

Hybrid Deep Learning Architectural Framework for Analysis of Hateful Sentiment on Twitter (X)

Hyellamada Simon 

Department of Computer Science, Federal Polytechnic, Mubi, Nigeria.

Corresponding author: hyellasimon@gmail.com

Abstract

In recent years, the pervasive use of social media platforms, such as Twitter, has led to an exponential increase in the dissemination of information and opinions. However, this phenomenon has also given rise to the alarming prevalence of hateful sentiment, posing significant challenges for online communities and societal harmony. To address this issue, a Hybrid Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) Deep Learning Architectural Framework was designed specifically for the analysis of hateful sentiment in Twitter content. The framework combines the spatial feature extraction capabilities of CNNs with the sequential learning proficiency of LSTMs, aiming to capture both local patterns and long-term dependencies within the textual data. The framework achieves a comprehensive understanding of the sequential and contextual nature of language, thereby enhancing the accuracy of hateful sentiment analysis. To evaluate the effectiveness of the proposed framework, extensive experiments were conducted on two large datasets of labeled tweets containing hateful sentiments. The work was done in two tasks using different datasets and the results demonstrate superior performance in both tasks compared to existing state-of-the-art models. Task 1 generated a 0.82 macro F1-score and 0.92 weighted F1-score, and Task 2 generated a 0.82 macro F1-score and 0.90 weighted F1-score, showcasing the ability of the proposed architecture to discern subtle variations in hate speech, sarcasm, and disguised forms of harmful language. Beyond sentiment analysis, the implications of this research extend to the development of robust tools for fostering a safer online environment.

Keywords: Hybrid Deep Learning, Hate Speech, Sentiment Analysis, Twitter, Architectural Framework, CNN, LSTM.

Introduction

Social media platforms, especially Twitter, are widely used and have become indispensable resources for expression and communication. Tragically, this transparency has also led to the emergence of a serious problem: the spread of hate speech, fake news, derogatory language, and harmful content. Thus, it is essential to have efficient techniques for identifying and combating hate sentiment to preserve a positive online environment.

According to (Watanabe *et al.* 2018), the term "hate speech" refers to the use of aggressive, violent, or abusive language directed toward a particular group of people who share a common characteristic. This group may include members of the same gender, ethnic

group, or racial discrimination, ideology, or religion. Although most social networks and microblogging sites prohibit hate speech, it is practically hard to control everything posted on these sites due to the volume and frequency of messages.

Significant unrest has resulted from the dissemination of hate speech among different political parties, religious organizations, and cultural groups (Simon *et al.*, 2022). Unfortunately, it can be difficult and time-consuming to identify and remove hurtful statements from Twitter due to the massive volume of tweets that are generated every second. Scholars promoted the argument of personal independence to support the broad protections afforded to freedom of expression. Gelashvili (2018) argues that the individual's dignity and the greater good can both be served by exercising one's right to free speech. This suggests that preserving one's feeling of autonomy depends on having the right to express oneself in all forms, including political ones. The world of communication has changed due to digital technology, and the way Internet corporations develop their technological infrastructure has a significant impact on the right to freedom of expression. Balkin (2004) conjectured that, despite the new opportunities brought about by the digital revolution, the Internet era has made it necessary to rethink freedom of speech from a different perspective and that it is crucial to adapt to the challenges it also brought.

Therefore, detecting and addressing these harmful sentiments is essential to maintaining a safe and inclusive online environment. This paper explores the potential of a Hybrid CNN and LSTM deep learning architecture as an innovative solution to identify and analyze this escalating problem of hateful sentiment on Twitter.

The subsequent section of this paper presents the related works in section 2, section 3 presents experimental design and model analysis technique, then section 4 presents results and discussion of results.

