Journal of Applied Research and Technology

www.jart.icat.unam.mx

ICAT Instituto de Ciencias Aplicadas y Tecnología

Journal of Applied Research and Technology 24 (2026) 778-787

Original

A Deep Learning Approach for Hate Speech Detection on Twitter: Comparative Analysis with Traditional Models

Abdulwahed Alothman[1*] • Bassel Alkhatib[1] • Mohamed Khatib[2]

[1]*Syrian Virtual University, Damascus, Syria*
[2]*Artificial Intelligence Department, University of Aleppo, Aleppo, Syria*

Received: 20 02 2025; Accepted: 17 08 2025
Available: 31 08 2026

Abstract: In the modern era of technology, social media platforms have become the primary channels for communication between people and organizations and for sharing and disseminating information, but the open nature of social media platforms makes them a fertile ground for harmful content, including hate speech. Hate speech is a form of expression that incites or encourages discrimination, hatred, or violence against individuals or groups based on certain characteristics, including but not limited to race, religion, gender, or sexual orientation, and has become a growing concern, making the spread of such content pose significant challenges to online communities and social cohesion. In this study, we propose a deep learning-based model to classify Twitter tweets into three categories: hate speech, neither, and abusive. The proposed model relies on the use of a multi-layer neural network with a set of dropout layers to enhance the generalization of the model and avoid the problem of overfitting. The performance of the proposed approach was analyzed and contrasted with traditional machine learning (ML) algorithms, including decision tree, random forest, and support vector machine (SVM). The experimental results showed that the proposed model reached the best accuracy of 97%, significantly outperforming the basic machine learning algorithms. The results confirm the effectiveness of the proposed model and its ability to capture the underlying patterns and semantic and linguistic complexities within the tweet text content, which enables it to classify hate speech with high accuracy.

Keywords: **Social media, hate speech, deep learning, classification, Twitter**

*Corresponding author.
E-mail address: abdulwhed.alothman@gmail.com (A.A.J. Al Noor).
Peer Review under the responsibility of Universidad Nacional Autónoma de México.

1. Introduction

In our time, social media platforms have become the basic channels of communication between people and organizations for the sharing and dissemination of information, but the open nature of social media platforms makes them fertile ground for harmful content, including hate speech (Lin, 2022; Meier & Johnson, 2022). Hate speech, which is characterized by expressions that encourage discrimination, hostility, or violence against individuals or groups based on features such as race, religion, sex, or sexual orientation, has become an increasing source of concern, making the spread of such content pose a great challenge for online communities and social cohesion (Alkomah & Ma, 2022).

Twitter is a social networking platform that provides a widely used blogging service that allows users to share ideas, opinions, and news in real time. Therefore, detecting hate speech and mitigating it on Twitter is very important to enhance the online communication environment and make it safer and more comprehensive. Relying on scrutiny to detect hate speech by those responsible for the platform is an inefficient solution due to the huge number of daily tweets generated. This shows an urgent need to develop automatic systems based on advanced artificial intelligence techniques to identify and filter hate speech efficiently, ensuring that harmful content is addressed immediately.der to define and filter hate speech efficiently, ensuring that harmful content is addressed immediately.

The unique characteristics of tweets, such as their brevity, the use of informal language, the use of abbreviations, as well as the dynamic nature of the topics that are discussed, constitute great challenges in detecting hate speech (Pereira-Kohatsu et al., 2019). Therefore, the distinction between hate speech and legitimate expressions of opinion or mockery requires advanced techniques in addressing the natural language capable of analyzing texts, extracting semantic and grammatical patterns, and capturing complex relationships between texts that enable them to understand the context and the meaning.

To address these challenges, many methods are used to detect hate speech, including dictionary-based methods (Lescoat et al., 2018), machine learning (ML) models, and deep learning (DL) models (Janiesch et al., 2021).

The dictionary-based methods include the use of dictionaries containing pre-specified words, including hateful terms, which are used to identify sentences that include hate speech or offensive content (William et al., 2022). The methods based on the dictionary are often limited to capturing the advanced nature of hate speech and changes that can occur in the form and nature of tweets and abbreviations that can include them. On the other hand, machine learning models that have been trained on large data sets of tweets can achieve higher accuracy through learning patterns and nuances in the language (Bataineh et al., 2023). Deep learning models, especially those that benefit from transformer-based structures such as BERT (Bidirectional Encoder Representations from Transformers), have shown remarkable performance in understanding complex linguistic structures and contextual consequences (Wang et al., 2024).

