International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I7P120

Volume 72 Issue 7, 187-197, July 2024

Original Article

Novel Approach to Offensive Language Detection on Social Media: Tree CNN Integration with Adversarial Bi-LSTM

V. Uma Maheswari[1], R. Priya[2]

[1]*Department of Computer Science, School of Computing Sciences, Vels Institute of Science, Technology & Advanced Studies (VISTAS), Chennai, India*
[2]*Department of Computer Applications, School of Computing Sciences, Vels Institute of Science, Technology & Advanced Studies (VISTAS), Chennai, India.*

[1]*Corresponding Author : uma219ravi@gmail.com*

Received: 04 February 2024 Revised: 30 May 2024 Accepted: 15 June 2024 Published: 26 July 2024

Abstract *- Offensive language detection on social media platforms is a crucial task for maintaining a healthy online environment and ensuring user safety. Traditional methods often struggle to effectively capture the nuances and dynamic nature of offensive language in such diverse and rapidly evolving contexts. Our research presents a novel approach to offensive language detection on social media by integrating Tree Convolutional Neural Networks (CNN) with Adversarial Bidirectional Long Short-Term Memory (Bi-LSTM) networks. We address the challenges of imbalanced data and semantic understanding by employing the Synthetic Minority Over-sampling Technique (SMOTE) and Word2Vec for feature extraction. To enhance model interpretability and focus on relevant features, we incorporate attention mechanisms within both the Tree CNN and the Adversarial Bi-LSTM. We utilize two attention mechanisms: one for identifying repetitive patterns using an Entropy Pruning Method and another for error loss monitoring during training. This dual attention mechanism enables our model to effectively distinguish offensive from non-offensive text while also providing insights into model decisions. Experimental findings on benchmarking social media datasets show that our suggested methodology outperforms cutting-edge approaches in terms of accuracy and interpretation. Our study contributes to advancing offensive language detection techniques on social media platforms and provides a framework for developing more reliable and interpretable models for online content moderation.*

Keywords *- Offensive language detection, Social media, Tree CNN, Adversarial Bi-LSTM, Imbalanced data handling, SMOTE, Word2Vec, Attention mechanisms.*

1. Introduction

Social media networks have evolved into integral parts of modern communication, providing users with unprecedented opportunities for interaction and expression. However, alongside the benefits of connectivity and information sharing, these platforms also harbor significant challenges, particularly concerning offensive language [1]. The detection and mitigation of offensive content on social media are essential for maintaining a safe and inclusive online environment, protecting users from harm, and upholding community standards [2]. Traditional methods of offensive language detection often fall short of adequately addressing the dynamic and context-dependent nature of offensive language in the vast and diverse landscape of social media [3]. In response to these challenges, our research proposes a novel approach to offensive language detection on social media, leveraging advanced deep learning techniques. We introduce the integration of Tree Convolutional Neural Networks (CNN) [4] with Bi-LSTM [5] networks to tackle the complexities of offensive language detection in this unique context. This integration harnesses the strengths of both architectures, allowing for the exploitation of hierarchical structures in social media conversations and the capture of temporal dependencies in language use. One of the primary challenges in offensive language detection on social media is the imbalance between offensive and non-offensive content. To deal with this issue, we employ the SMOTE [6], a widely used method for balancing imbalanced datasets. Additionally, we utilize Word2Vec [7, 8] for feature extraction, enabling the model to capture semantic information and understand the nuances of language. Interpretability is crucial for understanding model decisions and gaining insights into the features driving classification. To enhance interpretability, we incorporate attention mechanisms [9] within both the Tree CNN and the Adversarial Bi-LSTM. These attention mechanisms serve multiple purposes: identifying repetitive

patterns indicative of offensive language, monitoring error loss during training, and providing explanations for model predictions [10]. Our proposed approach offers several key advantages over existing methods. Firstly, it demonstrates superior performance in terms of both accuracy and interpretability, as evidenced by experimental results on benchmark social media datasets. Secondly, by leveraging advanced DL techniques, our model can adapt to the dynamic nature of social media language and effectively distinguish offensive from non-offensive text in diverse contexts.

Offensive language detection on online platforms is critical for cultivating a respectful virtual community. While natural language processing has progressed, prevailing strategies commonly fail to apprehend the intricate and shifting character of toxic speech fully. This is attributable to complications, including skewed data, the multifaceted settings where harmful language surfaces, and the necessity for contextual comprehension. Moreover, additional roadblocks derive from code that emphasizes efficiency over empathy, disregarding the diversity of human experiences and perspectives. By acknowledging such limitations and actively pursuing interdisciplinary cooperation, technologists can better serve all users with solutions that prioritize inclusion over speed alone. Our research addresses these gaps by integrating Tree CNN with Adversarial Bi-LSTM networks. This novel approach aims to improve detection accuracy and model interpretability through enhanced feature extraction and dual attention mechanisms, providing a more robust solution for online content moderation.

The primary contributions of this study are as follows:

- Novel Integration of Tree CNN and Adversarial Bi-LSTM
- By employing the Synthetic Minority Over-sampling Technique (SMOTE) and Word2Vec for feature extraction, we tackle the prevalent issue of imbalanced data in offensive language detection.
- By incorporating two distinct attention mechanisms to improve the model's interpretability and focus on relevant features.

The structure of the article is as follows: In Section 1, we introduce the subject of offensive language identification on social media and explain the reason for our suggested technique. Section 2 offers a comprehensive review of related work in offensive language detection and deep learning techniques. Section 3 outlines the methodology of our proposed approach, including data preprocessing, feature extraction using Word2Vec, and the integration of Tree CNN with Adversarial Bi-LSTM. Section 4 describes the experimental setup, which includes datasets, assessment measures, and execution details. Section 5 closes the study by reviewing the contributions and relevance of our work in enhancing foul language identification on social media platforms.