Literature Review

Previous research has explored various methods for detecting hate speech and offensive content on Twitter and other social media platforms. These approaches span traditional Natural Language Processing (NLP) techniques to advanced deep learning methodologies. The work of (Chhabra and Vishwakarma 2023), who provided a thorough examination of definitions of hate speech, as well as the motivations behind detection and traditional methods of textual analysis which are crucial for identifying hate speech are noteworthy contributions. Contemporary approaches to hate speech detection were also discussed, emphasizing manually designed feature-based and deep learning-based algorithms that considered multimodal and multilingual inputs and enumerated the advantages and disadvantages of each technique. The survey also provides well-known benchmark datasets for the identification of hate speech and offensive language, detailing the challenges associated with these datasets, best practices for achieving optimal classification outcomes, and dataset attributes such as sample size, modalities, languages, classifications, etc.

Ayo *et al.* (2020) presented an overview of machine-learning techniques for detecting hate speech in Twitter data streams. The study found that while there is still a lot of research being done on characterizing hate speech in Twitter data streams, there hasn't been much done on threshold settings, fragmentation issues, or general metadata architecture design. The approaches for categorizing hate speech in literature are limited by the aforementioned issues, even though they alleviate some of the concerns with Twitter data streams. The benchmark datasets for hate speech that the authors have provided can be used to assess how well classification techniques work.

Similarly, (Jahan and Oussalah 2023) conducted a thorough study of the literature for all works on hate speech published between 2000 and 2021 using the Google Scholar and ACM digital library databases. It was discovered that 463 papers met the inclusion and exclusion criteria for PRISMA. The results show that the SVM technique and several TF-IDF feature types were initially the most popular. The methodologies used to analyze hate speech, however, underwent a rapid transition as a result of advances in deep learning technology. The scientific community favored using various word embedding techniques using CNN and RNN architectures. In comparison to more conventional machine learning models like SVM, LR, NB, and RF models, various comparative studies conducted between 2017 and 2021 have demonstrated the advantages of deep learning models like CNN, and RNN employing word2Vec, GloVe, and FastText. However, it is still early days for deep learning model comparison. For instance, when contrasting CNN with RNN models, mixed results were found. Despite some authors' claims to the contrary, it has been discovered that GRU was better suited for long sentences while CNN seems to be beneficial in extracting characteristics. Furthermore, several articles suggested that combining two or more deep learning models outperformed using just one deep learning model. For instance, CNN plus LSTM and CNN + GRU both outperformed the use of LSTM and CNN alone.

Reddy *et al.* (2023) provide an automated hate speech detection algorithm that was tested on a publicly available Twitter dataset to address the issue of the prevalence of hate speech on social media. To balance an unbalanced dataset, the proposed approach prioritizes data pre-processing, including stemming, term frequency-inverse document frequency (TF-IDF) for feature extraction, and several sampling techniques (random sampler, synthetic minority over-sampling technique (SMOTE), and ALL-KNN). To lessen overfitting and assess performance, hold-out cross-validation was used to train and test the machine learning classifiers for logistic regression, support vector machine (SVM), and k-nearest neighbor (k-NN). Utilizing criteria like accuracy, precision, and confusion matrix, the performance was measured. According to the findings, the logistic regression classifier utilizing the SMOTE approach performed the best, with an accuracy of 82%, a macro average of precision of 80%, and recall of 82%, and an F1-score of 79%.

Modern speech detection techniques for detecting hate speech are examined in depth (Yadav *et al.*, 2023). The researchers' use of a deep multi-task learning system to combine the knowledge gained from various tasks in a variety of benchmark datasets demonstrates

the usefulness of the approach and provides strong support for the most cutting-edge models. Such Deep Learning methods are applied to the benchmark datasets to identify hate speech. To detect hate speech, the researchers also carried out a comparison analysis using a variety of machine learning methods, including deep learning and word embedding techniques. Deep learning techniques outperformed machine learning techniques in the comparative analysis and discussion, which revealed that both had good results. However, deep learning without word embedding methods outperforms and achieves more than 95% performance efficiency outcomes using LSTM over state-of-the-art techniques. Deep learning with word embedding techniques has achieved more than 92% performance using Bi-GRU-GloVe.

