

Enhanced Convolutional Neural Network Model Using Bidirectional Encoder Representations from Transformers and Long Short-Term Memory for Hate Speech Detection

Chukwuemeka Okechukwu.^{1*}, Chinweoke Nwodo², P.C. Anyaora³, Gungbias Hosea⁴, S.O. Subairu⁵,
J.A. Ojeniyi⁶

^{1,2,3,4,5,6}Federal University of Technology Minna

*Anthony.pg208286@st.futminna.edu.ng, +2348068573605

ABSTRACT

Hate speech has become a prevalent and concerning issue in this digital age as a result of the adoption of social media as a means of communication, social interaction, and driving social discussion. This trend has opened a novel front in cybercrime, cyberbullying, privacy violations, and exposure of the people component of information security to violence, subsequently violating the governance and internal security of nations. The detection and mitigation of hate speech on online platforms have gained significant attention due to the growing influence of social media and the internet as communication channels and the adoption of the same as a means of propagating hate and hate crimes. This thesis offers a model for the more accurate detection of hate speech on an online platform using Bidirectional Encoder Representations from Transformers (BERT) to enhance the feature selection layer of Convolutional Neural Networks (CNN) and a hybrid of CNN and Long-Short-Term Memory (LSTM) to detect hate speech. The proposed BERT-CNN-LSTM model achieved an accuracy of 96.1%, which is an increase of over 0.45% from 95.63% in the benchmark paper, thereby contributing a novel and more accurate tool that, when adapted into the security policies of organisations and governments, will help safeguard the rights and privacy of users, thereby protecting them from both physical and psychological harm resulting from hate propagation over the internet. **Keywords:** Hate Speech, Ensemble, Machine Learning, Detection, Social Media

1. INTRODUCTION

Internet penetration with subsequent adoption of the social media space for driving social conversations and discussions have led to the adoption of the same as a means of propagating hate, offensive content, racial and ethnic discrimination, and violence thereby exposing the users of the cyber space to risks and privacy violations.

Recent years have seen a transformation in how individuals connect, share information, and express their ideas thanks to the growth of social media platforms and online communities. While these platforms offer numerous benefits, they have also

become breeding grounds for hate speech, which poses significant threats to individuals, groups, and societal harmony (Council of Europe, 2023). The phrase "hate speech" covers a broad range of statements that promote, provoke, justify, invite animosity, prejudice or violence, targeted at a person or a group of people for a variety of reasons (Council of Europe, 2023). It can have negative effects that are severe, such as causing emotional suffering, inciting aggression, and maintaining discrimination.

Any speech that disparages, demean, or incite hatred toward a class of humans or a group of individuals as a result of their racial origin, their religious beliefs, skin colour, gender identity, sexual orientation,

disability, ethnicity, or nation of origin is known as hate speech (American Library Association, 2017). There are various legal definitions of what hate speeches are in the United States and the First Amendment protects these speeches (American Library Association, 2017). It is an outright assault on individuals based on what are referred to as their "protected traits" (Allan, 2017). It is widely assumed that it consists of utterances of hatred or denigration of a person or group of persons because of a feature shared by them, such as religion, colour, sex, national origin, handicap, or race.

Hate speech highlights the "identity factors" of an individual or a group that are real or seen to be such, such as nationality, ethnicity, religion, descent, colour, race, and such factors (United Nations, 2023). The coherence of a democratic society, the defence of the rule of law and human rights, are all seriously endangered by hate speech, an extreme form of intolerance that promotes hate and hate-related crimes (Council of Europe, 2023).

1.1 Statement of the Research Problem

Hate speech committed through the cyber space dubbed Cyberhate, is often motivated by ideologies, is the use of violent, aggressive, or offensive language on the Internet and social media, targeting specific groups based on shared property and power imbalances (Castaño-Pulgarín *et al.*, 2021).

Cybersecurity is indispensable in safeguarding individuals from a myriad of threats that are inherent in the digital landscape. With the exponential growth of online activities and the increasing reliance on digital platforms for communication, commerce, and social interaction, the protection of personal data,

privacy, and people's safety has become paramount. Cybersecurity measures are essential for mitigating the risks posed by malicious actors who seek to exploit vulnerabilities in digital systems to perpetrate crimes such as identity theft, financial fraud, cyberbullying and hate speech. Furthermore, the proliferation of hate speech, misinformation, and online harassment underscores the urgent need for measures to protect individuals from physical and psychological harm and ensure a safe and inclusive online environment. Effective cybersecurity practices not only protect people's personal and sensitive information but also uphold fundamental principles of human dignity, freedom, and security in the digital age.

