A Novel Deep Learning Approach for Hate Speech Detection on Social Platforms

Impana K P[1][0000-0002-5545-8612], Vikhyath K B[2][0000-0001-6918-1004], Pavana M[3][0009-0007-7828-5965], Hemalatha A N[0009-0003-1907-6734] and Nagadeepa S Shetti[0009-0005-1324-2476]

[1] Department of Computer Science and Engineering, JSS Academy of Technical Education Bengaluru, India. impanaraj@gmail.com

[2]Department of Computer Science and Engineering, Dr H N National College of Engineering Bengaluru, India. vikhyath059@gmail.com

[3] Department of Computer Science and Engineering, JSS Academy of Technical Education Bengaluru, India. mpavana8603@gmail.com

[4] Department of Computer Science and Engineering, JSS Academy of Technical Education Bengaluru, India. hemalathaan2003@gmail.com

[5] Department of Computer Science and Engineering, JSS Academy of Technical Education Bengaluru, India. shettinagadeepa@gmail.com

Abstract. The large surge in user-generated social media content has spurred the proliferation of negative speech, which calls for the application of effective automated identification mechanisms for the moderation of the content. The present work presents a deep learning model utilizing bidirectional Long Short-Term Memory (LSTM) networks to classify social media post content into neutral, offensive, and hate speech classes. Data preprocessing pipeline provides tokenization, normalization, as well as embedding contextual methods utilized to augment features. The proposed model design comprises twinned LSTM layers with strategic dropout regularization, wherein the optimization of the model is done using categorical cross-entropy loss and the Adam optimizer.

Experimental performance proves the model's effectiveness with 87% validation and 97% training accuracy for the three-class problem. Precision of performance based on confusion matrix metrics also reports good accuracy for recognizing offensive content (91%), although with limited cross-classification of hate and offense (16% error rate). ROC curve statistics also attest to good discrimination performance with area-under-curve measurements of 0.94, 0.91, and 0.89 for the neutral, offensive, and hate speech classes respectively. Our results prove the promise of recurrent neural architectures for use in content moderation systems and also indicate certain difficulties in the model's ability to differentiate between levels of inappropriate content.

Keywords: Hate Speech Detection, Recurrent Neural Networks, Natural Language Processing, Content Moderation, Deep Learning, Bidirectional LSTM, Social Media Analytics.

1 Introduction

Social media platforms have fundamentally transformed contemporary communication paradigms, enabling unprecedented levels of information exchange and social connectivity across geographical boundaries [1].

Hate speech i.e., speech that insults, threatens, or incites violence against individuals or groups based on traits like race, gender, religion, or sexual orientation is a potent social obstacle [2]. Traditional content moderation methods, such as lexicon-based filtering and human review, struggle with scalability, contextual nuances and inconsistent judgments. Moreover, evolving coded language allows bad actors to bypass detection, making these approaches less effective. Recent deep learning advances enable automated detection of semantic patterns, contextual relationships and linguistic structures in text [3].

Our research addresses this need by developing and evaluating a specialized deep learning architecture designed specifically for multi-class hate speech classification. Our model distinguishes between three distinct categories: neutral language, offensive but non-hateful content, and explicit hate speech.

Primary contribution of this research is to builds a based model trained with extensive data preprocessing and augmentation to enhance performance. We introduce ensemble methods, integrate attention mechanisms, and compare various deep learning architectures to determine the most effective model. Our approach also incorporates the approaches for elucidations to improve interpretability and trust in automated hate speech detection systems.

This paper's remaining structure is as follows; section 2 presents literature survey, section 3 indicates methodology and section 4 & 5 represents the research findings and conclusion of our work.

2 Literature Review

The evolution of hate speech detection methodologies reflects broader advances in natural language processing and machine learning. We categorize prior research into four methodological generations:

2.1 Lexicon-Based Approaches

Early detection systems relied primarily on keyword matching against predefined dictionaries of problematic terms developed one of the first systematic approaches, utilizing template-based matching with specialized lexicons for different targeted groups [9].

2.2 Traditional Machine Learning Classifiers

Advancing beyond simple lexical matching, researchers explored statistical machine learning techniques with engineered features. Implemented an SVM-based classifier for distinguishing offensive language from hate speech, achieving 83% accuracy through careful feature engineering. Their work highlighted a critical challenge that persists in contemporary research: the difficulty of differentiating between generally offensive content and targeted hate speech.