Similarly, (Fazil *et al.* 2023) examine hate speech and offers a system for automatically detecting such material. To classify hateful content, the researchers presented an attentional multi-channel convolutional-BiLSTM network. To capture semantic relations at different windows, our model makes use of existing word representation approaches in a multi-channel environment that has many filters with variable kernel sizes. An attention-aware stacked 2-layer BiLSTM network processes the encoded representation from several channels. The output of a two-layer BiLSTM stack was concatenated, weighted by an attention layer, and then sent via a dense layer. The text is finally categorized by an output layer that uses a sigmoid function. On three benchmark datasets relevant to Twitter, the researchers looked at the performance of the proposed model using four evaluation indicators. In a comparative examination, the model outperformed the same number of baseline models and five state-of-the-art models. The ablation investigation demonstrates that the proposed model performs best when channels and attention mechanisms are excluded. The usage of their ideal values is determined by the empirical analysis examining the effects of various word representation strategies, optimization algorithms, activation functions, and batch size on the provided model.

Kemal *et al.* (2023) introduced a deep learning approach to recognize bilingual hate speech in texts written in Afaan Oromo and Amharic on social media. The researchers used an existing binary Amharic dataset and a bilingual dataset created from newly gathered Afaan Oromo texts from Facebook to create the models. The dataset used contained binary classes "Free" and "Hate". Also, the Word2vec feature representation was used to create bidirectional RNNs and attention techniques. The word2vec model was trained following the skip-gram model. Five-fold and ten-fold cross-validation were both used to train the models. Using 5-fold cross-validation on the dataset, the results demonstrate good performance. The best-performing model was then chosen after several experiments, the BiLSTM model with an accuracy and F1-score of 94.2% outperformed them all.

Kar and Debbarma (2023) suggested an ideal feature extraction and hybrid for multiple features extraction from provided code-mixed texts after preprocessing. The retrieved characteristics were then optimized using a quantum search optimization approach, which lessens data dimensionality problems in the following detection phase. A hybrid diagonal-

gated recurrent neural network (Hyb-DGRNN) is then used to identify hate speech and assess sentiment in the language used. The authors ran experiments on the HASOC 2019 dataset to verify the viability of our suggested FE-DGRNN approach. The dataset contains postings that were published in English, Hindi, and German, enabling the evaluation of the approach's effectiveness across several languages. The accuracy of FE-DGRNN for Task-1, Task-2, and Task-3 for the multilingual code-mixed texts dataset was discovered to be 87.74%, 88.98%, and 84.74%, respectively, from the simulation results. Overall, compared to existing classifiers, the proposed FE-DGRNN technique significantly improves accuracy, precision, recall, and F-measure, suggesting that it has the potential to be a reliable and efficient tool for sentimental analysis and hate speech identification in texts with multiple languages and code mixtures.

Del Valle-Cano *et al.* (2023) provided a thorough analysis of earlier initiatives and tactics for identifying and suppressing hateful content on the social network (Twitter), as well as a cutting-edge categorization method based on user profiles, the surrounding social context, and generated tweets. The research achieved a three-fold contribution (i) a boost in HaterNet's efficiency, an expert system designed in conjunction with the Spanish National Office Against Hate Crimes of the Spanish State Secretariat for Security (Ministry of the Interior) that can recognize and track the development of hate speech on Twitter using the architecture of an LTSM + MLP neural network. Additionally, a BERT-based model called HaterBERT has been developed and evaluated using the public dataset for HaterNet, yielding results that demonstrate a significant enhancement; (ii) a method for building a user database that takes the form of a relational network to deduce textual and importance features. This contribution, SocialGraph, has been independently evaluated using a variety of conventional machine learning and deep learning techniques, demonstrating its value in identifying haters; (iii) the final model, SocialHaterBERT, integrates the first two strategies by examining features other than those found in the text. Results of experiments show that the final contribution significantly enhances results, creating a new field of study that transcends textual limits and opening the door for further research on combined models from a diachronic and dynamic perspective.