Researchers have contributed solutions to the problem of hate speech using various approaches ranging from using linguistic features, meta information and lexical resources by Schmidt and Wiegand, (2017), to machine learning models by Davidson *et al.* (2017) with precision of 0.91 and F-1 score of 0.90, deep neural models like CNN and Long-Short Term Memory (LSTM) and their derivatives, Zhang *et al.* (2018) with F-1 of 0.94 and a work by Hasan *et al.* (2022) in which hate speech among migrant women were analysed and detected using Transformer-CNN and MLP with 95.63% Accuracy and F-1 of 0.938. In our previous research work in Okechukwu *et al.*, (2023), the majority voting ensemble machine learning with TF-IDF model obtained an Accuracy of 95% and an F-1 Score of 0.95.

From the result of these researches, it can be seen that in the existing models, there is a need to improve on the performance (Accuracy and F-1 score) of the detection models thereby reducing their error rate as an undetected hate speech can go viral and cause real harm before they are reported by humans.

1.2 Aim and Objectives of the Study

1.2.1 Aim of the study

The aim of this research work is to enhance a CNN model using LSTM and BERT to detect hate speeches online which will be implemented into security policies thereby minimising security threats on users, employees and the communities at large.

1.2.2 Objectives of the study

The objectives to achieve the aim of this study are to:

- improve the feature selection layer of CNN by adding a fine-tuned BERT layer to CNN input layer.
- develop an enhanced CNN model for hate speech detection using LSTM
- evaluate the performance of the enhanced CNN model using F-1 score and accuracy performance metrics.

1.3 Scope of the Study

This study mainly focuses on developing a CNN based deep learning model for the detection of hate speeches in online forums, social media and blog sites by hybridizing the BERT and LSTM models with the base CNN model.

2. METHODOLOGY

The steps shown in figure 3.1 are followed from dataset collection, text preprocessing, splitting, tokenization, model training, and model performance result evaluation in order to detect hate speech. The dataset was first downloaded from a publicly accessible repository on kaggle.com, preprocessed to

remove links, special characters, and other noises, and divided into training and test sets. The pretrained BERT model is then adjusted to make it suitable for the hate speech task using the training data, which has been tokenized using the BERT tokenizer. A CNN and LSTM model receive additional input from the refined BERT model. To determine how well the model performed, the predictions and outcomes were compared to the original class. Figure 3.2 shows this same hate speech detection process in a flowchart form.

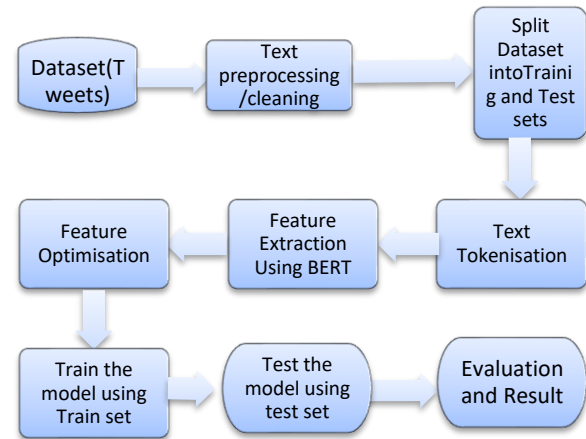


Figure 2.1: Proposed Hate Speech Detection Procedure.

Table 2.1: Dataset Description

SN	Feature Name	Data Type
0		Integer
1	Count	Integer
2	hate_speech	Integer
3	offensive_language	Integer
4	Neither	Integer
5	Class	String
6	Tweet	String

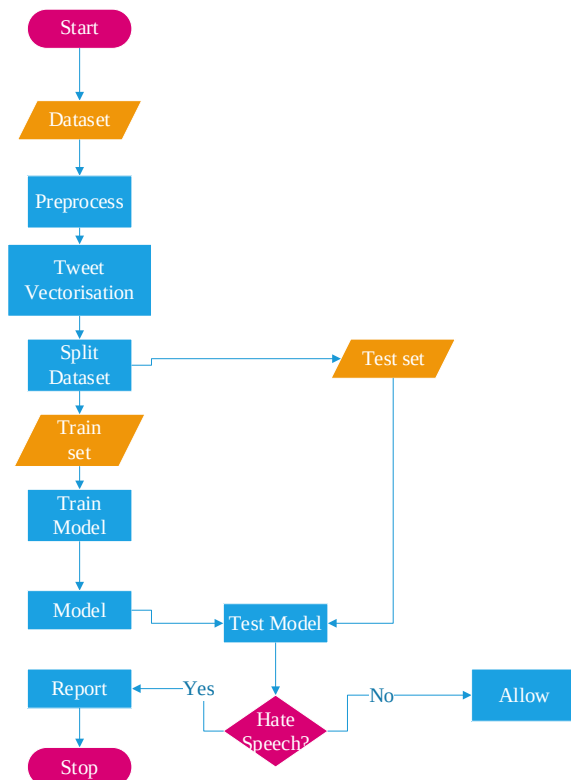


Figure 2.2: Flowchart of Hate speech detection.