Detecting hate speech on Twitter is a complex but crucial endeavor that requires a multifaceted approach that combines technological innovation, ethical principles, and community engagement. In response, this research seeks to use deep learning techniques to detect hate speech posted by users, which can help assess user behavior on social media.

2. Literature Reviews

The existing literature explores numerous approaches to studying various aspects of hate speech on online social media. These detection methods primarily fall into three categories:

- Quantitative evaluation of written material.
- Pattern mining techniques.
- ML and deep learning approaches.

In conventional methodologies that rely on manually crafted attributes, investigators initiate their process by isolating characteristics from written data, user demographics, pattern-based templates, and social network interactions involving tweets. These attributes are subsequently utilized to develop and assess machine learning categorization models, encompassing probabilistic models, hierarchical decision structures, ensemble methods, and extreme gradient boosting algorithms. In a study by Davidson et al., (2017), the authors formulated attributes such as single-word units, two-word combinations, part-of-speech tag-based sequences, emotional intensity scores, and other linguistic traits. These attributes were assessed using a probability-based classification model for identifying hostile speech. Similarly, in another approach involving multiple categories, Malmasi and Zampieri (2017) utilized a support vector machine (SVM) trained on features derived from character and word n-grams to classify text into hate, offensive, and neutral categories. Their findings indicated that the character 4-gram-based model achieved the best performance.

Classical ML models lack robustness, as users can exploit and bypass the human-designed features by altering their behavior to counteract these features. In recent years, DL techniques have increasingly supplanted ML models across various applications due to their ability to learn complex patterns directly from the data. The study by Roy et al. (2020) introduced a deep convolutional neural network (CNN)-based framework that achieved a notable accuracy of 92% on a Twitter dataset. Recently, scientists have also begun investigations into the hate content detection problem in code-mixed languages like Hinglish. The study by Pamungkas et al. (2021) introduced a zero-shot learning-based framework for the detection of hate speech within cross-lingual content. Further, different aspects of hate speech that are being considered by researchers include analyzing social groups that are targeted by hate (Mossie & Wang, 2020) and assessing the generalizability of hate speech detection algorithms tested on different datasets and hate categories (Fortuna et al., 2021).

3. The Proposed Methodology

The methodology used in this research is based on building a model based on deep learning techniques to detect hate speech in tweets published on Twitter. The proposed methodology includes a set of steps that are illustrated in Figure 1.

3.1. Data Collection

Our work relied on the Twitter dataset obtained from Kaggle (Samoshyn, 2000). The dataset includes 25,296 tweets classified into 3 categories (hate speech, offensive language, and neither).

3.2. Data Preprocessing

This step involves a set of preprocessing operations performed on the tweet texts. Preprocessing operations are a very important step in artificial intelligence (AI) models because they help reduce noise and thus focus on meaningful patterns, which leads to better predictions (Maharana et al., 2022). In addition, preprocessing operations on texts help in standardizing the text and reducing dimensions by using lowercase letters, removing special characters, and removing unimportant words, which reduces the complexity of the feature space, improves computational efficiency, and improves the ability of the model to generalize better to unseen data, which prevents overfitting.

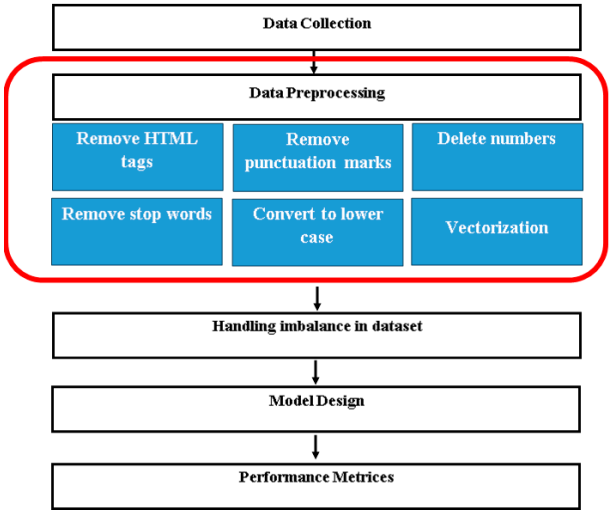

Figure 1. Steps followed in the proposed methodology.

3.2.1. Remove HTML Tags

Removing HTML tags refers to the process of deleting HTML elements (such as <p>, <b>, <a>, etc.) from a text string, leaving only the plain textual content. Unorganized written content frequently includes substantial irrelevant data, particularly when methods such as web or page extraction are employed. HTML and HTTP markers typically consist of elements that contribute little to comprehending and examining the text. Consequently, in this segment, we eliminate superfluous markers and preserve the textual data in all documents (Chai, 2023).