Mazari *et al.* (2023) suggested a multi-aspect hate speech detection method based on labeling text with a variety of categories, such as "identity hate," "insult," "threat," "severe toxic," And "obscene". The suggested method used Deep Learning (DL) models in combination with pre-trained Bidirectional Encoder Representations from Transformers (BERT) models to create a variety of ensemble learning architectures. Bidirectional Long-Short Term Memory (Bi-LSTM) and/or Bidirectional Gated Recurrent Unit (Bi-GRU) are stacked on GloVe and FastText word embeddings to create the DL models that are used. These models and BERT are trained separately on multi-label hostile datasets and combined for tasks involving detecting hate speech on social media. The method showed that encoding texts with modern word embedding methods like FastText, GloVe, Bi-LSTM,

and Bi-GRU together with BERT can provide models that have improved the ROC-AUC score to 98.63%.

Ayo *et al.* (2021) created a probabilistic clustering model for Twitter to address the issues of hate speech classification. A crowd-sourced panel of experts categorized the collected hate tweets into two categories: hate speech and non-hate speech. A metadata extractor was utilized to gather tweets containing hate speech keywords. They augmented the TF-IDF model's features with themes inferred by a Bayes classifier. The researchers automatically sorted real-time tweets into the appropriate topic clusters using a rule-based clustering technique. Then a semantic fuzzy rule and a score computation module were used to classify hate speech. With an F1-score of 0.93 and a 5-fold cross-validation, the evaluation findings showed that the proposed model outperformed other methods for detecting hate speech. When compared to related models, the generated model for categorizing hate speech fared better. Archiving an F1-score of 91.50 and AUC of 0.96 in comparison to other methods, the work shows outstanding performance.

The work of (Malik *et al.* 2022) was divided into three categories, including word embedding-based deep methods, transformers-based deep methods, and traditional (shallow) classification methods (Support Vector Machine (SVM), extreme gradient boosting (XGB), and multi-layer perceptrons (MLP) with TF-IDF-based word representations). They used Transformer encoder architecture by the BERT family of models to process each input text token in the context of all tokens that came before and after it. Cross-entropy loss was used to train the classifier. Fourteen popular/state-of-the-art models, including three TF-IDF-based classical classifiers, three Glove-based models, and eight transformer-based models were produced by combining all of these factors in the study. The findings indicate that the transformers-based models produced the greatest metrics, with BERT+CNN generating the best macro F1-score of 0.76 and the best weighted average F1-score of 0.91 (Davidson *et al.*, 2017).

Kapil and Ekbal (2020) proposed a deep neural network-based multi-task learning approach to hate speech detection on Twitter using CNN, LSTM, and CNN + Gated Recurrent Unit (GRU). They used Word embedding and character embedding for feature representation. The result generated with CNN achieved 0.72, and LSTM generated 0.62. Also, CNN + GRU achieved 0.73. The models provided a comprehensive representation of text data, making it suitable for hate speech detection tasks. Likewise, (Salminen *et al.* 2020) proposed a method for identifying hate speech on multiple social media platforms by combining SVM, LR, NB, XGBoost, and neural network approaches with feature representations such as Bag-of-Words, Word2Vec, TF-IDF, BERT, and their combinations. In the experiment, XGBoost outperformed them all.

Similarly, (Faris *et al.* 2020) developed a hybrid deep learning model for hate speech detection on Twitter using an Arabic dataset for training and testing of the model. To classify hate speech in a particular code-switched pair of Hindi and English (commonly known as Hinglish), (Kapoor *et al.* 2019) also employed transfer learning. They concluded

that the task was challenging but valuable despite Hinglish's non-fixed syntax, lexicon, semantics, and spelling. In Lakhotia (2020) a hate speech detection model was proposed for identification of tweets based on LR, RF, NB, and SVM. The models used TF-IDF as a feature extration technique and obtained the best result with LR achieved 0.82 macro F1-score and 0.89 weighted F1-score. This approach is straightforward and interpretable approach for text analysis, particularly when the goal is to predict continuous numeric values.