2.1 Data Collection and Preprocessing

This study, like many others, made use of a dataset from Kaggle.com called (Hate-Speech-Data-Analysis | Kaggle, 2020) to conduct its research. It contains 24783 examples of hateful, offensive, and non-hateful speeches in English language. The dataset has the following attributes: count, hate speech, offensive language, neither, class, and tweet, as given in Table 2.1. Table 2.2 presents a sample of the dataset before preprocessing, and Table 2.3 presents a sample of the preprocessed dataset.

Table 2.2: Sample Dataset

	tweet	class
0	!!! RT @mayasolovely: As a woman you shouldn't...	0
1	!!!! RT @mleew17: boy dats cold...tyga dwn ba...	1
2	!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...	1
3	!!!!!!! RT @C_G_Anderson: @viva_based she lo...	1
4	!!!!!!! RT @ShenikaRoberts: The shit you...	1
5	!!!!!!!@T_Madison_x: The shit just...	1
6	!!!!!!"@_BrighterDays: I can not just sit up ...	1
7	!!!!“@selfiequeenbri: cause I'm tired of...	1
8	" & you might not get ya bitch back & ...	1
9	" @rhythmixx_:hobbies include: fighting Maria...	1

Table 2.3: Sample of Preprocessed Dataset

	tweet	class
0	rt mayasolovely woman complain cleaning house ...	0
1	rt mleew boy dats coldtyga dwn bad cuffin dat ...	1
2	rt urkindofbrand dawg rt sbaby life ever fuck ...	1
3	rt c g anderson viva based look like tranny	1
4	rt shenikaroberts shit hear might true might f...	1
5	madison x shit blows meclain faithful somebody...	1
6	brighterdays sit hate another bitch got much s...	1
7	selfiequeenbri cause tired big bitches coming ...	1
8	amp might get ya bitch back amp thats	1
9	rhythmixx hobbies include fighting mariam bitch	1

2.2 Data Preprocessing

First, multiple pre-processing processes are conducted to both the training and testing datasets before the tweets are fed to the input layer of the classification models. A number of preprocessing processes were performed on the data before supplying it as an input to the proposed models. The following steps were taken: We changed user tags to "user," got rid of urls, got rid of superfluous symbols that add a lot of noise to the dataset, and got rid of stopwords. This action was made to prevent discrimination against particular classes.

2.1.1 Feature transformation

The dataset as seen in Table 2.1 has three classes hate, offensive and None. In this research, hate and offensive languages were grouped as a class and None as another class and designated as 1 and 0 respectively. The purpose for this is to have a binary class that will improve the performance of our proposed model both in training time, accuracy and prediction time. The tweet and class attributes are the only attributes used in this research. The tweet will be tokenized using a BERT Tokenizer algorithm from Tensorflow package before feature extraction is carried out by the overlaying CNN Model.

3.1.2 Feature selection using BERT model

For this research work, we used of a deep learning algorithm called Bidirectional Encoder Representations from Transformers (BERT) in extracting relevant features from the tweet text and how they relate to the class label of the tweet. To be able to use BERT, we first tokenised the tweets. By tokenisation, we mean breaking the sentences into word tokens and assigning numerical values to them. BERT base comes pretrained with 110M parameters 12 attention heads, 12 layers and 764 hidden layers from Book Corpus of 800M words and Wikipedia of 2,500M words and there is need to finetune this model to make it fit for the prediction task.

Firstly, the input text is transformed into a BERT acceptable format by tokenizing it and adding [CLS] and [SEP] tokens to specify the beginning and end of a sequence of text tokens. We specify a maximum value for this sequence as 80 and any sequence which falls short will be padded or truncated as the case may be. The length 80 was chosen after running a function through the entire tweets in the dataset to determine that their maximum

length is 73. The sequence includes a binary mask called the attention mask that determines whether a token is a real word or just padding. The mask would be 1 if the token contained [CLS], [SEP], or any other real word, and [PAD] with an attention mask of 0 otherwise. The BERT model will have 19826 input nodes which is the 70% of the total.

As depicted in figure 2.3, the tweet text will be fed into the BERT Tokenizer and used to train the pretrained BERT base model, the output from the top of the encoder stack is the feature extracted by the BERT Model which will be used to train our CNN model.