3.2.2. Remove Punctuation Marks

During this phase, all collections of typographical symbols are eliminated from the text. These symbols appear universally in sentences, and their removal is expected to yield beneficial outcomes (Chai, 2023). This process is commonly used in text preprocessing to clean data for further analysis, such as natural language processing (NLP).

3.2.3. Delete Numbers

During text preprocessing, numbers in the text data are removed because they are not relevant to text analysis, have no relation to hate speech, and do not provide topic-relevant information.

3.2.4. Remove Stop Words

Stop words refer to frequently occurring terms that carry little to no significant meaning, particularly when extracting semantic attributes from text (Sarica & Luo, 2021). These words typically consist of conjunctions, articles,

prepositions, and similar parts of speech. Examples of stop words include and, the, an, and a. In the context of natural language processing, keeping these words takes up memory space and takes more processing time, so these words are removed from the text during the preprocessing stage.

3.2.5. Convert to Lowercase

Lowercase conversion refers to the process of converting all uppercase letters in a text to their lowercase equivalents and is commonly used in text preprocessing to ensure consistency, especially when working with natural language processing (NLP).

3.2.6. Vectorization

In the context of machine learning and natural language processing (NLP), vectorization refers to the process of converting text data (or other types of data) into numerical representations that machine learning or deep learning models can process and analyze (Sabri et al., 2022). Since these models only deal with numerical inputs, text data must be converted into numerical vectors while preserving as much meaningful information as possible. For the vectorization process, the TF-IDF (Term Frequency-Inverse Document Frequency) method was used, which is a measure that measures the importance of a word by considering its frequency in a document relative to all documents in the dataset (Widianto et al., 2024). TF-IDF is computed as the product of two components:

- Term Frequency (TF).

$$\mathrm{TF(w)} = \frac{\#\mathrm{wappearsinadocument}}{\#\mathrm{wordsinthedocument}}$$

- Inverse Document Frequency (IDF).
  IDF measures how important a word is by penalizing frequently occurring words across the entire corpus.

$$\mathrm{IDF(w)} = log\left(\frac{\mathrm{Totalnumberofdocuments}}{\mathrm{Numberofdocumentscontainingthewordw}}\right)$$

- The final TF-IDF score is computed as:

$$TF\text{ - }IDF = TF(w) \times IDF\ (w)$$

3.3. Manage Imbalance in Dataset

Data imbalance is the greatest challenge, especially in the field of AI. Among the many factors that cause the above imbalance in a dataset, one is the fact that a particular class contains a significantly higher number of samples than the other classes. Several problems can arise (Hasanin et al., 2019), including:

- Biased models: These types of models are biased toward the majority class. The model prioritizes the majority class such that it learns very poorly from minority classes, and hence the performance of the model becomes poor.
- Unreliable performance metrics: Some performance metrics such as accuracy become unreliable in case the data is unbalanced as they could yield high results that do not represent the performance of the model. This happens because the majority of predictions regarding the majority class turn out to be correct.
- Overfitting: The model ends up memorizing the features and patterns from the majority class, thereby keeping it from generalizing well to the minority class.

Thus, at the present stage, a corrective action must be taken in defecting the data in the dataset. The number of samples corresponding to each class is illustrated in Figure 2.

One way to solve this problem is to undersample the majority class; this implies that certain entries from the predominant category are omitted, thereby equalizing the number of instances in both categories, but this process leads to the loss of a lot of data that can be very important and useful in the process of training the model. Another option is to oversample the minority class, which means that a random iteration of the rows of the minority class is performed until the number of samples for the majority and minority classes becomes equal, but the problem with this method is that it leads to the problem of overfitting because the model learns from the same examples.

To solve these problems, this research will rely on the synthetic minority oversampling technique (SMOTE), while introducing some new topics to this kind of forum, you may be mistaken for spamming (Chawla et al., 2002). SMOTE is a preprocessing technique used to address imbalance in a dataset. The working mechanism of SMOTE can be summarized as follows (Fernández et al., 2018):

- First, a sample is selected from the minority class, and the difference between the sample and its nearest neighbor is calculated.
- The random number generated is the multiplier.
- This figure is then added to the sample to form a completely new variant of that sample.
- The process is repeated N/100 times for each data sample selected. In other words, if the required oversampling is 300, then (300/100= 3) neighbors are selected from the K nearest neighbors, and one sample is generated in the direction of each neighbor.

- All the steps are repeated until the appropriate number of samples is reached, which makes the dataset balanced.