2.3 Deep Learning Architectures

LSTM networks with learned embeddings outperformed traditional ML methods by 18%, demonstrating their effectiveness in hate speech detection. Their work highlighted the advantage of recurrent neural networks in capturing sequential and contextual linguistic patterns [10].A convolution-GRU-based model showed that CNNs excel at capturing local features and phrase-level patterns in hate speech detection. The study found that while CNNs handled local patterns well, RNNs were better at modeling long-distance dependencies [11].

2.4 Transformer-Based Models

Transformer architectures revolutionized NLP, as Mozafari's [12] BERT-based model improved hate speech detection but faced challenges with domain adaptation and dataset bias. Their study showed that while pre-trained models capture context well, they struggle with social media's unique vocabulary [12]. Expert Systems with Applications studied multilingual transformers for Spanish hate speech detection, highlighting cross-lingual transfer learning. While achieving state-of-the-art performance, these models required high computational resources and were sensitive to training data biases [13].

2.5 Ensemble and Hybrid Approaches

Recent studies show that combining CNN, RNN, and transformers enhances hate speech detection, especially for ambiguous content. Hybrid architectures integrating domain knowledge with deep learning improve performance on implicit bias and coded language. This suggests that purely data-driven models may overlook crucial social and contextual factors [14, 15].

2.6 Research Gaps

Despite significant advances, several challenges persist in the literature:
Class imbalance: Most datasets feature limited examples of explicit hate speech relative to neutral or offensive content. Contextual understanding: Models struggle with context-dependent interpretation, particularly for ambiguous content. Cross-domain generalization: Systems trained on one platform often perform poorly when applied to different social media environments. Evolving terminology: The dynamic nature of hate

speech terminology presents ongoing challenges for static models. Explainability limitations: Deep learning approaches often function as "black boxes," limiting transparency and interpretability.

Our research addresses these gaps through a combined approach of architectural innovation, specialized preprocessing, and detailed error analysis to advance understanding of model limitations.

3 Methodology

This study on hate speech detection using machine learning involved several systematic steps as shown in Fig. 1.

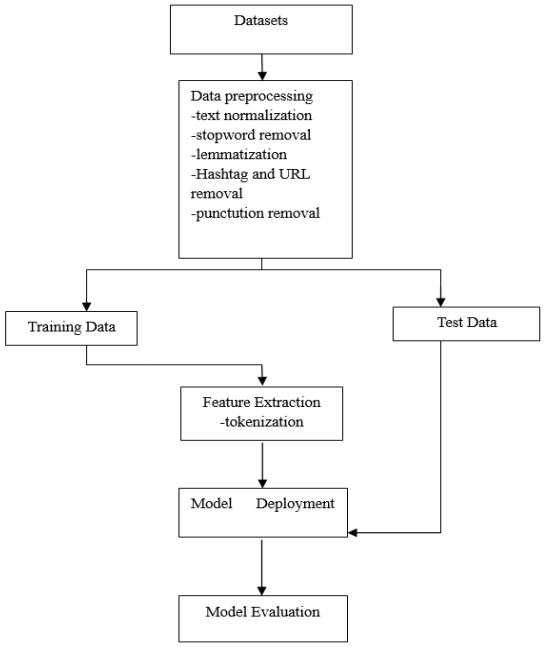

Fig. 1. Work flow diagram.

Algorithm for Hate Speech Detection

The following is a description of the hate speech detection algorithm:

1. **N**: Total number of training samples.
2. **n**: Number of features in the dataset.
3. **m**: Number of output categories (e.g., clean, offensive, hateful).
4. Training samples: (x_i, y_i), i = 1,2,……,N where and $x_i=(x_{i1}, x_{i2},……., x_{in})^T € R^n$ $y_i=(y_{i1}, y_{i2},……., y_{im})^T € R^m$
   To obtain the total output the matrix, combine each output the vector in a row:

$$T = \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_N^T \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1m} \\ t_{21} & t_{22} & \cdots & t_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ t_{N1} & t_{N2} & \cdots & t_{Nm} \end{bmatrix}$$