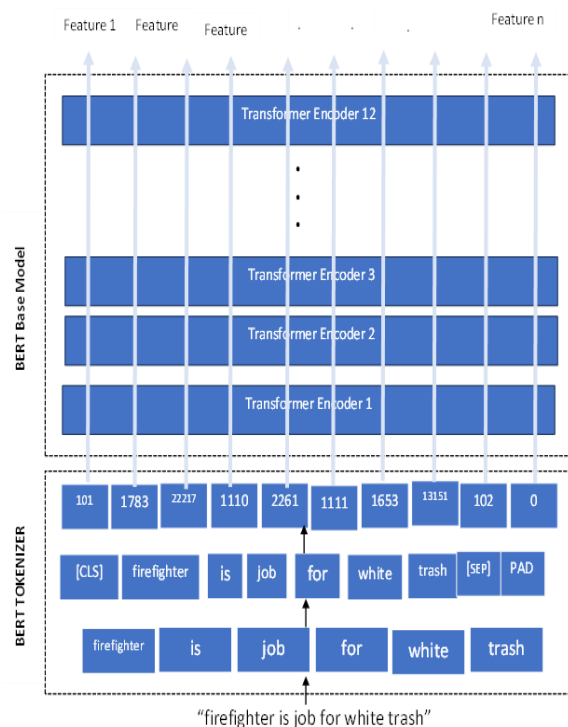


Figure 2.3: The BERT Layer for feature selection.

2.2 The Enhanced CNN Model

For this research, we propose a hybrid algorithm of CNN and LSTM. The fine-tuned BERT layer will be fed to a CNN model and LSTM was added to enhance the Neural network to prevent overfitting during the training time of the model by introducing remember

or forget gates based on the importance of the feature to the overall outcome.

Convolutional neural networks (CNNs) for extracting spatial features and long short-term memory (LSTM) networks for processing sequential information are combined in the hybrid model. While LSTM manages temporal dependencies, CNNs identify local patterns within the data.

2.3 The Model Design

BERT contextual embeddings for each token in a sequence. Let's represent the embeddings for a sentence as a matrix E , where each row corresponds to the embedding of a token.

$$E = [e_1, e_2, \dots, e_n] \quad (2.1)$$

where n is the number of tokens in the sequence and e_i is the contextual embedding for the i -th token.

For better context, a 1D convolutional layer of CNN with multiple filters apply these filters to the contextual embeddings to extract local features. Let's denote the filters as F and the output feature maps as C .

$$C = [\text{Conv1D}(E, F_1), \text{Conv1D}(E, F_2), \dots, \text{Conv1D}(E, F_k)] \quad (2.2)$$

where $F_1, F_2, \dots, F_k$ are the convolutional filters, and k is the number of filters. Conv1D denotes the 1D convolution operation.

After applying the convolutional filters, it's common to use pooling layers (e.g., max pooling) to capture the most salient features from the local regions. This is to further reduce the dimensionality of the feature maps.

$$P = [\text{MaxPooling}(C[1]), \text{MaxPooling}(C[2]), \dots, \text{MaxPooling}(C[k])] \quad (2.3)$$

where P represents the pooled feature maps.

The output from CNN is flattened and represented as a matrix C , where each row corresponds to the feature map of a different filter.

$$C = [c_1, c_2, \dots, c_k] \quad (2.4)$$

where k is the number of filters, and c_i is the feature map for the i -th filter.

The CNN's output C is then passed as input sequence for the LSTM. Each row of C represents the local features extracted by a filter from the CNN.

$$X = [c_1, c_2, \dots, c_k] \quad (2.5)$$

where X is the input sequence for the LSTM.

The LSTM layers process the input sequence X to capture sequential dependencies and patterns in the local features. For a simplicity, for a single LSTM unit, we have:

$$i[t] = \text{sigmoid}(W_i \times X[t] + U_i \times h[t-1] + b_i) \quad (2.6)$$

$$f[t] = \text{sigmoid}(W_f \times X[t] + U_f \times h[t-1] + b_f) \quad (2.7)$$

$$g[t] = \tanh(W_g \times X[t] + U_g \times h[t-1] + b_g) \quad (2.8)$$

$$o[t] = \text{sigmoid}(W_o \times X[t] + U_o \times h[t-1] + b_o) \quad (2.9)$$

$$c[t] = f[t] \times c[t-1] + i[t] \times g[t] \quad (2.10)$$

$$h[t] = o[t] \times \tanh(c[t]) \quad (2.11)$$

where:

- i. $i[t]$, $f[t]$, $o[t]$ the input, forget, and output gates' activations at time step t .
- ii. $g[t]$ is the candidate cell state at time step t .
- iii. $c[t]$ is the updated cell state at time step t .
- iv. $h[t]$ is the hidden state at time step t .

- v. $X[t]$ is the input at time step t .
- vi. W and U are weight matrices, and b is the bias term.

The final hidden state of the LSTM can be passed through a fully connected layer to obtain the desired classification output.

Algorithm 1-The Model's Algorithm

Start

Input: x – tweet

Y – class label

Output: Prediction - Class of the tweet

Foreach tweet do:

Foreach word in tweet do:

Map word to BERT Token using

tokeniser

Add attention mask to vector

Return word embedding

End

Return (word embedding, attention mask)

End

apply word embedding to CNN)

apply convolutional layers with various filter **sizes**.

apply max pooling to capture important features from the convolutional outputs.

flatten the pooled features to create a feature vector.

pass the flattened feature vector to LSTM

apply the output to a fully connected layer to perform the final classification.

apply a ReLU activation function to obtain class probabilities.

