-Espinoza.*