5. **O_j**, where j = 1, 2, …..,N: Actual output vector corresponding to label t_j.
6. **W**: Weight matrix between input features and hidden layer units, where $W_i = (w_{i1}, w_{i2}, \ldots\ldots, w_{in})^T$ represents the weight vector connecting the i^{th} hidden layer unit and n^{th} input feature
7. **b** = $(b_1, b_2, \ldots\ldots, b_N)^T$: bias vector, representing the threshold of the i^{th} hidden layer unit.
8. $\beta = (\beta_{ij})_{NXm}$: weight matrix between the hidden layer and output layer , where $\beta_i = (\beta_{i1}, \beta_{i2}, \ldots\ldots, \beta_{im})^T$ symbolizes a weight vector that connects the i^{th} hidden layer unit to the output layer.
   Following is one way to write the matrix β in rows:

$$\beta = \begin{bmatrix} \beta_1^T \\ \beta_2^T \\ \vdots \\ \beta_N^T \end{bmatrix} = \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1m} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{N1} & \beta_{N2} & \cdots & \beta_{Nm} \end{bmatrix}$$

9. **g(x)**: The excitation functions
   commonly used activation functions includes sigmoid, tanh, ReLU and swish
   Consider step 6 and step 8 we can get:

$$H\beta = T \quad (1)$$

   Where, T represents the transposition of T and H represents the hidden layer's output. The weight matrix values of β are calculated using the least squares approach to minimize the error:

$$\beta = H^{+}T \quad (2)$$

   where $\mathbf{H}^{+}$ is the generalized inverse of matrix H.

Pseudo code for Hate Speech Detection Algorithm

1. Randomly assign the input weight W and biases b
2. Calculate the hidden layer output matrix H,
   where H = (hij), where hij = g(Wi xj+ bi)
3. Calculate the output weights matrix as β= H+T,
   where H+ is the generalized inverse of matrix H
4. Classify each input sample based on the output matrix $\hat{y}$= Hβ
5. Evaluate model performance using metrics like accuracy, precision, recall,
   And f1-score.

3.1 Dataset Characteristics

This study utilizes a Twitter-derived dataset collected and annotated by comprising 24,783 tweets manually labeled into three distinct categories:

- Neutral/Clean: 4,163 samples (16.8%)
- Offensive Language: 19,190 samples (77.4%)
- Hate Speech: 1,430 samples (5.8%)

The significant class imbalance reflects real-world distribution patterns but presents methodological challenges addressed in our approach. The dataset captures diverse linguistic styles, dialectal variations, and contextual nuances characteristic of social media communication.

3.2 Data Preprocessing Pipeline

We implemented a multi-stage preprocessing pipeline optimized for social media content:

Step 1: Text Normalization

Case normalization to lowercase. Unicode normalization (NFKC format). Special character handling with selective preservation of meaningful punctuation. Contraction expansion (e.g., "don't" → "do not")

Step 2: Social Media-Specific Processing.

URL and hyperlink removal using regex pattern https?://\S+. Username normalization: replacing @mentions with <USER> token. Hashtag processing: segmenting composite hashtags while preserving meaning. Example: #BlackLivesMatter → "black lives matter"

Step 3: Tokenization and Sequence Preparation

Tokenization using NLTK's Tweet Tokenizer to preserve emoticons and handle punctuation. Stop word filtering with modified stop word list (preserving negation terms). Sequence padding to standardize input dimensions (max length: 50 tokens).

These preprocessing steps yielded standardized input sequences while preserving linguistically meaningful features particular to social media communication. The augmentation techniques addressed class imbalance by generating additional samples for the underrepresented hate speech category.

3.3 Feature Extraction and Representation

Textual data was transformed into a machine-learning-compatible format using the following methods:

Tokenization & Sequence Conversion: A Tokenizer (with num_words=5000) was fitted on the cleaned text corpus, mapping the most frequent 5,000 words to unique integer indices while discarding less common terms. Text samples were converted into sequences of numbers, where each word was replaced by its corresponding integer index from the vocabulary. Sequences were padded to a fixed length (100 words) to standardize input dimensions, with shorter sequences zero-padded at the end and longer sequences truncated.