Stop

Algorithm 2 – Hyper parameter tuning/ Optimisation

Start

Input:

loss function (cross entropy), learning rate, number of filters, dropout rate

Output:

optimal parameters

foreach batch

train the entire model end-to-end using the training dataset.

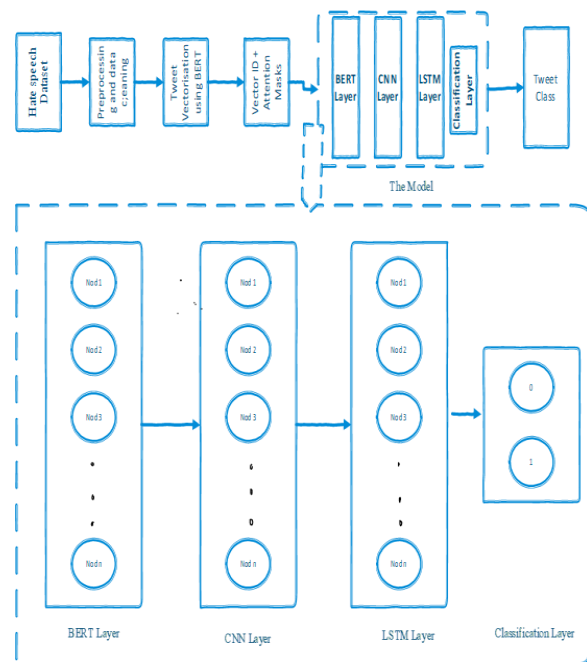
apply dropout and batch normalization to prevent overfitting

evaluate the model's performance on the validation dataset.

tune hyperparameters based on validation performance.

return optimal parameters

end



**Figure 2.4 The Architecture of the Enhanced
CNN Model**

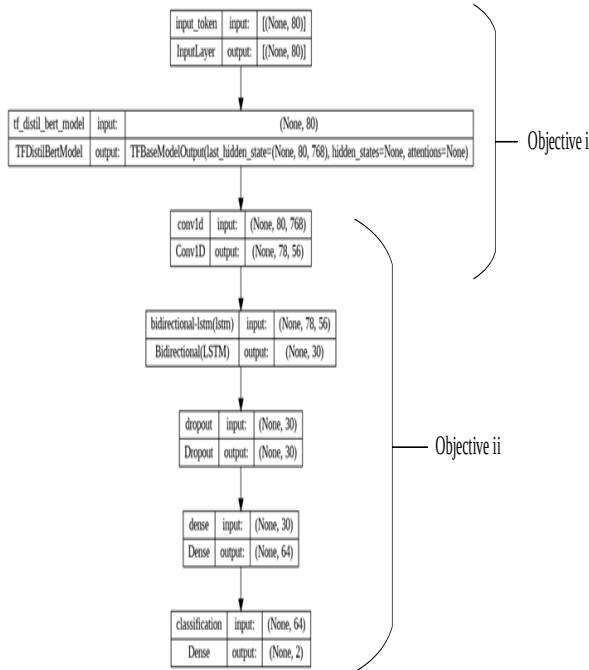


Figure 2.5 The Enhanced CNN Model

The model's architecture is shown in figure 2.4 and figure 2.5 shows the model itself which consist of layers of models from the BERT pretrained model to CNN and LSTM models. The dataset is tokenised using BERT tokenizer and used to fine-tune the BERT model for a better embedding representation and as input to enhance the CNN layer in feature selection. LSTM layer is added after the CNN layer to capture and retain more global information for better classification. Dropout layer is added to help prevent overfitting during training. The output of the dropout layer is passed to a fully connected layer and then to the classification layer.

The model is trained from a 70% split of the dataset in 10 epochs, which means that the training data is used 10 times during the learning process. The optimization technique used is AdamW with a batch size of 32 and a learning rate of 1×10^{-4} . Categorical cross entropy loss function is used. The objective was to minimize the loss function and get the hyper parameters at this

minimal point. The experiment was carried out onGoogle Colab cloud platform with system RAM of 12.7G and GPU RAM of 15G.

2.4 Performance Evaluation Metrics

The confusion matrix in Table 2.4 is utilised for evaluation and validation of the research work. By comparing the expected and actual values of the class label, false negatives (FN), true negatives (TN), true positives (TP), and false positives (FP) were computed. Then, in this work, F1-measures, recall (R), accuracy, and precision (P) will equally be used.

Table 2.4: Confusion Metrics of Results

	Predicted positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

True Positive (TP) are the sum of all the positive prediction which are actually positive.

False Positive (FP) are sum of all negatives that are incorrectly predicted to be positive.

False Negative (FN) are sum of all positives that are incorrectly predicted to be negative.