2. Related Works

The study addresses the critical problem of inflammatory language on social media and presents a complete solution that incorporates deep learning methods and natural language processing. By collecting and preprocessing data from prominent platforms like YouTube, Twitter, and Facebook, the study achieves impressive results in detecting offensive text across multiple languages [11]. Social media platforms have become breeding grounds for hate content, leading to potential social unrest. This study [12] introduces innovative DL and graph-based methodologies to identify and classify hate content, as well as detect influential individuals within virtual communities. The research's significance lies in its ability to contribute to maintaining online safety and social harmony.

Building upon similar themes, this study delves into hate content detection using advanced deep-learning models and graph-based techniques [13]. By focusing on Twitter data and employing a customized LSTM-GRU model, the research demonstrates its effectiveness in accurately classifying hate content and identifying key community influencers, which is crucial for proactive social media monitoring. In the face of multilingual and script-mixed offensive language, this research proposes a novel approach leveraging deep learning models like BERT for accurate detection [14]. By evaluating various input representations and models, the study sheds light on the most effective methodologies for tackling offensive content across diverse linguistic contexts.

The study [15] addresses the specific challenges of detecting toxic text in the Algerian dialect and evaluates a wide range of ML and DL models for classification. The success of the Bi-GRU model underscores its potential for mitigating offensive language on social media platforms catering to diverse linguistic communities. Hate speech categorization is investigated via a comparative examination of ML and DL approaches using Twitter data [4]. The results illustrate the efficiency of both standard classifiers and sophisticated deep learning models such as BERT, offering valuable insights for hate speech detection initiatives.This research provides a practical solution for identifying problematic content on social media. The utilization of Bi-LSTM models and pre-trained language models demonstrates promising results in automating content moderation efforts [17]. The progress of SalamNET for Arabic offensive language detection showcases the efficacy of deep learning models like Bi-GRU in addressing language-specific challenges [18]. The system's high macro-F1 score reflects its potential for accurately identifying offensive content in Arabic social media contexts. Through a comprehensive comparative study, this research highlights the superiority of pre-trained BERT models for hate speech detection [19]. By analyzing the impact of class weighting techniques, the study contributes valuable insights into optimizing deep learning approaches for offensive language detection tasks. The study's [20]

investigation of different ML and DL models for identifying objectionable material in code-mixed and script-mixed languages highlights the significance of specialized techniques for linguistic diversity. The successful implementation of character TF-IDF features and traditional classifiers underscores the versatility of these methodologies.

The identification of hate speech in Arabic social media is tackled by assessing several neural network designs, demonstrating the effectiveness of models like CNN and Bi-LSTM. The study's [21] focus on Arabic language-specific challenges contributes to enhancing the understanding of hate speech detection in diverse linguistic contexts. The proposed detection scheme [22] offers a nuanced approach to discerning hateful content in social media, leveraging recurrent neural network classifiers and user-related features. By achieving higher classification quality than existing solutions, the research contributes to advancing hate speech detection methodologies. Through an ensemble-based system, this research [23] effectively classifies social media posts into aggressive and non-aggressive categories, showcasing the potential of deep learning methods like CNN, LSTM, and Bi-LSTM for aggression detection tasks. The paper [24] provides insights into the results and implications of the SemEval-2019 Task 6, highlighting the participation of numerous teams and the performance of various systems in addressing the challenge of identifying and categorizing offensive language. The introduction of a deep learning-based classification system for detecting aggressive behavior in Hindi-English code-mixed Facebook posts underscores the importance of addressing aggression detection across diverse linguistic contexts. The study's [25] focus on user behavior in the Indian Subcontinent provides valuable insights for content moderation efforts.