Ahluwalia *et al.* (2018) concentrated on creating machine learning models to instantly spot instances of sexism and violence against women in tweets. Ensembles of five classifiers were trained in a supervised manner on a bag of words (including words unigrams and bigrams) and scored better overall in the models that compared supervised and unsupervised learning approaches to sexism detection in tweets. Results with a 78.51% accuracy rate and macro-average F1 scores of 0.15 for category classification and 0.55 for target detection were obtained. The models were tested on the validation set after being trained on the training set. They suggested using an expanded labeled dataset to raise the neural networks' efficiency.

Davidson *et al.* (2017) employed machine learning techniques for hate speech detection on Twitter, and (Zhang *et al.* 2018), applied CNNs to identify offensive language. Similarly, (Park and Fung 2018) developed a hybrid CNN method to efficiently detect abusive language on Twitter. Based on three CNN models, the method classified offensive speech as either sexist or racist. A machine learning and neural network-based model for the automatic detection of abusive language on Twitter in the languages of English, German, and Hindi was also presented (Mujadia *et al.*, 2019).

Based on a combination of BiLSTM and CNN learning architectures, (Wiedemann *et al.* 2018) created a model for the detection of foul language on German Twitter data (GermEval 2018). Three transfer learning tasks were examined using the model to enhance the performance of categorization using background information. There are three types of category transfer: supervised (social media data annotated with nearly offensive language), weakly supervised (tweets annotated with emojis), and unsupervised (tweets annotated with topic clusters identified by Latent Dirichlet Allocation (LDA)). The findings suggested that transfer learning generally enhances the ability to identify offensive language. However, pre-trained models on the unsupervised topic clustering of tweets in conjunction with thematic user cluster data produced the greatest results.

A bidirectional LSTM-based neural network was created by (Garain and Basu 2019) for Twitter that can first detect aggressive behavior and then hate speech. The effort was centered on two distinct target demographics: immigrants and women, and English-language tweets were taken into consideration. The task was a three-step process to determine whether a tweet contained hate speech, whether it displayed aggressive behavior, and last, whether it was intended for an individual or a group of individuals. They concluded that the model completed the tasks with satisfactory results.

For the automatic detection of hate speech on Twitter in the Arabic region, (Faris *et al.* 2020) developed a hybrid deep learning approach. They gathered a dataset that records hate speech on Twitter from the Arabic-speaking world about various topics of discussion. A collection of features was extracted by the model from the dataset using the word embeddings approach. The word embeddings were fed into a hybrid CNN and LSTM network-based deep learning system. The proposed method successfully classified tweets as Hate or Normal with an accuracy rate of 71.68%.

Naveen and Kumar (2019) proposed an approach to automatically classify tweets on Twitter into three different classes: offensive, hateful, and clean. Using a dataset from Twitter, the authors experimented with n-grams as features and passed their term frequency-inverse document frequency (TFIDF) values to multiple machine learning models. They also performed a comparative analysis of the models considering several values of n in n-grams and TFIDF normalization techniques. The model presented the best results with 95.6% accuracy when it was evaluated on test data. The work presented a module that serves as an intermediate between user and the Twitter.

Molina-Gonzalez et al. (2019) proposed an ensemble learning to detect the aggressiveness of Mexican Spanish tweets given texts containing offensive messages that ridicule or embarrass specific targets. The model was based on a machine learning approach using several supervised classifiers, such as Support Vector Machine, Decision Tree, Multinomial Naive Bayes, Random Forest and Logistic Regression, and the use of n-gram features. After experimenting with each classifier independently, they came up with Logistic Regression (LR), Multinomial Naive Bayes (Multinomial NB), and SVM classifiers. The combination of the method was done via a majority voting ensemble classifier which increased the Macro F1-score in all the classifiers.