True Negative (TN) are the sum of all the negative prediction which are actually negative.

Precision (P): Is the measure of the proportion of positively predicted values which are correctly positive.

$$P = TP / (TP + FN) \quad (2.12)$$

Recall (R): It is the proportion of actual positives which are predicted positive.

$$R = TP / ((TP + FP)) \quad (2.13)$$

Accuracy: Is the measure of the proportion of instances correctly classified.

$$\text{Accuracy} = (TP + TN) / ((TP + TN + FP + FN + FP)) \quad (2.14)$$

F1-Measure: This is determined by averaging precision and recall harmonically.

$$F1 - Measure = 2 \times ((P \times R)) / ((P + R)) \quad (2.15)$$

Error rate: This is a measure of how often a classification model makes incorrect predictions. It is calculated as the ratio of the number of misclassified observations to the total number of observations. Error rate is typically represented as 1-Accuracy and is calculated by dividing the total number of occurrences in the dataset by the sum of all incorrect predictions.

3. RESULTS AND DISCUSSIONS

The proposed BERT-CNN-LSTM model's performance with un-enhance CNN only and CNN-LSTM using only word embeddings is shown in Table 3.0. From the results, while there is a marginal increase in the performance accuracy of CNN only from 88.3% to 89.08% in CNN-LSTM and to 96.1% in the proposed model, the F-1 score increased sharply from 0.69 to 0.93 and to 0.95 in the proposed model. Subsequently, figure 3.0 and figure 3.1 show the performance accuracies and the F-1 score of the un-enhanced and enhanced CNNs respectively.

Table 3.1 compares the proposed model to those of other researchers and subsequently depicted in figures 3.2 and figure 3.3 for the reported F-1 score and accuracies respectively.

It is noteworthy that some researchers reported only the accuracy performance from their models while others reported only the F-1 score. The model developed in this work has improved the detection accuracy of hate speech detection. The new model has shown an improved accuracy of over 0.45% over the bench mark paper by (Hasan *et al.*, 2022) as clearly

depicted in table 3.1 and subsequently in the figures 3.3.

4. **Table 3.0:** The Performance Comparison of the Proposed Enhanced BERT-CNN-LSTM Model with Unenhanced CNN and CNN-LSTM Models

Model	Method	F-1	Accuracy(%)
Un-enhanced CNN	Word embedding-CNN	0.69	88.30
Un-enhanced CNN-LSTM	Word embedding-CNN-LSTM	0.93	89.03
The proposed Model	BERT-CNN-LSTM	0.95	96.1

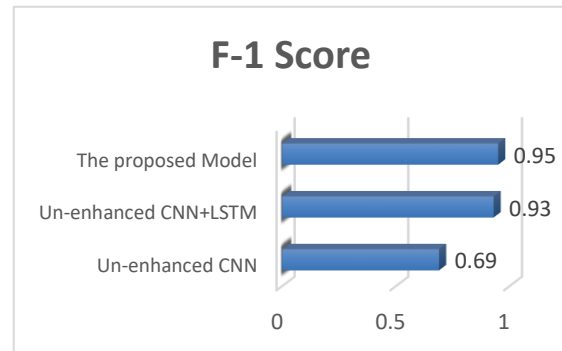


Figure 3.0 The F-1 Performance Comparison of the Proposed Model with un-Enhanced Models.

Table 3.1: The Performance Comparison of The Proposed Model with that of Other Researchers.

Author	Used Algorithm	F-1	Accuracy%
(MacAvaney <i>et al.</i> , 2019)	Multi-view SVM	-	80
(Zhang and Luo, 2019)	CNN+sCNN and CNN+GRU	0.94	
(Watanabe <i>et al.</i> , 2018)	unigrams and patterns	-	87.4
(Roy <i>et al.</i> , 2020)	(DCNN)	0.92	-
(Safaya <i>et al.</i> , 2020)	BERT-CNN	0.90	-
(Miok <i>et al.</i> , 2022)	Bayesian Attention Networks	0.90	91
(Okechukwu <i>et al.</i> , 2023)	TF-IDF Majority Voting	0.95	95
(Hasan <i>et al.</i> , 2022)	Transformer-CNN and MLP	0.94	95.63
The proposed Model	BERT+CNN+LSTM	0.95	96.1

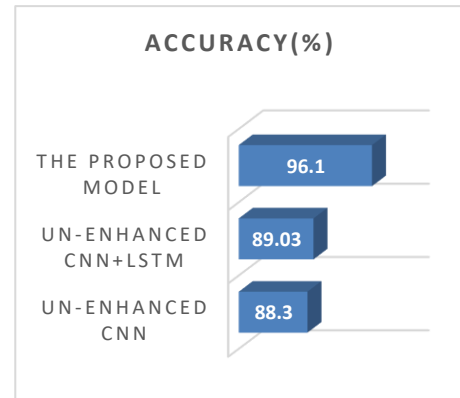


Figure 3.1 The Accuracy Performance Comparison of The Proposed Model and the Un-Enhanced Models.

