Tikrit Journal of Pure Science Vol. 30 (5) 2025

DOI: *https://doi.org/10.25130/tjps.v30i5.1522*

IRAQI Academic Scientific Journals

Tikrit Journal of Pure Science

ISSN: 1813 – 1662 (Print) --- E-ISSN: 2415 – 1726 (Online)

Journal Homepage: https://tjpsj.org/

Twitter Text Classification using Convolutional Neural Network Method

Fadya Abdulfattah Habeeb

Department of Mathematics, College of Education for women, Tikrit University, Tikrit, Iraq

Received: 22 Aug. 2023 **Received in revised forum:** 18 Oct. 2023 **Accepted:** 21 Oct. 2023

Final Proof Reading: 2 Oct. 2025 **Available online:** 25 Oct. 2025

ABSTRACT

Text classification on social media platforms such as Twitter has become increasingly crucial. Convolutional Neural Networks (CNNs) have demonstrated their effectiveness across a range of natural language processing tasks, including text classification. The primary goal of this article is to create a reliable and precise text classification model for Twitter by employing CNNs. The CNN architecture is tailored for the text classification task through the application of one-dimensional convolutions on the word embedding. In this article utilize multiple convolutional layers with diverse kernel sizes to capture various levels of contextual information within the input text. Max-pooling layers are employed to extract the most pertinent features from the convolved results. To assess the performance of the text classification model based on CNNs, this study carry out experiments using a diverse dataset of Twitter messages. The dataset is annotated with various categories such as sentiment (positive, negative). Experimental results demonstrate that the proposed CNN model achieves competitive performance compared to state-of-the-art methods for text classification on Twitter. then compared the proposed model with some ML methods like Logistic Regression (LR), Naive Bayes (NB), Stochastic Gradient Descent (SGD), and k-nearest neighbours (KNN) and got the following accuracy: 98%, 97%, 89%, 83%, and 94%.

***Keywords*:** Convolutional Neural Networks, Machine learning, Natural language processing, Sentiment analysis, Text classification.

Name: **E-mail:** Fadya.habeeb@tu.edu.iq

Tikrit Journal of Pure Science Vol. 30 (5) 2025

DOI: *https://doi.org/10.25130/tjps.v30i5.1522*

تصنيف نص تويتر باستخدام طريقة CNN

فاديه عبد الفتاح حبيب

قسم الرياضيات، كلية التربية بنات، جامعة تكريت، تكريت، العراق

الملخص

أصبح تصنيف النص على منصات التواصل الاجتماعي مثل تويتر أمرًا بالغ الأهمية بشكل متزايد. أثبتت الشبكات العصبية التلافيفية (CNN)فعاليتها عبر مجموعة من مهام معالجة اللغة الطبيعية، بما في ذلك تصنيف النص. في هذه المقالة الهدف الأساسي من هذه المقالة هو إنشاء نموذج موثوق ودقيق لتصنيف النص لتويتر من خلال استخدام شبكاتCNN . تم تصميم بنية CNN لمهمة تصنيف النص من خلال تطبيق تلافيفات أحادية البعد على تضمين الكلمات. ثم استخدم طبقات تلافيفية متعددة بأحجام نواة متنوعة لالتقاط مستويات مختلفة من المعلومات السياقية داخل نص الإدخال. يتم استخدام طبقات التجميع القصوى (Max polling) لاستخراج الميزات الأكثر صلة من النتائج المجمعة. لتقييم أداء نموذج تصنيف النص استنادًا إلى شبكاتCNN ، نقوم بإجراء تجارب باستخدام مجموعة بيانات متنوعة من رسائل تويتر . يتم شرح مجموعة البيانات بفئات مختلفة مثل المشاعر (الإيجابية والسلبية). توضح النتائج التجريبية أن نموذج CNN المقترح يحقق أداءً تنافسيًا مقارنة بأحدث الأساليب لتصنيف النص على تويتر. ثم قارنت النموذج المقترح مع بعض طرق تعلم الآلة مثل LR، NB، SGD، وKNN وحصلت على الدقة التالية: 98%، 97%، 89%، 83%، و94%.