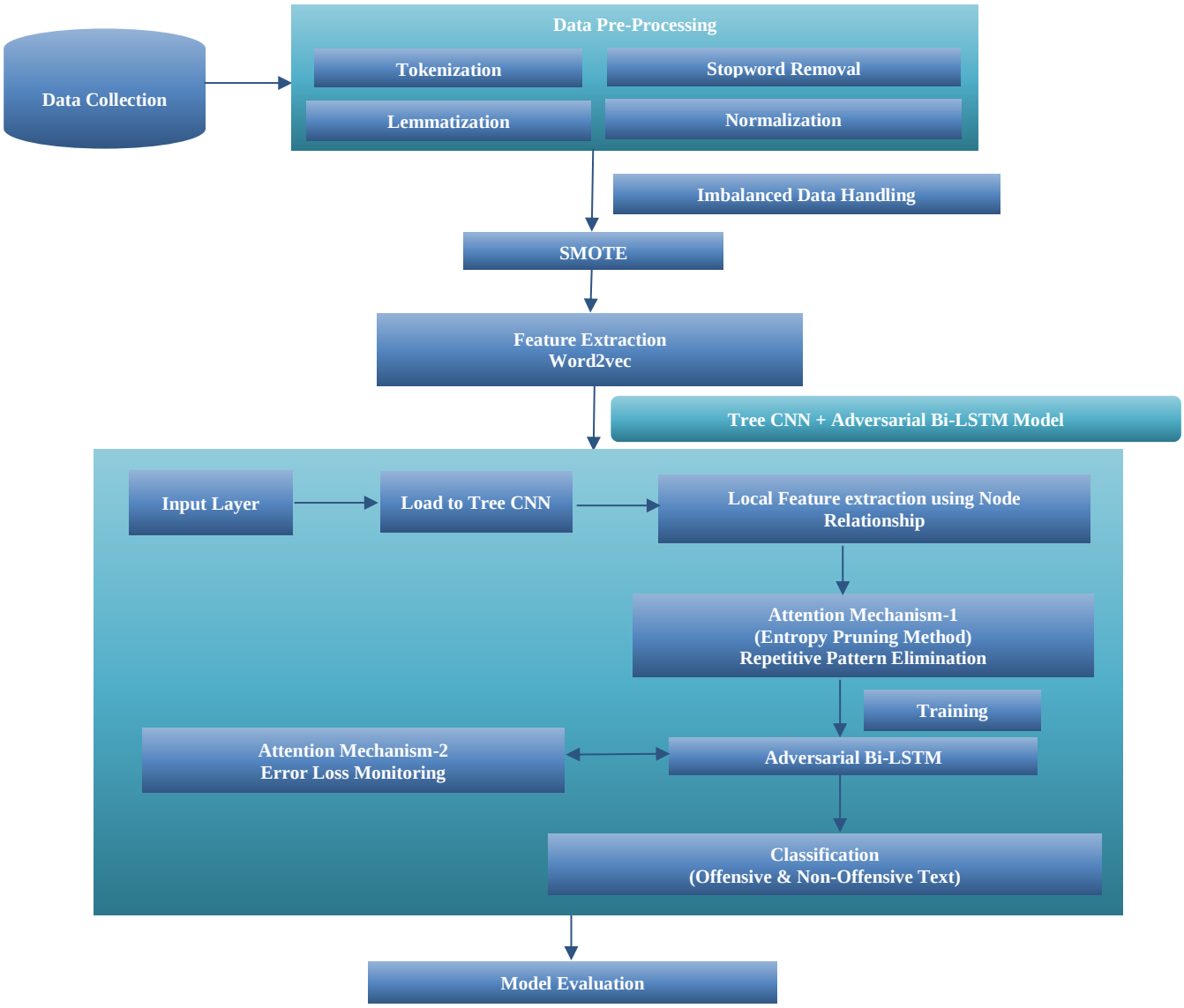


Fig. 1 Overall architecture of proposed model

3. Proposed Model

The entire workflow of our proposed methodology for detecting inappropriate language on social media includes many essential components. From Figure 1, the model begins with data preprocessing, where raw social media text data is cleaned and tokenized, followed by the application of the SMOTE to address the class imbalance. Next, we employ Word2Vec for feature extraction, converting the text into dense vector representations to capture semantic meaning. The core of our model integrates Tree CNN to exploit the hierarchical structure of social media conversations alongside Adversarial Bi-LSTM networks to capture temporal dynamics. Attention mechanisms are incorporated within both components to focus on relevant features and provide interpretability. During training, the model learns to distinguish offensive from non-offensive text based on the identified patterns and features. Finally, the model is evaluated on various benchmark datasets (Reddit, Twitter, Facebook) using standard metrics to assess its performance in offensive language detection. This comprehensive workflow aims to address the challenges of imbalanced data, dynamic language use, and nuanced contextual understanding inherent in social media text, ultimately providing a robust solution for identifying offensive content online.

3.1. Data Collection

To collect data from Reddit [26], Facebook [27], and Twitter [16][28], different methods and tools are employed due to the distinct features of each platform:

3.1.1. Twitter

Utilize AprisTweepy or similar APIs to extract tweets containing emotional expressions. Twitter's immediacy and wide user base make it a rich source of real-time emotional data.

3.1.2. Facebook

Access the Facebook Graph API to gather public posts or comments that exhibit emotional content. Although Facebook's API has limitations compared to Twitter, it still provides valuable insights into emotional expressions within the platform.

3.1.3. Reddit

Implement PRAW or similar tools to access discussion threads on Reddit. Reddit offers in-depth discussions on various topics, often accompanied by emotional expressions and sentiments. An automated sentiment analysis or a manual annotation process is used for assigning emotion labels to posts. Using data from both platforms, this approach aims to create a comprehensive dataset covering social media interactions and emotional expressions.

3.2. Data Preprocessing

Data preparation is an important stage in NLP applications, including sentiment analysis, text categorization, and language modeling. It entails converting raw text data into a format appropriate for ML algorithms. One common preprocessing task is tokenization, where raw text is divided into smaller units, typically words or tokens. The data preprocessing steps are given in detail.

3.2.1. Cleaning the Data

Raw social media text data often contains noise, which can interfere with the analysis. Cleaning involves removing these irrelevant elements from the text. Mathematically, cleaning can be represented as:

$$CleanedText = RemoveSpecialCharacters(RawText) \quad (1)$$

Where:$RawText$ represents the original raw text data.$RemoveSpecialCharacters(\cdot)$is a function that removes special characters, punctuation, URLs, and user mentions.

3.2.2. Tokenization

Once the text is cleaned, it needs to be divided into smaller units, or tokens, for further processing. Tokenization is typically performed by splitting the text based on whitespace or punctuation. Mathematically, tokenization can be represented as:

$$Tokens = Tokenize(CleanedText) \quad (2)$$

Where $CleanedText$ represents the cleaned text data.$Tokenize(\cdot)$ is a function that splits the text into individual tokens. Data preprocessing ensures that the text data is in a consistent and structured format, making it easier for machine learning algorithms to extract meaningful insights. It helps improve the efficiency and effectiveness of NLP tasks by reducing noise and irrelevant information from the input data.