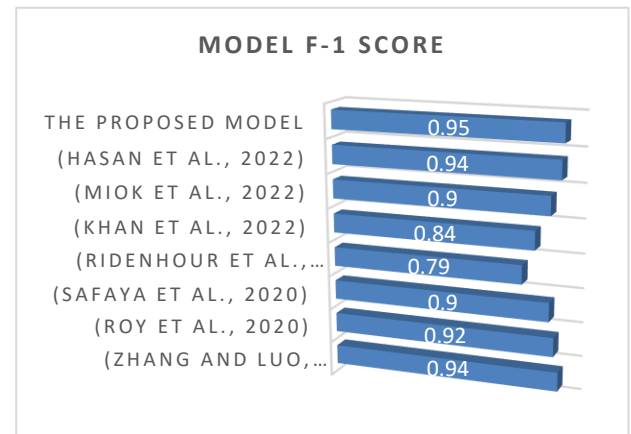


Figure 3.2 The F-1 Performance Comparison of the Proposed Model with other Researchers

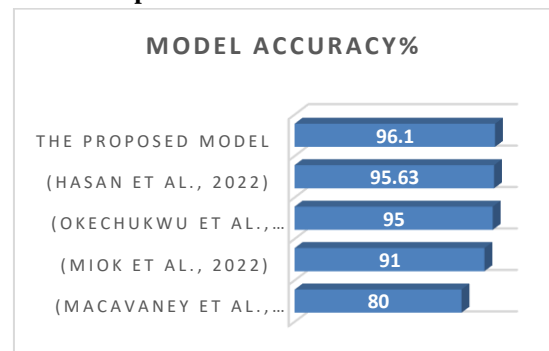


Figure 3.3 The Accuracy Performance Comparison of Proposed Model with other Researchers

4. CONCLUSION

Online hate speech is on the rise and has the potential to disrupt regional and global peace and security, which has alarmed many nations around the world, including the United Nations. Since they can be used to abuse, intimidate, promote or incite violent actions against a person or a group of individuals based on their sexual orientation, gender, color, and socio-political inclinations. It is critical to identify these harmful statements as soon as they occur to prevent translation of such statements to physical realities. This work successfully used the BERT Natural Language Processing Model to improve a combination of CNN and LSTM models to classify offensive and hateful tweets and achieved a classification accuracy of 96.1% and an F1-score of 0.95. This model when implemented and used in security policies will help enforce a secure climate from both psychological and physical harm resulting from hate related crimes on the cyber space.

4.2 Recommendations

Future Research in this area presents promising opportunities to fill in the gaps in the following existing Areas:

Multi-classification of hate speech: One future direction could be to develop models that can classify different types of hate speech, such as racism, sexism, appearance-related, intellectual, and political as equally recommended by Alkomah and Ma, (2022).

Dataset comparison methodology: There is a need for a new methodology for dataset comparison so that datasets can be rigorously compared (Alkomah and Ma, 2022).

Generalization of models: Hate speech detection models need to be more generalizable across different settings(Yin and Zubiaga, 2021).

Evaluation of effectiveness: There is a need to evaluate the effectiveness of different popular detection models on diverse hate speech datasets(Guansong Pang, 2022).

Identification of obstacles: Future research should identify the main obstacles to generalization in hate speech detection (Yin and Zubiaga, 2021).

Improvement of generalization: Future research should aim to improve generalization in hate speech detection (Yin and Zubiaga, 2021).

Integration of traditional and deep learning-based classifiers: Future research could explore the combination of traditional classifiers and deep learning-based classifiers for hate speech detection (Guansong Pang, 2022).

Competing definitions: Future research should address the challenges posed by competing definitions of hate speech, which can make it difficult to directly access and compare different datasets(MacAvaney *et al.*, 2019).

Comparison of different approaches: Future research should compare different approaches to hate speech detection and identify the most effective methods(Malik *et al.* 2022).

4.3 Contribution to Knowledge

The thesis contributed the following to knowledge;

- 1 A novel BERT-CNN-LSTM model to detect hate speech online developed.
- 2 Improved the detection accuracy of hate speech detection by reaching an accuracy of 96.1%. and thus, has

reduced the rate at which hate speeches could escape the radar without being detected.

- 3 This research has contributed a tool that when integrated into security polices enforcements, can secure the internet space from hate and hate related crimes thereby, protecting the users of the cyber space.