Figure 3 displays the number of samples for each class after the application of the oversampling process using the SMOTE method.

3.4. Model Design

The DL model used in this research was designed based on the Keras framework within the Python programming language, and it is based on stacking a set of dense layers in addition to dropout layers. Figure 4 shows the design of the deep learning model.

The model shown in Figure 4 is a neural network designed using TensorFlow's Keras framework and is designed for a classification task with three classes in the output layer. The model starts with an input layer defined by the number of features in the dataset (12,085) resulting from preprocessing operations performed on the tweet texts. The first hidden layer consists of 128 dense (fully connected) neurons with a ReLU (rectified linear unit) activation function that helps introduce nonlinearity and improve the learning process of the model. To prevent overfitting, the dense layer is followed by a dropout layer with a random 50% activation rate of neurons during training. The next hidden layer contains 64 neurons with a ReLU activation function, followed by another dropout layer with a 50% dropout rate to enhance the generalizability of the model. Finally, the output layer contains 3 neurons with a softmax activation function that converts the outputs into probabilities across the three possible classes (hate speech, offensive language, and neither).

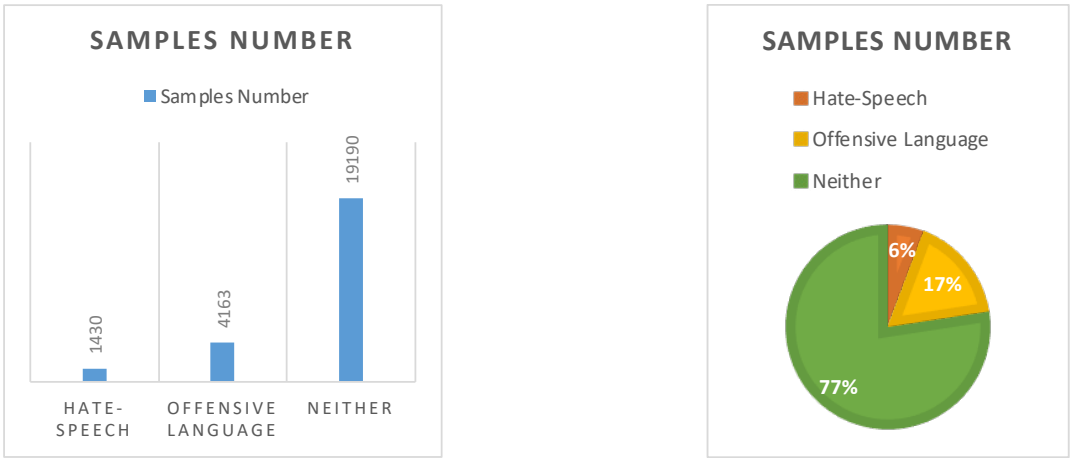

Figure 2. Number of samples for each class.

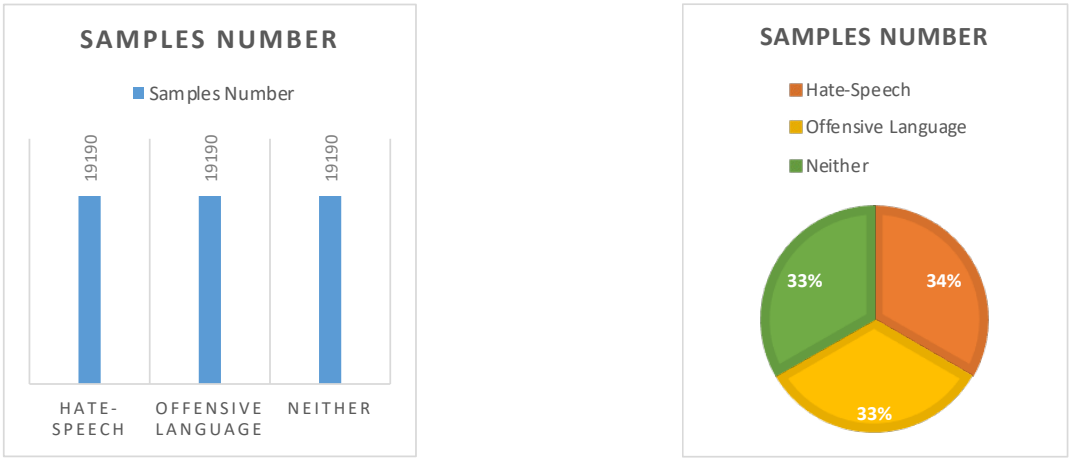

Figure 3. Number of instances, after applying the SMOTE method, in which each separate class occurred.