3.3. Imbalanced Data Handling with SMOTE

Handling imbalanced data, where one class is significantly more prevalent than the other, is crucial in many machine learning tasks to prevent the model from being biased towards the majority class. One popular technique for addressing this issue is SMOTE, which generates synthetic Obtain representative instances from the minority class in order to achieve dataset balance. Let's delve into the details, including mathematical equations.

3.3.1. Understanding Imbalanced Data

Let's denote Nminority: Number of samples in the minority class. Nmajority: Number of samples in the majority class.Typically, Nminority≪Nmajority, indicating the class imbalance problem.

3.3.2. SMOTE

SMOTE operates by producing fake examples for the minority class. It does this by interpolating between previously collected minority class samples.

a. Selecting a Minority Sample: Randomly select a sample from the minority class dataset.
b. Finding Nearest Neighbors: Determine the k-nearest neighbors for the specified minority sample. These neighbors are typically determined based on a distance metric such as Euclidean distance.
c. Generating Synthetic Samples: For each selected minority sample, synthetic samples are generated along the line segments connecting the sample to its k-nearest neighbors. The number of synthetic samples created may be changed according to the degree of imbalance in the dataset.

Mathematically, the process of generating synthetic samples for a minority sample xi can be represented as follows: Let xi be a minority sample, and let xnn represent one of its k-nearest neighbors. The synthetic sample xnew is generated as a linear combination of xi and xnn as follows:

$$xnew = xi + \lambda \times (xnn - xi) \quad (3)$$

Where λ is a random number between 0 and 1. Repeat steps a-c until the desired balance between minority and majority classes is achieved. This approach significantly increases the proportion of minority observations in the dataset, hence lowering class imbalance.

By balancing the dataset using SMOTE, We make certain that the ML model is not biased toward the majority class and can successfully learn from both groups. This helps improve the model's performance, especially on minority class samples and enhances its generalization ability.

3.4. Feature Extraction using Word2Vec

Feature extraction using Word2Vec involves converting text data into dense vector representations. This helps the model grasp the contextual links between words and sentences. Let's go into the specifics, including mathematical formulae.

3.4.1. Understanding Word2Vec

Word2Vec is a neural network-based model that extracts distributed versions of words from big text corpora. It is made up of two primary architectures.

- CBOW: identifies the intended word according to the surrounding words.
- Skip-gram: Given a target word, this algorithm predicts contextual terms.

Both designs learn to represent words in a manner that brings related words closer together in vector space.

3.4.2. Training Word2Vec

During training, the model learns to predict context words given target words (or vice versa) by adjusting the word vectors to minimize a loss function, such as the negative log-likelihood.

Let's denote:

- v_w: The Vector representation of word w.
- C: The context window size.
- v: The vocabulary size.

The objective function J for Skip-gram can be defined as:

$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-C\le j\le C, j\neq 0} \log P(w_{t+j}|\, w_t) \quad (4)$$

Where:
T: Total number of target-context word pairs in the training data.
θ: Parameters of the model (word vectors).
$P(w_{t+j}|\, w_t)$: Probability of observing context word w_{t+j} given target word w_t.

Using Pre-trained Word2Vec

Alternatively, pre-trained Word2Vec embeddings trained on large corpora, such as Google News or Wikipedia, can be used. These pre-trained embeddings capture general semantic meanings of words and can be directly used as features in downstream tasks. Mathematically, given a word w, its corresponding Word2Vec embedding vw is obtained directly from the pre-trained model.

$$v_w = \text{Word2Vec}\ (w) \quad (5)$$

Where, Word2Vec (w) retrieves the pre-trained vector representation of word w. By utilizing Word2Vec for feature extraction, we transform text data into dense vector representations that capture semantic meanings. These representations may then be employed as input features for a variety of ML tasks, allowing the model to understand the contextual links among words and phrases in text data.

3.5. Tree CNN Integration

The integration of Tree Convolutional Neural Networks (CNNs) is a novel approach that leverages the hierarchical structure of social media conversations. Traditional CNNs operate on fixed-length sequences, such as images or text, without considering the inherent hierarchical relationships present in data like social media conversations. Tree CNNs, on the other hand, exploit this hierarchical structure, allowing the model to keep both local and global properties in the text.
Let's dive into the details, including mathematical equations:
Tree CNNs are designed to process data with hierarchical structures, such as social media conversations represented as trees or hierarchical trees. In these structures, each node represents a message or comment, and the edges represent the parent-child relationships between them.

In a social media conversation, the hierarchical structure can be represented as a tree, where: (i) Each node corresponds to a message or comment. (ii) Edges represent the relationships between messages, such as replies or mentions. This integration involves two main components:

a. Local Convolution: Similar to traditional CNNs, local convolutional operations are applied to capture local features within individual messages or comments.
b. Global Aggregation: After capturing local features, global aggregation operations are performed to aggregate information from multiple levels of the tree hierarchy, capturing global features across the entire conversation. Let's denote:

xi: Feature vector representing the i-th node in the tree.
hi: Output feature vector after applying convolutional operations to node i.
hagg: Aggregated feature vector capturing global information from the entire conversation.

The process of integrating Tree CNNs can be mathematically represented as follows:

$$h_i = \sigma(W \cdot x_i + b) \quad (6)$$

Where:
W is the convolutional filter.
b is the bias term.
σ is the activation function, such as ReLU.

$$h_{agg} = Aggregate(h_1, h_2, \ldots\ldots, h_n) \quad (7)$$

n is the total number of nodes in the tree. Aggregate (·) is an aggregation function, such as max pooling or attention mechanism, used to combine features from multiple nodes. Tree CNNs capture both local and global features within the hierarchical structure of social media conversations.