REFERENCE

- Alkomah, F., & Ma, X. (2022). A Literature Review of Textual Hate Speech Detection Methods and Datasets. *Information (Switzerland)*, 13(6), 1–22. <https://doi.org/10.3390/info13060273>
- Allan, R. (2017). *Back to Newsroom Meta Hard Questions: Who Should Decide What Is Hate Speech in an Online Global Community?* <https://about.fb.com/news/2017/06/hard-questions-hate-speech/>
- American Library Association. (2017). *Hate Speech and Hate Crime | Advocacy, Legislation & Issues*. <https://www.ala.org/advocacy/intfreedom/hate>
- Castañó-Pulgarín, S. A., Suárez-Betancur, N., Vega, L. M. T., & López, H. M. H. (2021). Internet, social media and online hate speech. Systematic review. *Aggression and Violent Behavior*, 58, 101608. <https://doi.org/10.1016/j.avb.2021.101608>
- Council of Europe. (2023). *Hate speech and violence - European Commission against Racism and Intolerance (ECRI)*. <https://www.coe.int/en/web/european-commission-against-racism-and-intolerance/hate-speech-and-violence>
- Davidson, T., Warmesley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. *Proceedings of the 11th International Conference on Web and Social Media, ICWSM 2017, April*, 512–515. <https://doi.org/10.1609/icwsml.v11i1.14955>
- Guansong Pang. (2022, March 5). *Deep Learning for Hate Speech Detection: A Large-scale Empirical Evaluation | by Guansong Pang | Towards Data Science*. <https://towardsdatascience.com/deep-learning-for-hate-speech-detection-a-large-scale-empirical-evaluation-92831ded6bb6>
- Hasan, A., Sharma, T., Khan, A., & Hasan Ali Al-Abyadh, M. (2022). Analysing Hate Speech against Migrants and Women through Tweets Using Ensembled Deep Learning Model. *Computational Intelligence and Neuroscience*, 2022. <https://doi.org/10.1155/2022/8153791>
- MacAvaney, S., Yao, H. R., Yang, E., Russell, K., Goharian, N., & Frieder, O. (2019). Hate speech detection: Challenges and solutions. *PLoS ONE*, 14(8), 1–16. <https://doi.org/10.1371/journal.pone.0221152>
- Malik, J. S., Pang, G., & Hengel, A. van den. (2022). *Deep Learning for Hate Speech Detection: A Comparative Study*. https://datahack.analyticsvidhya.com/contest/practice-problem-twitter-sentiment-analysis/?utm_
- Miok, K., Škrlić, B., Zaharie, D., & Robnik-Šikonja, M. (2022). To BAN or Not to BAN: Bayesian Attention Networks for Reliable Hate Speech Detection. *Cognitive Computation*, 14(1), 353–371. <https://doi.org/10.1007/s12559-021-09826-9>
- Nations, U. (2023). *What is hate speech? | United Nations*. <https://www.un.org/en/hate-speech/understanding-hate-speech/what-is-hate-speech>
- Okechukwu, C., Idris, I., Ojieniyi, J. A., Adebayo, O. S., & Olarere, M. (2023). Hate and Offensive Speech Detection Using Term Frequency - Inverse Document Frequency (TF - IDF) and Majority Voting Ensemble Machine Learning Algorithms. *International Engineering Conference, Science, Cyber Security, Iec*. [http://repository.futminna.edu.ng:8080/jspui/bitstream/123456789/18703/1/Hate and Offensive Speech Detection.pdf](http://repository.futminna.edu.ng:8080/jspui/bitstream/123456789/18703/1/Hate%20and%20Offensive%20Speech%20Detection.pdf)
- 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>
- Safaya, A., Abdullatif, M., & Yuret, D. (2020). KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media. *14th International Workshops on Semantic Evaluation, SemEval 2020 - Co-Located 28th International Conference on Computational Linguistics, COLING 2020, Proceedings*, 2054–2059. <https://doi.org/10.18653/v1/2020.semeval-1.271>
- Schmidt, A., & Wiegand, M. (2017). A Survey on Hate Speech Detection using Natural Language Processing. *SocialNLP 2017 - 5th International Workshop on Natural Language Processing for Social Media, Proceedings of the Workshop AFNLP*

- SIG SocialNLP*, 2012, 1–10.
<https://doi.org/10.18653/v1/w17-1101>
- Watanabe, H., Bouazizi, M., & Ohtsuki, T. (2018). Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection. *IEEE Access*, 6, 13825–13835.
<https://doi.org/10.1109/ACCESS.2018.2806394>
- Yin, W., & Zubiaga, A. (2021). Towards generalisable hate speech detection: a review on obstacles and solutions. *PeerJ Computer Science*, 7, 1–38.
<https://doi.org/10.7717/PEERJ-CS.598>
- Zhang, Z., & Luo, L. (2019). Hate speech detection: A solved problem? The challenging case of long tail on Twitter. *Semantic Web*, 10(5), 925–945.
<https://doi.org/10.3233/SW-180338>
- Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10843 LNCS(March), 745–760.
https://doi.org/10.1007/978-3-319-93417-4_48