3.5. Performance Metrics

A confusion matrix (CM) is a matrix used to evaluate the performance of a model, especially in supervised learning (Haghighi et al., 2018; Markoulidakis et al., 2021). The structure of the CM is presented in Figure 5.

Where:

- TP (true positive): When the model accurately forecasts a favorable result.
- TN (true negative): When the model accurately forecasts an unfavorable result.
- FP (false positive): When the model mistakenly forecasts a favorable result.
- FN (false negative): When the model mistakenly forecasts an unfavorable result.

These parameters help in calculating the major performance metrics like recall, precision, and accuracy.

$$Recall = \frac{TP}{TP+FN} \quad (1)$$

This equation could be read as "it is the number of all actual positives divided by the number of those correctly identified by the model as being one of the positives" (Roth et al., 2022).

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

As above, it could be read as "measures the amount of actual right positive predictions" (MacEachern & Forkert, 2021).

Accuracy, in general, reflects the model's overall correctness, obtaining the number of correct predictions (both positives and negatives) compared to all predictions. The equation is as follows (MacEachern & Forkert, 2021):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

The F1 score is a common metric that tries to combine precision and recall to provide an idea of the overall performance of the model (MacEachern & Forkert, 2021). The formula is as follows:

$$F1_Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

4. Results and Discussions

A learning curve is a graphical representation that displays the progression of a specific metric throughout the training phase of a model. The x-axis represents progress, while the y-axis reflects error or performance.

The representation is useful to check the changes that occur in the performance of the model during the learning phase, enabling the detection of problems if they exist. One of the most common curves is the loss function

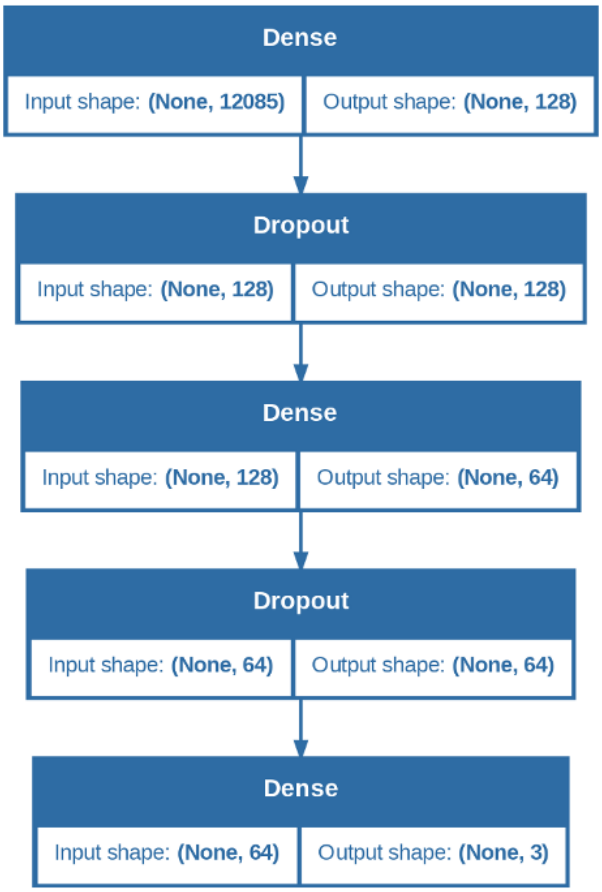

Figure 4. The design of the deep learning model.

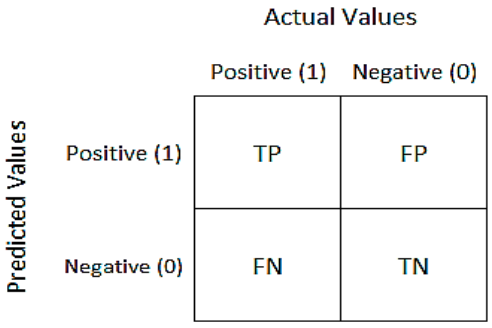

Figure 5. Confusion matrix architecture.

curve, which shows the changes in the loss function (model errors) over time during the training phase. As the loss decreases, the performance of the model improves.

The other learning curve is the accuracy curve, which measures the performance of the model during the training epochs, with increasing values of this curve indicating an improvement in the capabilities of the model. Figure 6 shows the accuracy and loss curves (only with the validation data) for the training data.