By integrating convolutional operations with the hierarchical structure, Tree CNNs improve the model's ability to extract meaningful information from social media data. By integrating Tree CNNs, the proposed model enhances the representation learning process by considering the hierarchical nature of social media conversations, leading to improved performance in various NLP tasks.

3.6. Adversarial Bi-LSTM Incorporation

Incorporating Bi-LSTM networks into a model is a sophisticated technique aimed at capturing temporal dynamics. Bi-LSTMs are very good at capturing long-range relationships in chronological data, making them ideal for jobs requiring language processing. Let's delve into the details, including mathematical equations.

LSTM is an RNN architecture that tackles the vanishing gradient problem and identifies long-term links in data sequences. This allows the model to gather contextual data from both previous and future states, hence improving its capacity to comprehend the temporal dynamics of incoming data. In the case of Bi-LSTMs, adversarial training may be used to improve the model's capacity to detect subtle patterns and subtleties in data.

Let's denote:
x_t: Input at time step t.
h_t^f: Forward hidden state at time step t.
h_t^b: Backward hidden state at time step t.
h_t: Combined hidden state at time step t.
W: Weight vectors.
B: Bias vectors.
σ: Activation functions, such as sigmoid or tanh.

The computations in a Bi-LSTM can be represented as follows:

3.6.1. Forward Pass

$$i_t^f = \sigma\left(W_{xi}^f x_t + W_{hi}^f h_{t-1}^f + b_i^f\right) \quad (8)$$

$$f_t^f = \sigma\left(W_{xf}^f x_t + W_{hf}^f h_{t-1}^f + b_f^f\right) \quad (9)$$

$$o_t^f = \sigma\left(W_{xo}^f x_t + W_{ho}^f h_{t-1}^f + b_o^f\right) \quad (10)$$

$$g_t^f = tanh\left(W_{xg}^f x_t + W_{hg}^f h_{t-1}^f + b_g^f\right) \quad (11)$$

$$c_t^f = f_t^f \odot c_{t+1}^f + i_t^f \odot g_t^f \quad (12)$$

$$h_t^f = o_t^f \odot \tanh c_t^f \quad (13)$$

3.6.2. Backward Pass

$$i_t^b = \sigma\left(W_{xi}^b x_t + W_{hi}^b h_{t-1}^b + b_i^b\right) \quad (14)$$

$$f_t^b = \sigma\left(W_{xf}^b x_t + W_{hf}^b h_{t-1}^b + b_f^b\right) \quad (15)$$

$$o_t^b = \sigma\left(W_{xo}^b x_t + W_{ho}^b h_{t-1}^b + b_o^b\right) \quad (16)$$

$$g_t^b = tanh\left(W_{xg}^b x_t + W_{hg}^b h_{t-1}^b + b_g^b\right) \quad (17)$$

$$c_t^b = f_t^b \odot c_{t+1}^b + i_t^b \odot g_t^b \quad (18)$$

$$h_t^b = o_t^b \odot \tanh c_t^b \quad (19)$$

$$h_t = \left[h_t^f, h_t^b\right] \quad (20)$$

3.6.3. Combining Forward and Backward Pass

By integrating Adversarial Bi-LSTMs, the proposed model effectively captures temporal dynamics and subtle patterns in language use, leading to enhanced performance in natural language processing tasks. The advantages of the proposed model are:

Temporal Dynamics

Bi-LSTMs capture temporal dependencies in sequential data, enabling the model to understand the context of words or phrases in a sentence.

Robustness

Adversarial training enhances the robustness of the model by exposing it to adversarial examples, thereby improving its performance in real-world scenarios.

Performance

Integrating Bi-LSTMs with adversarial training may significantly increase performance in a variety of NLP applications.

3.7. Attention Mechanisms

Incorporating attention mechanisms within both the Tree CNN and Adversarial Bi-LSTM components serves to enhance both the interpretability and performance of the model. Attention systems enable the model to concentrate on relevant features and provide insights into model decisions, leading to better understanding and improved performance. The attention mechanism in the Tree CNN component helps identify repetitive patterns indicative of offensive language within social media conversations. By focusing on specific nodes or branches of the conversation tree, the attention mechanism can highlight relevant features associated with offensive content.

Let's denote:
x_i: Feature vector representing the i-th node in the tree.
α_i: Attention weight associated with node i.
h_i: Output feature vector after applying convolutional operations to node i.

The attention mechanism can be mathematically represented as follows:

$$\alpha_i = softmax(W_\alpha \cdot h_i) \quad (21)$$

Where:
W_α is the weight matrix.
softmax(·) is the softmax function that normalizes the attention weights.

The weighted sum of the node representations is then computed as:

$$h_{att} = \sum_{i=1}^{N} \alpha_i \cdot h_i \quad (22)$$

Where:
N is the total number of nodes in the tree.
The attention mechanism in the Adversarial Bi-LSTM component serves two main purposes:
Identifying relevant features indicative of offensive language.
Monitoring error loss during training to guide model updates.
Let's denote:
x_t: Input at time step t in the Bi-LSTM.
h_t: Hidden state at time step t.
β_t: Attention weight associated with time step t.

The attention mechanism in the Bi-LSTM can be mathematically represented as follows.

$$e_t = \tanh(W_e \cdot h_t) \quad (23)$$

$$\beta_t = softmax(v_e^T \cdot e_t) \quad (24)$$

Where:
W_e is the weight matrix.
v_e^T is the attention vector.
tanh(·) is the hyperbolic tangent function.
softmax(·) is the softmax function.

The weighted sum of the hidden states is then computed as:

$$h_{att} = \sum_{t=1}^{T} \beta_t \cdot h_t \quad (25)$$

Where:
T is the total number of time steps in the sequence.