INTRODUCTION

The exponential growth of social media platforms, particularly Twitter, has led to an unprecedented influx of textual data, making effective analysis and classification of this data a critical challenge [1]. Text classification, a fundamental task in natural language processing (NLP), plays a pivotal role in extracting meaningful insights from this vast and dynamic stream of information [2]. Convolutional Neural Networks (CNNs), renowned for their prowess in image processing, have shown remarkable potential in text classification tasks as well [3].

Twitter, characterized by its concise and informal communication style, presents a distinctive landscape for text classification. The brevity of tweets, limited to 280 characters, demands specialized techniques to capture context, sentiment, and intent accurately [4]. Traditional text classification methods, often reliant on n-grams and handcrafted features [5], may fall short in grasping the nuanced semantics and subtle patterns prevalent in Twitter data [6]. Convolutional Neural Networks, with their inherent ability to automatically learn hierarchical features, offer a promising solution to extract rich contextual information from these short and contextually dense messages [3].

The CNN architecture's suitability for image recognition tasks has been extended to NLP, particularly text classification, by leveraging one-dimensional convolutions over word embeddings [7]. This transformation allows the model to learn spatial hierarchies of features within sentences, thereby capturing local and global patterns simultaneously [8]. The use of multiple convolutional and max-pooling layers enables the extraction of relevant features from various levels of abstraction, empowering the model to recognize intricate patterns within tweets [9]. For effective training and reliable outcomes, CNN requires a large amount of data [10]. Combining features and creating several feature maps using the CNN network [11].

In that study, the challenges specific to text classification on Twitter, including the presence of noisy data, informal language, and limited context. By harnessing the power of CNNs, this study aim to enhance the accuracy and robustness of classification across diverse categories such as sentiment, topic, and intent [12].

In summary, this study embarks on a comprehensive exploration of applying Convolutional Neural Networks to the task of text classification on Twitter. The subsequent sections delve into the methodology, dataset, experiments, and results, shedding light on the efficacy of this approach in navigating the challenges of Twitter data. By marrying the strengths of CNNs with the idiosyncrasies of Twitter communication, this study aspire to contribute to a more accurate and insightful understanding of the information landscape in the realm of social media.

RELATED WORK

The authors attempt to address the problems of text and emoji classification on Twitter.

Algorithms Conditional Random Field (CRF), Lon g ShortTerm Model (LSTM), Gate Recurrent Unit (GRU), and CNNThen, test a number of experiments to see how well those algorithms function when used with text alone, then with text and emoji. When it came to CRF, emoji results were lower relative to text-only results. Customized architectures that combine sequential and non-sequential traits were used for the CNN and LSTM algorithms. This yields a maximum accuracy of 79% by used CNN algorithms [13].

The study concentrated on examined how the CNN model addressed issues related to text classification. The study utilized not only a non-standard dataset but also six benchmark datasets specifically, Ag News, Amazon Full, Polarity, Yahoo Question Answer, Yelp Full, and Polarity to train their rating mode. The suggested model underwent testing on Twitter US airlines data, and it's important to note that this method relies on raw data without employing any manual feature extraction or feature selection methods. The results from the Twitter US airlines dataset are as follows: Accuracy at 0.860, Precision at 0.840, Recall at 0.890, and an F-score of 0.864 [14].

The authors employed the CNN method to address text classification issues.

The key feature of TextConvoNet is its ability to capture both intra-sentence n-gram features within text data and inter-sentence n-gram features. This is accomplished by utilizing a 2-D CNN model to create an alternative input representation for the text data. The study performed an experiment on five binary and multi-class classification datasets to assess the TextConvoNet's performance in text classification. The evaluation encompassed eight performance metrics, including accuracy, precision, recall, F1-score, specificity, gmean1, gmean2, and the Mathews correlation coefficient (MCC). Accuracy 0.904 F1 score 0.883. Recall 0.978Precision 0.978 [15].

The issue of cyberbullying and hate speech on Twitter. Hate speech targets multiple protected characteristics, such as gender, religion, race, and disability. Filtering such a vast influx of information manually is nearly unmanageable. In relation to this matter, an automated system has been created utilizing the Deep Convolutional Neural Network (DCNN). The DCNN model presented makes use of GloVe embedding vectors for tweet text, enabling the capture of tweet semantics through convolution operations, resulting in the accomplishment of a particular goal. precision, recall and F1-score value as 0.97, 0.88, 0.92 [16]. Table 1 explains the summary of previous studies.