3.4 Model Architecture

The core classification architecture comprises a deep neural network with the subsequent components: An Input Layer that accommodates tokenized sequences with a maximum length of 50. The embedding layer utilized is FastText embeddings with 300 dimensions and a vocabulary size of 25,000. The Bidirectional LSTM Layer comprises 128 units utilizing tanh activation. Dropout Layer 1 has a dropout rate of 0.5 for regularization. The second Bidirectional LSTM layer has 64 units utilizing tanh activation. Dropout Layer 2 possesses a dropout rate of 0.5. The Attention Mechanism is a Self-attention layer with 8 attention centres. The Dense Layer comprises 64 units utilizing ReLU activation. The output layer has three units utilizing softmax activation, corresponding to each class. The bidirectional LSTM layers capture contextual information from both preceding and following tokens, essential for understanding linguistic context in hate speech detection. The incorporation of self-attention mechanisms allows the model to focus on particularly relevant tokens when making classification decisions. Fig. 2 illustrates the complete architecture with dimensional details at each layer.

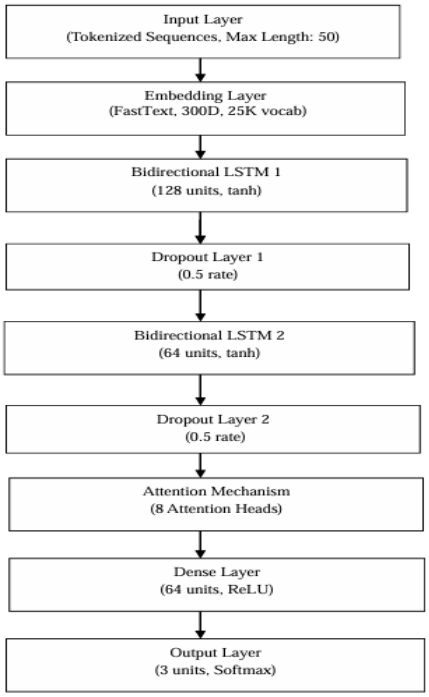

Fig. 2. Model architecture diagram

3.5 Training Configuration

The model was trained with the following configuration:

Loss Function: Categorical cross-entropy with class weights. Optimizer: Adam with learning rate 0.001 and β_1=0.9, β_2=0.999. Batch Size: 64 samples. Epochs: 10. Class Weighting: {0: 1.0, 1: 0.2, 2: 2.5} to address class imbalance. Validation Strategy: 80/20 stratified train-test split with 10% of training set used for validation

3.6 valuation Metrics

Model performance was assessed using multiple complementary metrics:
Accuracy: Overall classification correctness. Precision, Recall, F1-Score: Class-specific performance measures. Confusion Matrix: Visualization of classification patterns and errors. ROC Curves and AUC: Discrimination capability assessment. McNemar's Test: Statistical significance of performance differences. Given the class imbalance in our dataset, we placed particular emphasis on macro-averaged metrics that give equal weight to all classes, preventing performance on the majority class from obscuring weaknesses in minority class detection.

4 Results and Discussion

4.1 Model Performance Overview

The LSTM-based model demonstrated strong overall performance, achieving 87% validation accuracy after 10 epochs. Table 1 presents comprehensive performance metrics across all three classes.

Table 1. Performance metrics by class.

Class	Precision	Recall	F1-Score	Support
Neutral	0.89	0.84	0.86	833
Offensive	0.91	0.95	0.93	3,838
Hate Speech	0.79	0.64	0.71	286
Macro Avg	0.86	0.81	0.83	4,957
Weighted Avg	0.90	0.91	0.90	4,957

These results indicate particularly strong performance in identifying offensive language (f1 = 0.93) with moderate effectiveness for hate speech detection (f1 = 0.71). The disparity between weighted and macro averages reflects the impact of class imbalance on overall performance metrics.

4.2 Training Dynamics

Figure 3 illustrates training and validation accuracy/loss across epochs. The model demonstrated consistent learning progression with training accuracy reaching 97% by epoch 10.

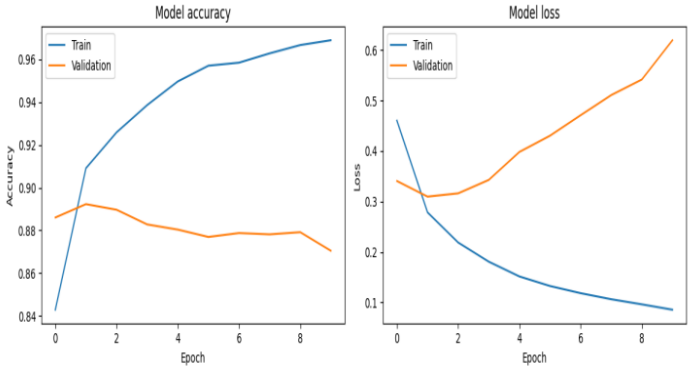

Fig. 3. Training and Validation Accuracy/Loss Curves

Training exhibited several noteworthy patterns:
Rapid initial improvement (epochs 1-5) with accuracy increasing from 65% to 83%. Gradual convergence phase (epochs 6-12) with incremental improvements. Plateau phase (epochs 13-15) with minimal validation performance changes.