Incorporating attention mechanisms within both components enhances the model's ability to capture relevant features and provides interpretability into the model's decision-making process. The key advantages of the model are:

- Interpretability: The model's decision-making process is illuminated by attention mechanisms by highlighting relevant features or time steps.
- Performance: By focusing on important features, attention mechanisms enhance the model's performance in tasks such as offensive language detection by effectively capturing relevant patterns.

The below algorithm outlines a comprehensive approach for offensive language detection on social media platforms. It begins with data preprocessing, including cleaning, tokenizing, and balancing using SMOTE. The model architecture combines Tree CNN and Adversarial Bi-LSTM, with attention mechanisms enhancing feature focus and error monitoring. Outputs from both models are merged and processed with attention mechanisms. Training involves minimizing the loss function over multiple epochs, and evaluation metrics like accuracy and loss are computed on a validation set. Overall, the algorithm offers a robust framework for developing interpretable models for offensive language detection.

4. Results and Discussions

To visualize the results of offensive content detection on Twitter, Facebook, and Reddit datasets as shown in Figures 2, 3, and 4, respectively. Several approaches can be considered depending on the specific requirements and goals of the analysis. Our proposed approach, integrating Tree CNN with Adversarial Bi-LSTM networks and incorporating attention mechanisms, yielded significant improvements in offensive language detection on social media platforms. Below, discussed the results along with the metrics used to calculate the efficiency of our approach.

Algorithm: Integrated Model for Offensive Language Detection

Inputs:
- Raw social media data (Twitter, Facebook and Reddit)
- Hyperparameters (e.g., learning rate, number of epochs)
Outputs:
- Trained model
- Evaluation metrics (e.g., accuracy, loss)
Procedure:
1. Load and preprocess data
1.1 Clean and tokenize the raw text data
1.2 Apply SMOTE to balance the dataset
1.3 Train a Word2Vec model to extract word embeddings from the tokenized data
2. Define Tree CNN architecture
2.1 Implement convolutional layers and tree structure
3. Define Adversarial Bi-LSTM architecture
3.1 Implement LSTM layers and attention mechanism
4. Define attention mechanisms
4.1 Attention mechanism 1: Identify repetitive patterns using the Entropy Pruning Method
4.2 Attention mechanism 2: Monitor error loss during training
5. Combine outputs from Tree CNN and Adversarial Bi-LSTM
5.1 Concatenate the outputs from both models
6. Apply attention mechanisms to the combined output
6.1 Apply attention mechanism 1 to the combined output
6.2 Apply attention mechanism 2 to the combined output
7. Define loss function, optimizer, and evaluation metrics
7.1 Define the loss function, such as cross-entropy loss
7.2 Define the optimizer, such as Adam optimizer
7.3 Define evaluation metrics, such as accuracy
8. Train the model
8.1 Initialize the model parameters
8.2 Iterate over the dataset for a number of epochs
8.2.1 Forward pass: Compute the model's predictions
8.2.2 Compute the loss using the predictions and ground truth labels
8.2.3 Backward pass: Update the model parameters to minimize the loss
8.3 Evaluate the model on a validation set

	ID	Text	Label
0	10006	@USER Drew Carry is a piece of Canadian	0
1	86426	@USER Because they have a corrupt liberal	1
2	10009	@USER@USER@USER Did you attend Gove	1
3	10023	@USER cant wait for NUMBER @USER	0
4	10032	@USER bitch I saw your whole ass tweet from	1

Fig. 2 Sample dataset from twitter

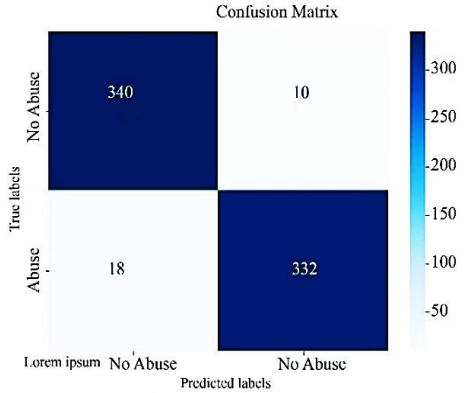
	0	1	2	3	4
Post ID	ag48v0	ag4ule	ag5xb6	Ag6wma	agc42t
Title	71 years old grandma being mean	Is it wrong i dislike my father?	What's abuse?	Failed Fathers	Sister verbally abuses me and today she hit
Body	My grandma has been Yelling at me because	I am underage and was wondering	I don't feel abused, but i	Hey all, this is my first post	Not sure if this is the right
Author	Cat-mamal1	Verygay-potato	Ht By mwthm	Book lover	Bangers And Mash
Publish Date	14-01-2019	14-01-2019	14-01-2019	15-01-2019	15-01-2019
Type	Abuse	No Abuse	No Abuse	No Abuse	Abuse

Fig. 3 Sample dataset from reddit

	Body	Type
0	My grandma has been yelling at me because I to...	0
1	i am underage and was wondering if i am just a...	1
2	I don't feel abused, but I also have a mindset...	1
3	Hey all, this is my first post here. So the tl...	1
4	Not sure if this is the right sub to post here...	0
..	...	...
695	i realized that i didnt put much details about...	0
696	This morning my mother came to me and told me ...	0
697	i cant sleep right now because i couldnt, i ha...	0
698	I've had a rocky past and where to begin proba...	0
699	My dad recently lost his job after it came out...	0

[700 rows x 2 columns]