Table 1: a summary of the evaluations of the CNN methods

DL Method	Reference	Acc (%)	precision	Recall	F1-score
CNN	[13]	%78.2	%79	%78.73	%78.3
	[14]	%86	%84	%89	%86.4
	[15]	%90.4	%97.8	%97.8	%88.3
	[16]		%97	%88	%92

PROPOSED APPROACH (METHODOLOGY)

Dataset Description

[17] Generated the dataset utilized in this article, comprising 31,962 tweets sourced from Twitter through the Twitter API. These tweets were collected from Twitter users who shared content containing terms derived from Hatebase.org's compilation of offensive language. Human annotators have categorized all instances as either containing negative speech or not. The majority of the tweets, amounting to 92.99 percent, are identified as non-negative speech, while only 7.01 percent of the tweets fall under the category of negative speech as presented in figure 1. In scenarios involving imbalanced datasets, many classifiers tend to perform inadequately due to the likelihood of overlooking members of the underrepresented category. To address this concern, oversampling techniques are employed in this study to mitigate the issue of class imbalance.

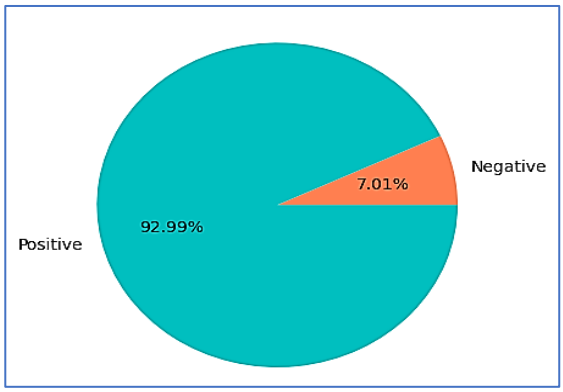

Fig 1: sentiment classes distribution

Pre-processing

Preprocessing using The Natural Language Toolkit (NLTK) for text classification on Twitter involves a series of steps to clean and transform the raw text data into a format suitable for machine learning algorithms. Here's a concise explanation of the key preprocessing operations:

- Text Lowercasing: Convert all text to lowercase to ensure consistent treatment of words regardless of their case.
- Tokenization: Split the text into individual words or tokens. This step is crucial for understanding the structure of the text.
- Removing Special Characters and URLs: Remove any special characters, symbols, and URLs that do not contribute to the meaningful content of the text.
- Removing Stopwords: Exclude common words like "and," "the," "is," etc., as they typically do not carry significant meaning in classification tasks.
- Stemming/Lemmatization: Reduce words to their root forms using stemming or lemmatization. This helps to consolidate variations of the same word.
- Handling Emojis and Hashtags: Depending on the context, you might choose to remove, replace, or retain emojis and hashtags, as they can carry sentiment or thematic information.
- Handling User Handles (@mentions):** Decide whether to remove or replace user handles, as they might not contribute to the classification task. Retaining them could provide insight into user interactions.
- Handling Numerical Data: If applicable, convert numbers to a placeholder token to maintain text consistency.

- Vectorization: Convert the processed text into numerical representations suitable for machine learning algorithms. This can be achieved using the technique word embeddings (this article used GloVe).
- Padding: Ensure that all text sequences have the same length by adding padding to shorter sequences, which is essential for input consistency in neural network models.
- Handling Imbalanced Data: Address class imbalances if present by oversampling technique.
- Data Splitting: Divide the preprocessed dataset into training, validation, and testing sets to evaluate and tune the classification model in this paper the dataset is split into 70% for training and 30% for testing.

These preprocessing operations collectively enhance the quality of the text data and prepare it for effective machine learning-based text classification on Twitter. Keep in mind that the specific steps might vary depending on the characteristics of your dataset and the objectives of your classification task.