As shown in Figure 6, the performance of the model increases over the training epochs, with both the training and validation metrics trending upward. This indicates that the model's performance improves on both the training (visible) and validation (unseen) data, suggesting successful convergence and better generalization, thereby mitigating the overfitting issue. The best accuracy of 99% was achieved for the training data and 97% for the validation data. Figure 7 presents the results of the CM generated by the proposed model.

The main diagonal elements of the confusion matrix indicate the correct classifications performed by the model.

The confusion matrix results show that the model accurately classified 3831 samples from class 0 (hate speech), 3505 samples from class 1 (neither), and 3844 samples from class 2 (offensive).

Elements that are outliers of the main diagonal within the confusion matrix indicate misclassifications. For instance, the element in the second row, first column (129) indicates that 129 samples of class 1 (neither) were misclassified as class 0 (hate-speech). Table 1 shows the model's results based on precision, recall, and F1-score for each class.

Table 1. The model's results based on precision, recall, and F1-score for each class.

	Precision	Recall	F1-score
Hate speech	0.96	1.00	0.98
Neither	0.99	0.93	0.96
Offensive	0.96	0.99	0.98

Precision: This metric measures the precision of the positive predictions made by the model; in other words, how many correct predictions the model made for a given class relative to the total number of predictions for that class. For example, the model achieved a high precision of 0.99 for the Neither class, meaning that when it predicted a sample as belonging to the Neither class, it was 99% correct.

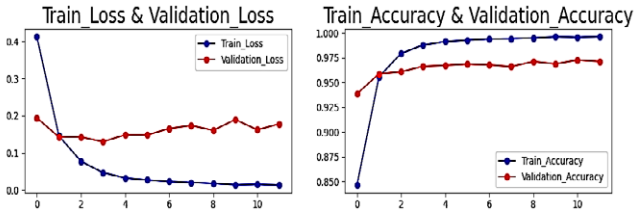

Figure 6. The accuracy and loss curves for both the training and validation processes for the proposed model.

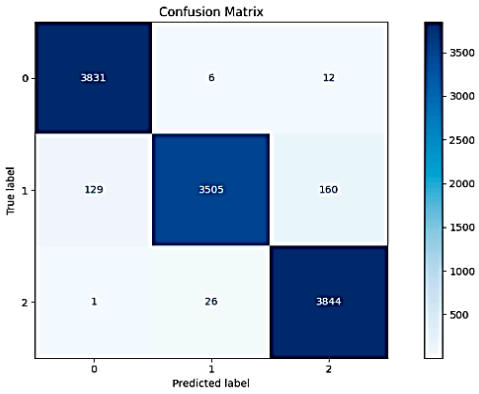

Figure 7. The results of the confusion matrix achieved by the proposed model.

Recall: This metric tells us the proportion of actual cases of a given sample type that were correctly identified by the model. For example, the model achieved a recall of 1.00 for class 1 (Hate-speech), meaning that it correctly identified 100% of all hate speech cases in the test dataset.

F1-score: This measure is the harmonic mean of precision and recall and provides a balanced measure of model performance. This measure can be used to determine the overall performance of the model, and based on the results shown in Table 1, it can be said that the model performs well in classifying the three classes (Hate speech, Neither, Offensive), achieving high F1-score values of 0.98, 0.96, and 0.98, respectively. Table 2 shows the values of precision, recall, and F1-score for the machine learning algorithms.

Table 2. The values of precision, recall, and F1-score for the machine learning algorithms.

Decision Tree			
	Precision	Recall	F1-score
Hate speech	0.92	0.96	0.94
Neither	0.94	0.89	0.92
Offensive	0.95	0.97	0.96
Random Forest			
	Precision	Recall	F1_Score
Hate speech	0.90	0.79	0.84
Neither	0.80	0.86	0.83
Offensive	0.84	0.88	0.86
Support Vector Machine			
	Precision	Recall	F1_Score
Hate speech	0.52	0.86	0.65
Neither	0.92	0.88	0.90
Offensive	0.75	0.29	0.42

Based on the results obtained, we note that:

- In terms of precision, recall, and so forth, the suggested model outperforms all other models, and F1 score, indicating its strong ability to perform the classification process with high accuracy.
- The decision tree model performs relatively well but fails slightly in recall compared to the proposed model.
- The random forest model has good performance but also suffers in recall, especially for the hate speech category.
- The SVM model performs the worst, especially in detecting offensive speech.

For each category, we note that:

- Hate speech class:
- The proposed model achieves the highest recall (1.00), meaning it identifies all hate speech cases with no false negatives.
- Decision Tree performs well but shows lower precision, which means that it may misclassify neither content as hate speech.
- Random Forest struggles with recall (0.79), which means it is missing a significant number of hate speech cases.
- SVM has very low precision (0.52), indicating frequent false positives.