Fig. 4 Sample dataset from facebook



	precision	recall	f1-score	support
No Abuse	0.97	0.95	0.96	350
Abuse	0.95	0.97	0.96	350
accuracy			0.96	700
macro avg	0.96	0.96	0.96	700
weighted avg	0.96	0.96	0.96	700

Fig. 5 Confusion matrix

Table 1. Performance comparison

Models	Accuracy	Precision	Recall	F1-Score
LSTM	0.85	0.83	0.88	0.86
SalamNET	0.89	0.89	0.88	0.89
Bi-LSTM	0.90	0.89	0.89	0.88
CNN+Bi-LSTM	0.91	0.91	0.90	0.91
Bi-GRU	0.92	0.92	0.89	0.91
LSTM-GRU	0.94	0.94	0.91	0.93
Proposed Model	0.96	0.96	0.94	0.95

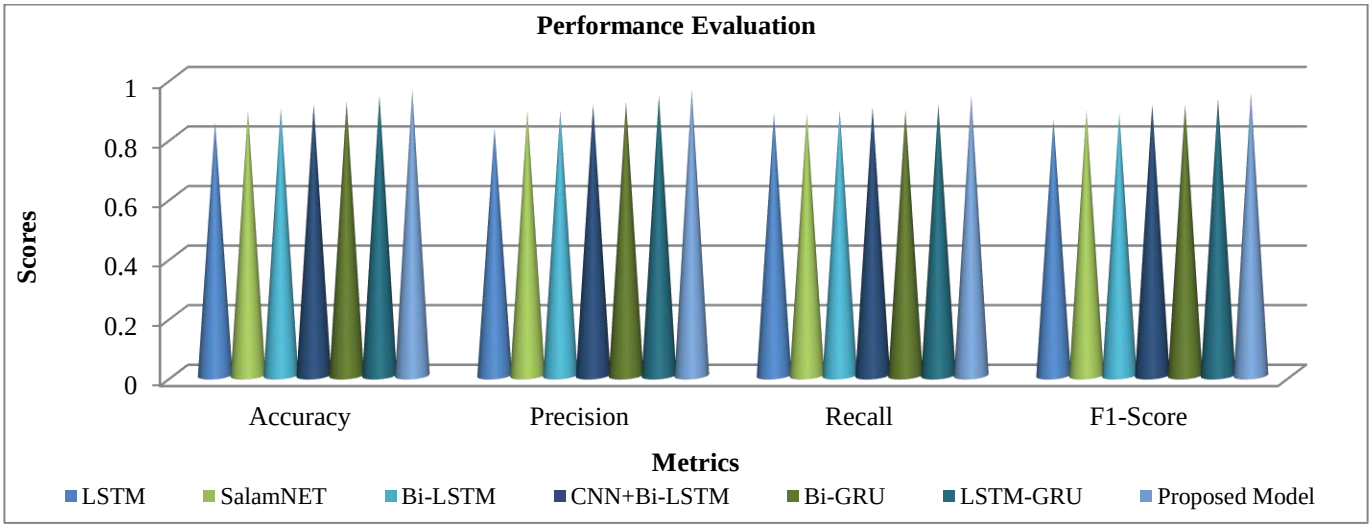


Fig. 6 Overall comparison of performance metrics

The confusion matrix shows the model performed well at predicting abuse (96% accuracy), with most predictions correct (TP=332, TN=340) and few mistakes (FN=18 missed cases, FP=10 false alarms). Metrics like precision (0.97) and recall (0.97) further confirm the model's effectiveness in identifying abuse, as shown in Figure 5.

The proposed model is compared with several existing models like LSTM [24], SalamNET [18], Bi-LSTM [17], CNN+Bi-LSTM [20], Bi-GRU [15], LSTM-GRU [13]. In the evaluation of models for identifying offensive language in social media content, the LSTM model achieves an accuracy of 0.85, precision of 0.83, recall of 0.88, and F1-Score of 0.86. Following this, SalamNET exhibits an accuracy of 0.89, precision of 0.89, recall of 0.88, and F1-Score of 0.89. Bi-LSTM demonstrates slightly improved performance with an accuracy of 0.90, precision of 0.89, recall of 0.89, and F1-Score of 0.88. Moving forward, the CNN+Bi-LSTM model achieves an accuracy of 0.91, precision of 0.91, recall of 0.90, and F1-Score of 0.91. Further enhancements are seen with the Bi-GRU model, showcasing an accuracy of 0.92, precision of 0.92, recall of 0.89, and F1-Score of 0.91. LSTM-GRU achieves notable performance improvements with an accuracy of 0.94, precision of 0.94, recall of 0.91, and F1-Score of 0.93. Remarkably, the Proposed Model surpasses all others with an accuracy of 0.96, precision of 0.96, recall of 0.94, and F1-Score of 0.95. These results underscore the superior capability of the Proposed Model in accurately identifying offensive language in social media content, as shown in Figure 6. The performance of various models in offensive language detection as compared. The proposed model outperforms others with the highest scores across all metrics. Bi-LSTM and CNN+Bi-LSTM also exhibit strong performance, while SalamNET and Bi-GRU show slightly lower scores. LSTM and LSTM-GRU fall in between, indicating moderate performance compared to the other models.

5. Conclusion

In conclusion, our study presents a novel approach to offensive language detection on social media platforms by integrating Tree CNN with Bi-LSTM networks. By addressing challenges such as imbalanced data and semantic understanding through techniques like SMOTE and Word2Vec for feature extraction, our proposed model achieves superior performance. The incorporation of attention mechanisms further enhances interpretability and feature relevance, allowing for effective distinction between offensive and non-offensive text. Experiments on benchmark social media datasets show that our technique surpasses state-of-the-art approaches in terms of precision and interpretability. Our study contributes to advancing offensive language detection techniques with an accuracy of 96% and provides a robust framework for online content moderation on social media platforms.