CNN Architecture Design

Certainly! When using a Convolutional Neural Network (CNN) for text classification, the architecture is slightly different from the traditional image-based CNN. Here's a description of the typical layers in a CNN model designed for text classification:

- Input Layer: The input layer accepts text data, which is usually represented as sequences of words or tokens. Each word can be encoded using techniques like word embeddings (this study used GloVe) to convert them into dense vectors.
- Embedding Layer: This layer converts the word tokens into dense vectors. Each word's embedding captures semantic relationships, helping the model understand the context of words in the text.
- Convolutional Layer (1D Convolution) Unlike traditional 2D convolutions used in image CNNs, 1D convolutions slide over sequences of word embeddings. These convolutions capture local patterns and relationships between neighboring words.
- Activation Function: After the convolution operation, an activation function (used ReLU in this paper) is applied element-wise to introduce non-linearity.
- Max Pooling Layer: Similar to image CNNs, max pooling is applied to reduce the spatial dimensionality of the feature maps, focusing on the most relevant information.
- Flatten Layer: The feature maps from the convolutional and pooling layers are flattened into a 1D vector to connect to fully connected layers.
- Fully Connected (Dense) Layers: These layers process the flattened features and perform classification based on the extracted patterns. The number of neurons in the output layer corresponds to the number of classes in the classification task.
- Output Layer: The final fully connected layer produces class probabilities. Activation functions sigmoid are used to convert raw scores into class probabilities.

Figure 3 below shows the proposed CNN model, in addition, table 2 presents the setting parameters and figure 2 presented Proposed CNN Model that were used in this study.

Table 2: CNN setting parameters

embedding dim	Max Length	Layer number	Activation	optimizer	epochs	batch size
100	100	5	Relu and sigmoid	Adam	10	10

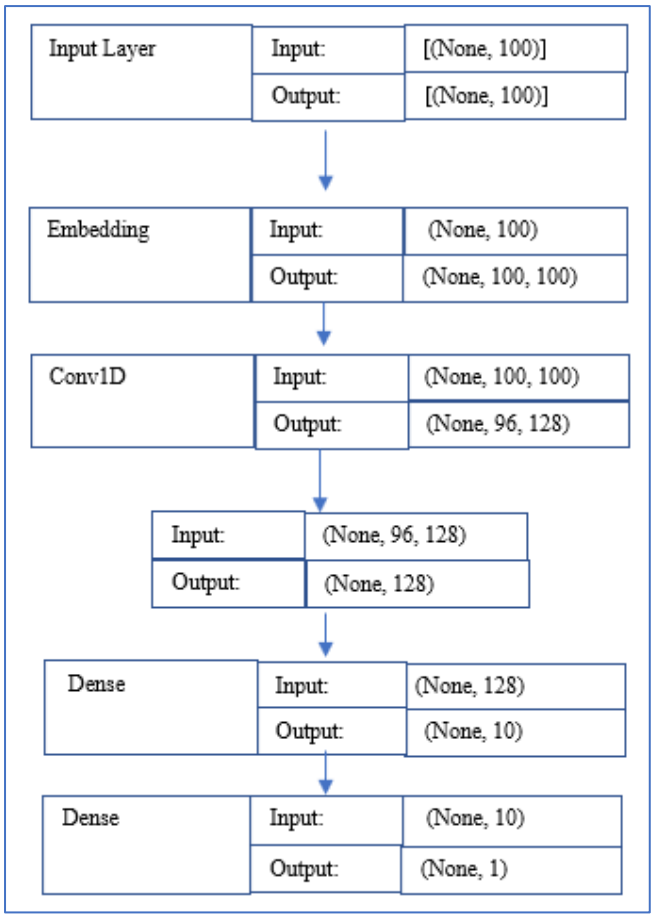

Fig 2: Proposed CNN Model

Model Evaluation

After training is complete, assess the model's performance on the testing set. This study apply the chosen evaluation metrics to measure the model's accuracy, robustness, and ability to generalize to new data. The following metrics are used to measure the efficiency of the proposed model:

- Confusion Matrix: Construct a confusion matrix to visualize the model's performance across different classes. This matrix provides information about true positives, true negatives, false positives, and false negatives.

True Positive (TP): is a situation that should be classed as hate speech.

True Negative (TN): classified as non-hate speech.