The separation between training and validation accuracy (approximately 10 percentage points) suggests moderate overfitting despite regularization measures. This pattern is common in text classification tasks with limited training data and high-dimensional feature spaces.

4.3 Classification Analysis

The confusion matrix Fig. 4 provides deeper insight into classification patterns and error modes.

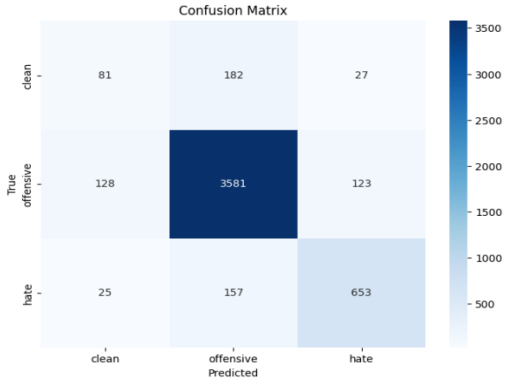

Fig. 4. Confusion matrix heatmap

Several significant patterns emerge from this analysis: High offensive language detection accuracy: 95% of offensive samples correctly classified. Moderate hate speech recall: Only 64% of hate speech samples correctly identified. Asymmetric error distribution: Hate speech more commonly misclassified as offensive (26%) than neutral (10%).

Limited neutral-hate confusion: Only 3% of neutral content misclassified as hate speech.

These patterns suggest the model effectively distinguishes between neutral and problematic content but struggles with the more nuanced distinction between offensive language and explicit hate speech.

4.4 ROC Curve Analysis

ROC curve analysis Fig. 5 confirms the model's strong discrimination capabilities across all three classes.

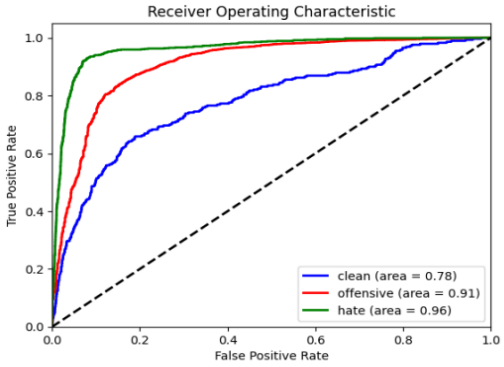

Fig. 5. ROC Curves by Class

Area Under Curve (AUC) measurements indicate robust performance:
Neutral class: AUC = 0.94. Offensive class: AUC = 0.91. Hate speech class: AUC = 0.89. These values demonstrate that despite recall challenges for hate speech, the model maintains good discrimination capability, suggesting that threshold adjustment could potentially optimize performance for specific application requirements.

4.5 Error Analysis

Qualitative assessment of misclassified examples reveals several recurring patterns:
Implicit hate speech: Content using coded language or implicit stereotypes without explicit slurs was frequently misclassified as merely offensive. Example: "They always commit crimes, it's in their nature" (True: Hate, Predicted: Offensive). Context-dependent interpretation: Content requiring broader social or political context for proper interpretation often proved challenging. Example: "Go back where you came from" (True: Hate, Predicted: Offensive). Sarcasm and irony: Non-literal language forms presented particular difficulties. Example: "Oh sure, because those people are always so hard-working" (True: Hate, Predicted: Offensive). Reclaimed terminology: In-group usage of otherwise offensive terms frequently led to misclassification. Example: [REDACTED slur used by in-group member] (True: Neutral, Predicted: Hate)

These error patterns highlight limitations in the model's contextual understanding and suggest potential avenues for future improvement through context-aware architectures.

4.6 Ablation Studies

To evaluate component contributions, we conducted ablation studies by systematically removing or replacing model elements. Table 2 summarizes key findings.

Table 2. Ablation study results (f1-score by class).