Neither class:

- The proposed model has the highest precision (0.99). It accurately classifies the “Neither” class with very few false positives.
- Decision Tree has balanced performance but a slightly lower recall (0.89), hence it misses some samples.
- Random Forest has lower precision (0.80).
- SVM performs well for this class, with a high F1-score (0.90), but recall is still lower than that of the proposed model.

Offensive class:

- The model’s precision outperforms that of all the proposed models (0.96), with a recall of 0.99 and minimal false negatives.
- Decision Tree performs well. Its recall is slightly lower than that of the proposed model (0.97).
- Random Forest has lower precision (0.84) than the other classes; therefore, it has more false positives.
- SVM fails significantly in this category with poor recall (0.29). It misses a lot of offensive content.

By comparing all models in terms of overall accuracy, we obtained the results shown in Figure 8.

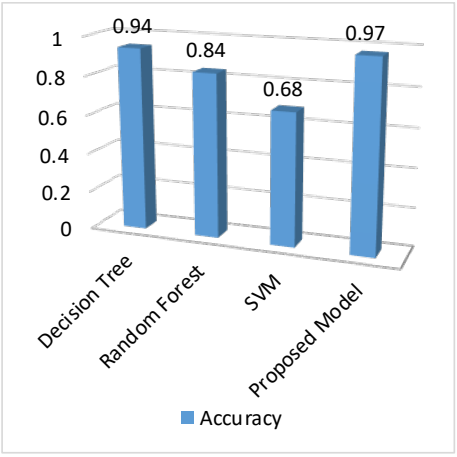

Figure 8. The accuracy value for all models.

5. Conclusions and Future Work

Hate speech is understood as an expression that encourages discrimination, enmity, and violence against individuals or groups based on characteristics such as race, religion, gender, or sexual orientation. It has

become a growing concern, making the spread of such content a major challenge to online communities and social cohesion. In this research, we worked to address the issues related to the spread of hate speech on Twitter, where a proposed deep learning model was used to classify tweet text and identify whether the tweet contains hate speech or abusive speech. A set of preprocessing operations was performed on the tweet texts before they were entered into the proposed deep learning model. The proposed model showed superior performance, achieving the highest accuracy of 97%, outperforming traditional machine learning algorithms such as decision tree, random forest, and support vector machine (SVM). The proposed model showed high precision, recall, and F1-score across all categories, while decision tree and random forest showed reasonable performance but struggled with precision and recall for some categories, and SVM performed poorly, especially in detecting offensive content. There are several areas for improvement and future exploration. Future work could focus on incorporating advanced deep learning techniques such as transformer-based models like BERT and RoBERTa to enhance contextual understanding of text and capture the semantic nuances inherent within tweet text. Additionally, we could improve hyperparameters and explore other feature extraction techniques which could improve the model's generalization capabilities.

Funding

This research received no external funding.

Conflict of interest

The authors declare that they have no conflict of interest.

References

Alkomah, F., & Ma, X. (2022). A literature review of textual hate speech detection methods and datasets. Information, 13(6), 273.
https://doi.org/10.3390/info13060273

Bataineh, A. Q., Abu-AlSondos, I. A., Almazaydeh, L., El Mokdad, S. S., & Allahham, M. (2023). Enhancing natural language processing with machine learning for conversational AI. In IET Conference Proceedings CP870 (Vol. 2023, No. 39, pp. 229-237). Stevenage, UK: The Institution of Engineering and Technology.
https://doi.org/10.1049/icp.2024.0492

Chai, C. P. (2023). Comparison of text preprocessing methods. Natural language engineering, 29(3), 509-553.
https://doi.org/10.1017/S1351324922000213

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.
https://doi.org/10.1613/jair.953

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In Proceedings of the international AAAI conference on web and social media (Vol. 11, No. 1, pp. 512-515).
https://doi.org/10.1609/icwsm.v11i1.14955

Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. Journal of artificial intelligence research, 61, 863-905.
https://doi.org/10.1613/jair.1.11192

Fortuna, P., Soler-Company, J., & Wanner, L. (2021). How well do hate speech, toxicity, abusive and offensive language classification models generalize across datasets?. Information Processing & Management, 58(3), 102524.
https://doi.org/10.1016/j.ipm.2021.102524

Haghighi, S., Jasemi, M., Hessabi, S., & Zolanvari, A. (2018). PyCM: Multiclass confusion matrix library in Python. Journal of Open Source Software, 3(25), 729.
https://doi.org/10.21105/joss.00729