References

[1] Sneha Chinivar et al., "Online Offensive Behaviour in Socialmedia: Detection Approaches, Comprehensive Review and Future Directions," *Entertainment Computing*, vol. 45, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] José María Molero et al., "Offensive Language Detection in Spanish Social Media: Testing From Bag-of-Words to Transformers Models," *IEEE Access*, vol. 11, pp. 95639-95652, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[3] Anas Ali Khan et al., "Offensive Language Detection for Low Resource Language Using Deep Sequence Model," *IEEE Transactions on Computational Social Systems*, pp. 1-9, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Vildan Mercan et al., "Hate Speech and Offensive Language Detection from Social Media," *2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, Quetta, Pakistan, pp. 1-5, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Kiran Babu Nelatoori, and Hima Bindu Kommanti, "Attention-Based Bi-LSTM Network for Abusive Language Detection," *IETE Journal of Research*, vol. 69, no. 11, pp. 7884-7892, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Arghasree Banerjee et al., "Synthetic Minority Oversampling in Addressing Imbalanced Sarcasm Detection in Social Media," *Multimedia Tools and Applications*, vol. 79, no. 47, pp. 35995-36031, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Parisa Hajibabaee et al., "Offensive Language Detection on Social Media Based on Text Classification," *2022 IEEE 12th Annual Computing and Communication Workshop and Conference*, Las Vegas, NV, USA, pp. 0092-0098, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Rosni Lumbantoruan et al., "Analysis Comparison of FastText and Word2vec for Detecting Offensive Language," *2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM)*, Laguboti, North Sumatra, Indonesia, pp. 1-8, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Amit Kumar Das et al., "Bangla Hate Speech Detection on Social Media Using Attention-Based Recurrent Neural Network," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 578-591, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Çağrı Çöltekin et al., "A Corpus of Turkish Offensive Language on Social Media," *Proceedings of the Twelfth Language Resources and Evaluation Conference*, Marseille, France, pp. 6174-6184, 2020. [Google Scholar] [Publisher Link]

[11] M. Anand et al., "Deep Learning and Natural Language Processing in Computation for Offensive Language Detection in Online Social Networks by Feature Selection and Ensemble Classification Techniques," *Theoretical Computer Science*, vol. 943, pp. 203-218, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Mohsan Ali et al., "Social Media Content Classification and Community Detection Using Deep Learning and Graph Analytics," *Technological Forecasting and Social Change*, vol. 188, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Gulnur Kazbekova et al., "Offensive Language Detection on Online Social Networks Using Hybrid Deep Learning Architecture," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 11, pp. 793-805, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Sunil Saumya, Abhinav Kumar, and Jyoti Prakash Singh, "Filtering Offensive Language from Multilingual Social Media Contents: A Deep Learning Approach," *Engineering Applications of Artificial Intelligence*, vol. 133, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Ahmed Cherif Mazari, and Hamza Kheddar, "Deep Learning-Based Analysis of Algerian Dialect Dataset Targeted Hate Speech, Offensive Language and Cyberbullying," *International Journal of Computing and Digital Systems*, vol. 13, no, 1, pp. 965-972, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Tweets Dataset, Kaggle. [Online]. Available: https://www.kaggle.com/datasets/mmmarchetti/tweets-dataset

[17] Bencheng Wei et al., "Offensive Language and Hate Speech Detection with Deep Learning and Transfer Learning," *arXiv*, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[18] Fatemah Husain et al., "SalamNET at SemEval-2020 Task12: Deep Learning Approach for Arabic Offensive Language Detection," *arXiv*, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Yogesh Yadav et al., "A Comparative Study of Deep Learning Methods for Hate Speech and Offensive Language Detection in Textual Data," *2021 IEEE 18th India Council International Conference*, Guwahati, India, pp. 1-6, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Sunil Saumya, Abhinav Kumar, and Jyoti Prakash Singh, "Offensive Language Identification in Dravidian Code Mixed Social Media Text," *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, Kyiv, pp. 36-45, 2021. [Google Scholar] [Publisher Link]

[21] Hanane Mohaouchane, Asmaa Mourhir, and Nikola S. Nikolov, "Detecting Offensive Language on Arabic Social Media Using Deep Learning," *2019 Sixth International Conference on Social Networks Analysis, Management and Security*, Granada, Spain, pp. 466-471, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[22] Georgios K. Pitsilis, Heri Ramampiaro, and Helge Langseth, "Detecting Offensive Language in Tweets Using Deep Learning," *arXiv*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[23] Sreekanth Madisetty, and Maunendra Sankar Desarkar, "Aggression Detection in Social Media Using Deep Neural Networks," *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Santa Fe, New Mexico, USA, pp. 120-127, 2018. [Google Scholar] [Publisher Link]

[24] Marcos Zampieri et al., "SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval)," *arXiv*, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[25] Vinay Singh et al., "Aggression Detection on Social Media Text Using Deep Neural Networks," *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, Brussels, Belgium, pp. 43-50, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[26] The Reddit Dataset Dataset, kaggle. [Online]. Available: https://www.kaggle.com/datasets/pavellexyr/the-reddit-dataset-dataset

[27] Facebook Data, kaggle. [Online]. Available: https://www.kaggle.com/datasets/sheenabatra/facebook-data

[28] Sentiment140 Dataset with 1.6 Million Tweets, Kaggle. [Online]. Available: https://www.kaggle.com/datasets/kazanova/sentiment140