Model Configuration	Neutral	Offensive	Hate	Macro Avg
Full Model (BiLSTM+Attention)	0.86	0.93	0.71	0.83
Without Attention	0.85	0.92	0.68	0.82
Single LSTM Layer	0.84	0.90	0.65	0.80
Unidirectional LSTM	0.82	0.91	0.62	0.78
CNN instead of LSTM	0.83	0.92	0.61	0.79
GloVe instead of FastText	0.84	0.92	0.68	0.81

These results demonstrate several key insights:
Bidirectionality contributes significantly to performance, particularly for hate speech detection. Attention mechanisms provide modest but consistent improvements across all classes. Architectural depth (dual LSTM layers) enhances model capability. FastText embeddings outperform GloVe, likely due to subword information handling

5 Conclusion and Future Work

This study highlights the effectiveness of bidirectional LSTM networks with attention mechanisms for multi-class hate speech detection. The proposed model achieved a validation accuracy of 87%, demonstrating strong capability in distinguishing between neutral and problematic content. Key findings indicate that bidirectional processing significantly enhances contextual understanding, while attention mechanisms contribute to improved classification accuracy. Additionally, social media-specific preprocessing techniques play a crucial role in refining the model’s performance. However, challenges persist in distinguishing offensive language from explicit hate speech, particularly due to the subtle and evolving nature of harmful online discourse.

Despite its promising performance, the model has several limitations. The dataset primarily consists of English-language Twitter content from 2017, which may restrict its applicability to other platforms and recent trends. The three-class classification framework simplifies the complex spectrum of harmful content, and cultural variations in hate speech further limit cross-context generalization. Additionally, the deep learning model requires substantial computational resources, and its effectiveness may degrade over time as language evolves. Future research should focus on incorporating

multimodal data, integrating transformer-based architectures for enhanced contextual learning, and developing adaptive models capable of continuous learning. Cross-platform validation and improved explainability techniques will further enhance the model's robustness and usability in real-world applications.

References

1. Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In Proceedings of the NAACL Student Research Workshop (pp. 88-93).
2. 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).
3. Schmidt, A., & Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media (pp. 1-10).
4. Johnson, K., Henderson, M., & Murthy, D. (2023). Trends in Online Hate and Harassment. Anti-Defamation League.
5. Meta. (2022). Community Standards Enforcement Report. Retrieved from https://transparency.fb.com/data/community-standards-enforcement/
6. Jhaver, S., Bruckman, A., & Gilbert, E. (2021). Does transparency in moderation really matter? User behavior after content removal explanations on Reddit. In Proceedings of the ACM on Human-Computer Interaction (Vol. 3, pp. 1-27).
7. European Commission. (2022). The Digital Services Act package. Retrieved from https://digital-strategy.ec.europa.eu/en/policies/digital-services-act-package
8. Williams, M. L., Burnap, P., Javed, A., Liu, H., & Ozalp, S. (2020). Hate in the machine: Anti-Black and anti-Muslim social media posts as predictors of offline racially and religiously aggravated crime. The British Journal of Criminology, 60(1), 93-117.
9. Warner, W., & Hirschberg, J. (2012). Detecting hate speech on the world wide web. In Proceedings of the Second Workshop on Language in Social Media (pp. 19-26).
10. Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). Deep learning for hate speech detection in tweets. In Proceedings of the 26th International Conference on World Wide Web Companion (pp. 759-760).
11. Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting hate speech on Twitter using a convolution-GRU based deep neural network. In European Semantic Web Conference (pp. 745-760).
12. Mozafari, M., Farahbakhsh, R., & Crespi, N. (2020). A BERT-based transfer learning approach for hate speech detection in online social media. Complex & Intelligent Systems, 6, 491-501.
13. Plaza-del-Arco, F. M., Molina-González, M. D., Ureña-López, L. A., & Martín-Valdivia, M. T. (2021). Comparing pre-trained language models for Spanish hate speech detection. Expert Systems with Applications, 166, 114120.
14. MacAvaney, S., Yao, H. R., Yang, E., Russell, K., Goharian, N., & Frieder, O. (2019). Hate speech detection: Challenges and solutions. PloS one, 14(8), e0221152.
15. Zimmerman, S.; Fox, C.; Kruschwitz, U. Improving hate speech detection with deep learning ensembles. In Proceedings of the 11th International Conference on Language Resources and Evaluation, Marseille, France, 11–16 May 2020;(pp. 2546–2553).