Hasanin, T., Khoshgoftaar, T. M., Leevy, J. L., & Seliya, N. (2019). Examining characteristics of predictive models with imbalanced big data. Journal of Big Data, 6(1), 69.
https://doi.org/10.1186/s40537-019-0231-2

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning: C. Janiesch et al. Electronic markets, 31(3), 685-695.
https://doi.org/10.1007/s12525-021-00475-2

Lescoat, T., Ovsjanikov, M., Memari, P., Thiery, J. M., & Boubekeur, T. (2018). A survey on data-driven dictionary-based methods for 3D modeling. In Computer Graphics Forum (Vol. 37, No. 2, pp. 577-601).
https://doi.org/10.1111/cgf.13384

Lin, Y. (2022). Social media for collaborative planning: A typology of support functions and challenges. Cities, 125, 103641.
https://doi.org/10.1016/j.cities.2022.103641

MacEachern, S. J., & Forkert, N. D. (2021). Machine learning for precision medicine. Genome, 64(4), 416-425. https://doi.org/10.1139/gen-2020-0131

Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. Global Transitions Proceedings, 3(1), 91-99. https://doi.org/10.1016/j.gltp.2022.04.020

Malmasi, S., & Zampieri, M. (2017). Detecting hate speech in social media. arXiv preprint arXiv:1712.06427. https://doi.org/10.48550/arXiv.1712.06427

Markoulidakis, I., Kopsiaftis, G., Rallis, I., & Georgoulas, I. (2021). Multi-class confusion matrix reduction method and its application on net promoter score classification problem. In Proceedings of the 14th PErvasive technologies related to assistive environments conference (pp. 412-419). https://doi.org/10.1145/3453892.3461323

Meier, A., & Johnson, B. K. (2022). Social comparison and envy on social media: A critical review. Current opinion in psychology, 45, 101302. https://doi.org/10.1016/j.copsyc.2022.101302

Mossie, Z., & Wang, J. H. (2020). Vulnerable community identification using hate speech detection on social media. Information Processing & Management, 57(3), 102087. https://doi.org/10.1016/j.ipm.2019.102087

Pamungkas, E. W., Basile, V., & Patti, V. (2021). A joint learning approach with knowledge injection for zero-shot cross-lingual hate speech detection. Information Processing & Management, 58(4), 102544. https://doi.org/10.1016/j.ipm.2021.102544

Pereira-Kohatsu, J. C., Quijano-Sánchez, L., Liberatore, F., & Camacho-Collados, M. (2019). Detecting and monitoring hate speech in Twitter. Sensors, 19(21), 4654. https://doi.org/10.3390/s19214654

Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., & Gehler, P. (2022). Towards total recall in industrial anomaly detection. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 14298-14308). IEEE. https://doi.org/10.1109/CVPR52688.2022.01392

Roy, P. K., Tripathy, A. K., Das, T. K., & Gao, X. Z. (2020). A framework for hate speech detection using deep convolutional neural network. IEEE Access, 8, 204951-204962. https://doi.org/10.1109/ACCESS.2020.3037073

Sabri, T., El Beggar, O., & Kissi, M. (2022). Comparative study of Arabic text classification using feature vectorization methods. Procedia Computer Science, 198, 269-275. https://doi.org/10.1016/j.procs.2021.12.239

Samoshyn, A. (2020). Hate speech and offensive language dataset. Kaggle. https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset

Sarica, S., & Luo, J. (2021). Stopwords in technical language processing. Plos one, 16(8), e0254937. https://doi.org/10.1371/journal.pone.0315195

Wang, J., Huang, J. X., Tu, X., Wang, J., Huang, A. J., Laskar, M. T. R., & Bhuiyan, A. (2024). Utilizing bert for information retrieval: Survey, applications, resources, and challenges. ACM Computing Surveys, 56(7), 1-33. https://doi.org/10.1145/3648471

Widianto, A., Pebriyanto, E., Fitriyanti, F., & Marna, M. (2024). Document similarity using term frequency-inverse document frequency representation and cosine similarity. Journal of Dinda: Data Science, Information Technology, and Data Analytics, 4(2), 149-153. https://doi.org/10.20895/dinda.v4i2.1589

William, P., Gade, R., esh Chaudhari, R., Pawar, A. B., & Jawale, M. A. (2022). Machine learning based automatic hate speech recognition system. In 2022 International conference on sustainable computing and data communication systems (ICSCDS) (pp. 315-318). IEEE. https://doi.org/10.1109/ICSCDS53736.2022.9760959