False Positive (FP): this is a non-hate speech case that has been incorrectly labelled as hate speech.

False Negative (FN): In this situation hate speech that was mistakenly labelled as non-hate speech.

- Accuracy: Calculate the proportion of correctly classified instances out of the total number of instances. While accuracy is a common metric, it might not be suitable for imbalanced datasets.

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+TN} \quad 4.1$$

- Precision and Recall: Precision represents the proportion of true positive predictions among all positive predictions, while recall represents the proportion of true positive predictions among all actual positive instances. These metrics are particularly useful for imbalanced datasets.

$$\text{Precision} = \frac{TP}{TP+FP} \quad 4.2$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad 4.3$$

- F1-Score: The F1-score is the harmonic mean of precision and recall, providing a balanced metric that considers both false positives and false negatives.

$$F1-\text{score} = \frac{2.\text{Precision}.\text{Recall}}{\text{Precision}+\text{Recall}} \quad 4.4$$

- Interpreting Results: Analyze the evaluation metrics and the confusion matrix to understand the model's strengths, weaknesses, and potential areas for improvement [18].

Comparative Analysis: Compare the model's performance to baseline models, other algorithms, or previous iterations to determine its effectiveness. this paper, compared the proposed model with some ML methods like Logistic Regression (LR), Naive Bayes (NB), Stochastic Gradient Descent (SGD), and k-nearest neighbors (KNN).

RESULTS AND COMPARISON

In this section, this article present the outcomes of our study focused on text classification on Twitter using various machine learning techniques. Due to the immense volume of textual content produced daily, Our investigation aimed to assess the efficacy of these methods in categorizing tweets into positive or negative based on their content. The first step for results is data preprocessing which is executed using steps in section 3.2. The following table presents tweets before and after preprocessing.

Table 3: tweets before and after preprocessing

id	Label	tweet	Clean tweet	#
0	0	@user when a father is dysfunctional and is so selfish he drags his kids into his dysfunction. #run	when father dysfunctional selfish drags kids into dysfunction	#run
1	0	@user @user thanks for #lyft credit i can't use cause they don't offer wheelchair vans in pdx. #disapointed #getthanked	thanks lyft credit cause they offer wheelchair vans disapointed getthanked	#lyft #disapointed #getthanked
2	0	bihday your majesty	bihday your majesty	Na
3	0	#model model love take with all the time in	model love take with time	#model
4	0	factsguide: society now #motivation	factsguide society motivation	#motivation
…	…	…	…	…
31935	1	lady banned from kentucky mall jcpenny Kentucky #jcpenny #kentuck	lady banned from kentucky mall jcpenny kentucky	#jcpenny #kentuck
…	…	…	…	…
31959	0	to see nina turner on the airwaves trying to wrap herself in the mantle of a genuine hero like shirley chisolm. #shame #imwithher	nina turner airwaves trying wrap herself mantle genuine hero like shirley chisolm shame imwithher	#shame #imwithher
31960	0	listening to sad songs on a monday morning otw to work is sad	listening songs monday morning work	Na
31961	1	@user #sikh #temple vandalised in in #calgary, #wso condemns act	sikh temple vandalised calgary condemns	#sikh #temple #calgary, #wso
31962	0	thank you @user for you follow	thank follow	Nan

The table above has several columns. First is 'ID,' which explains the number of tweets in the dataset. Next is 'Label,' which should have values of 0 or 1 based on the tweet's content—1 if it contains hate speech and 0 if it does not. Following that is 'Tweet,' which contains the original, unprocessed tweets. Then, there's 'CleanedTweet,' which includes tweets after preprocessing. Finally, there is a column labelled '#,' which contains the hashtags found in the tweets. At this point, the CNN model which is built in step 3.3 is run and then performed the evaluation metrics that are shown in section 3.4.

Finally, Table 4 shows the results obtained by various classifiers on the collected dataset based on selected metrics that were explained in the previous part 3.4.

Table 4: results of CNN with other classifiers

Method	Accuracy	Precision	Recall	F1-Scoure
CNN	98%	96.6%	93.9%	95%
LR	97%	91%	97%	94%
NB	89%	99%	45%	62%
SGD	83%	100%	10%	18%
KNN	94%	98%	69%	81%

Comparing classifiers in machine learning involves evaluating their performance on tweet classification to determine which one excels. Various metrics like

DOI: https://doi.org/10.25130/tjps.v30i5.1522

accuracy, precision, recall, and F1-score are used to gauge their effectiveness. This comparison aids in selecting the most suitable model for a specific problem, considering factors such as complexity, interpretability, and robustness. Classifier comparison empowers data scientists to make informed decisions, optimizing outcomes in diverse applications. Figure 3 presented the Comparing between CNN and other ML methods. While fig 4 shown the confusion matrix for each of them.

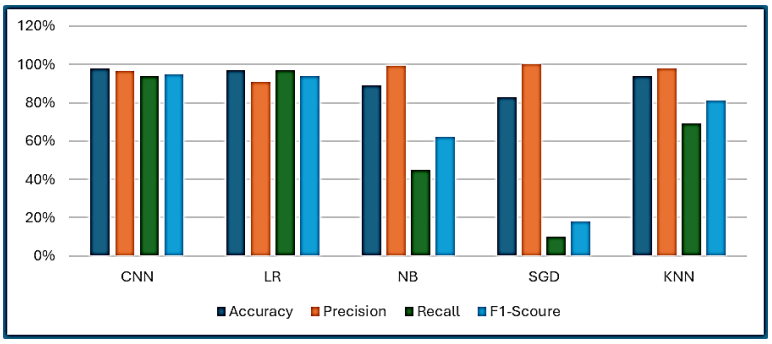

Fig 3: Comparing between CNN and other ML methods

CNN got the following results: Accuracy 98%, Precision 96.6%, Recall 93.9% and, F1-Score 95%. The CNN model demonstrates strong overall performance with high accuracy and balanced precision and recall. This suggests that the model is effective in correctly classifying instances of both positive and negative cases. while LR got Accuracy 97%, Precision 91%, Recall 97% and F1-Score94%. The LR model performs well in terms of accuracy and recall. However, the precision is slightly lower, indicating that there might be more false positives compared to the CNN model. In addition to NB: Accuracy is 89%, Precision 99% Recall 45%, and F1-Score 62%. The Naive Bayes model achieves high precision but at the cost of lower recall. This suggests that it is good at identifying positive cases but may miss a significant number of actual positive instances. Next SGD model got an accuracy of 83%, Precision of 100%, Recall: 10%, and F1-Score: of 18%. The SGD model exhibits high precision but very low recall, indicating that while it is good at identifying positive cases, it misses a substantial number of them. This might be a case of overfitting to the negative class. Finally, KNN results were Accuracy 94%, Precision 98%, Recall: 69%, and F1-Score: 81%. The KNN model provides a good balance between precision and recall, with relatively high accuracy. It seems to be effective in correctly classifying instances from both classes.

In summary, the choice of a model depends on the specific requirements of your task. The CNN and LR models seem to perform well overall, each having its strengths and weaknesses. The SGD model might need further tuning to improve its recall. The NB model, while having high precision, might need improvements in recall for better overall performance. So, the best one is CNN model.

DOI: https://doi.org/10.25130/tjps.v30i5.1522

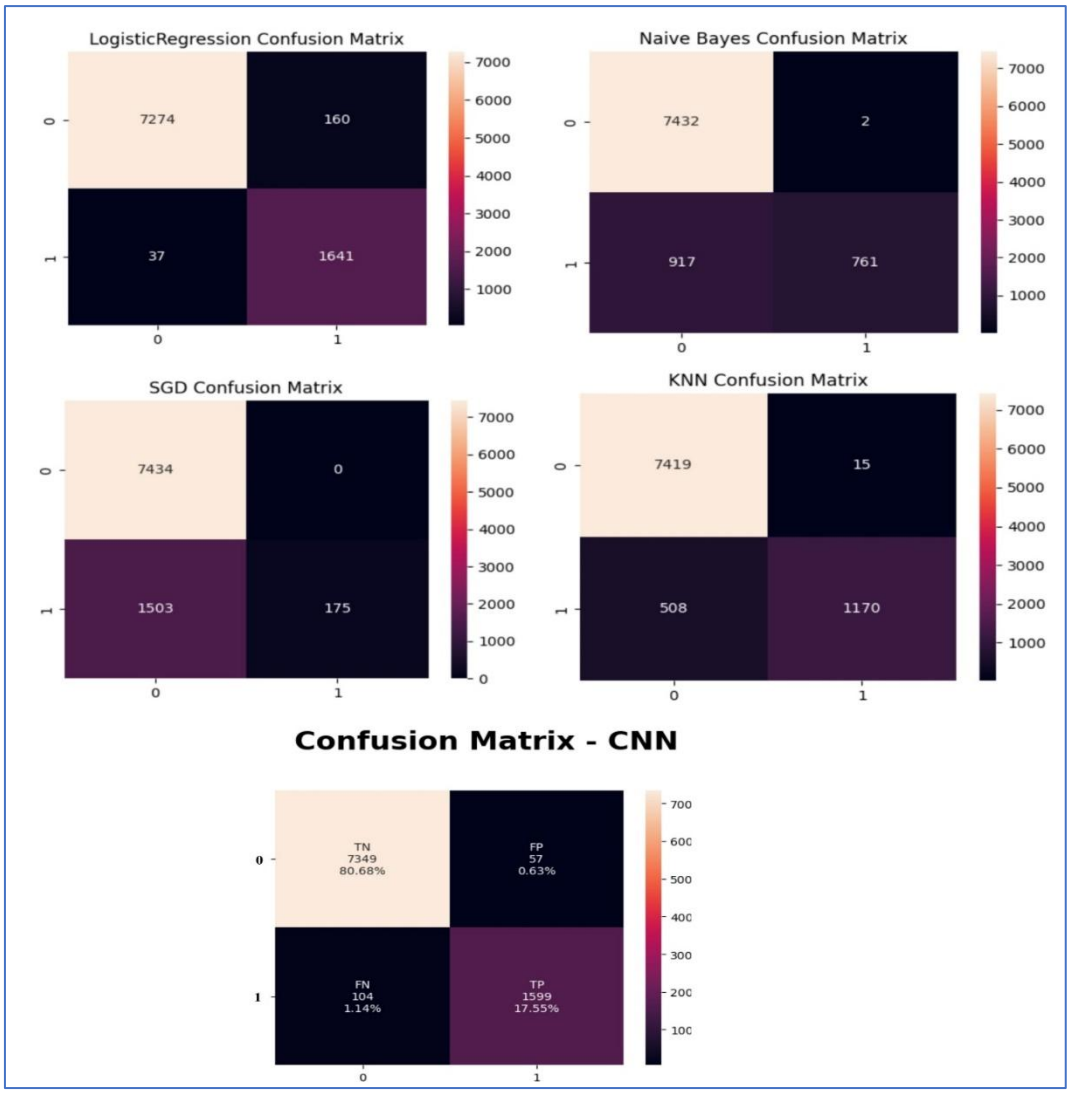

Fig. 5: Confusion Matrix

Our findings reveal notable distinctions in the performance of selected algorithms. The CNN consistently outperformed both LR, NB, SGD and KNN across evaluation metrics. This outcome underscores CNN's ability to capture intricate contextual nuances within the often dynamic and emotive language of Twitter.

CONCLUSION

This article enhanced the capabilities of the CNN model, designed for classifying a set of tweets sourced from Twitter into either positive or negative categories, alongside their respective characteristics. Initially, the tweets underwent thorough cleaning and preprocessing to ready them for the advanced stages of the project. As a result of this preprocessing, a CSV file containing the tweets and labels was generated, revealing that the majority of tweets, specifically 92.99%, were identified as positive, while only 7.01% were genuinely considered negative. Given the challenges posed by imbalanced datasets, where classifiers often falter due to overlooking underrepresented categories, this study employed oversampling as a technique to address the issue of class imbalance. Moving forward, recognizing that tweets serve as the foundational content structure characterized by natural human language, the study applied Natural Language Processing (NLP). Commonly, NLP applications involve a sequence of stages. Subsequently, the CNN architecture was customized for text classification, with a focus on optimizing the parameters of this model. All these

procedures were executed by implementing Python code. The proposed CNN model achieved competitive performance compared to state-of-the-art methods for text classification on Twitter. Then compared the proposed model with some ML methods like LR, NB, SGD, and KNN and are given the following accuracy: 98%, 97%, 89%, 83%, and 94%.

In this study, just text in Twitter was employed. But keep in mind that hate speech on Twitter can also appear in the way of images and videos. Future databases will need to contain not only text but also pictures and videos to meet this challenge. Future work needs to focus on creating algorithms that can collect both textual and non-textual material. The detection of hate speech could be significantly enhanced through these advancements.

REFERENCES

1. Theocharopoulos PC, Tsoukala A, Georgakopoulos SV, Tasoulis SK, Plagianakos VP. Text analysis of COVID-19 tweets. In: *Engineering Applications of Neural Networks*. 2022. p. 517–528.

2. Zhang F, Fleyeh H, Wang X, Lu M. Construction site accident analysis using text mining and natural language processing techniques. *Automation in Construction*. 2019;99:238–48. doi: https://doi.org/10.1016/j.autcon.2018.12.016

3. Taye MM. Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions. *Computation*. 2023;11(3). doi: 10.3390/computation11030052

4. Scott K. The pragmatics of hashtags: Inference and conversational style on Twitter. *Journal of Pragmatics*. 2015;81:8–20. https://doi.org/10.1016/j.pragma.2015.03.015

5. Nagar A, Bhasin A, Mathur G. Text classification using gated fusion of n-gram features and semantic features. *Computación y Sistemas*. 2019;23(3). doi: 10.13053/cys-23-3-3278

6. Agarwal A, Xie B, Vovsha I, Rambow O, Passonneau R. Sentiment analysis of Twitter data. In: *Proceedings of the Workshop on Language in Social Media (LSM 2011)*. 2011. p. 30–8. Available from: https://aclanthology.org/W11-0705

7. Soni S, Chouhan SS, Rathore SS. TextConvoNet: A convolutional neural network based architecture for text classification. *Applied Intelligence*. 2023;53(11):14249–68. doi: 10.1007/s10489-022-04221-9

8. Qi CR, Yi L, Su H, Guibas LJ. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: *Advances in Neural Information Processing Systems*. 2017;30. Available from: https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf

9. Zafar A, et al. A comparison of pooling methods for convolutional neural networks. *Applied Sciences*. 2022;12(17). doi: 10.3390/app12178643

10. Aljaloud S. Performance refinement of convolutional neural network architectures for solving big data problems. *Tikrit Journal of Pure Science*. 2023;28(1):89–95. doi: 10.25130/tjps.v28i1.1270

11. Hussein DM, Beitollahi H. A hybrid deep learning model to accurately detect anomalies in online social media. *Tikrit Journal of Pure Science*. 2022;27(5):105–16. doi: 10.25130/tjps.v27i5.24

12. Dang NC, Moreno-García MN, De la Prieta F. Sentiment analysis based on deep learning: A comparative study. *Electronics*. 2020;9(3):483. doi: 10.3390/electronics9030483

13. Messaoudi C, Guessoum Z, Ben Romdhane L. A deep learning model for opinion mining in Twitter combining text and emojis. *Procedia Computer Science*. 2022;207:2628–37. https://doi.org/10.1016/j.procs.2022.09.321

14. Umer M, et al. Impact of convolutional neural network and FastText embedding on text classification. *Multimedia Tools and Applications*. 2023;82(4):5569–85. doi: 10.1007/s11042-022-13459-x

15. Soni S, Chouhan SS, Rathore SS. TextConvoNet: A convolutional neural network

based architecture for text classification. *Applied Intelligence*. 2022. doi: 10.1007/s10489-022-04221-9

16. Roy PK, Tripathy AK, Das TK, Gao X-Z. A framework for hate speech detection using deep convolutional neural network. *IEEE Access*. 2020;8:204951–62. doi: 10.1109/ACCESS.2020.3037073

17. Kaggle. Twitter_Sentiment_Analysis [dataset]. Available from: https://www.kaggle.com/datasets

18. Habeeb MA. Hate speech detection using deep learning [Master's thesis]. 2021.