Materials and Method

Data Collection and Preprocessing

This research used two datasets. Firstly, a substantial dataset of Twitter posts containing hateful sentiments was meticulously collected via the Twitter Application Program Interface (D1-Twitter). The dataset was annotated with labels indicating the presence or absence of hate. The second dataset is a publicly available dataset from CrowdFlower (D2-CrowdFlower). Data preprocessing steps included conversion of emojis and emoticons to their equivalent words, conversion of chat words, data normalization, removal of stop words and special characters, lemmatizing, and tokenization.

Table 1: Labeled datasets distribution

Datasets	Total	Hate speech (1)	Non-hate (0)
Twitter (D1)	19858	2491	17367
CrowdFlower (D2)	24783	20620	4163

Hybrid CNN and LSTM Deep Learning Techniques

The proposed Hybrid CNN and LSTM deep learning architecture is designed to capture and analyze patterns within textual data effectively, enabling the accurate classification of hate sentiments on Twitter. The architecture comprises two main components.

Convolutional Neural Network Architecture

Local features are extracted from the input text by the CNN component. It uses several convolutional layers with various filter sizes to gather information at various information scales. CNNs are used to process data. For these data, matrix representations are employed. Data may take the shape of a 2-Dimensional matrix (image pixels) or a 1-Dimensional matrix (regular time intervals), which is a time series data. In practical use, CNNs are great models (Kresnakova et al., 2019). Convolution is a mathematical operation that is used to do calculations in the CNN layer. CNN's organizational chart is shown in Figure 1.

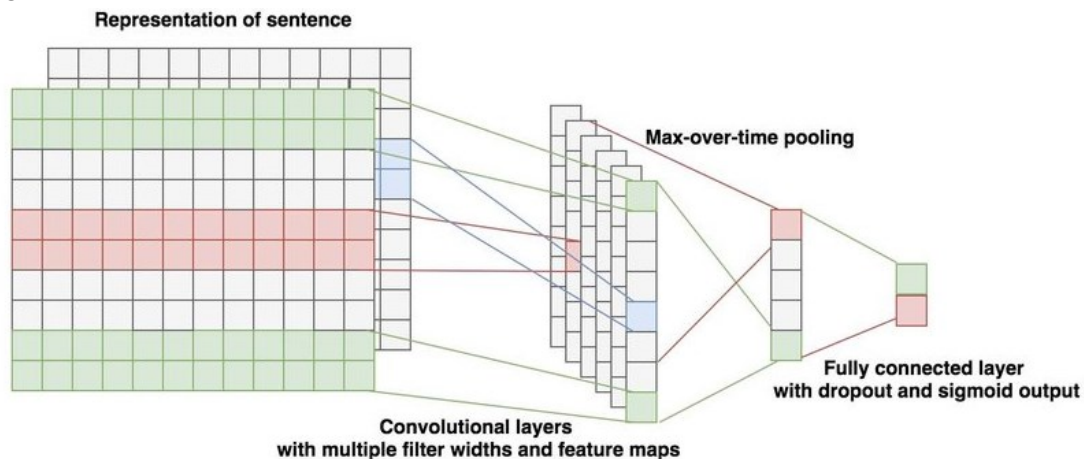


Figure 1: Structure of CNN (Kresnakova et al., 2019)

Consider the word sequence $w_{1:n} = w_1, \dots, w_n$, in which a dimension d embedding vector connects each word. Applying the same convolution filter or kernel to each window in the sequence after running a window of size k over a sentence results in a 1D convolution of size k . In other words, generating a result by taking the dot product of the weighted vector u and the sum of the embedding vectors in a certain window. This is typically followed by a non-linear activation function g . The CNN structure's Batista (2018) operation is responsible for the following equations. A window of words w_i, w_{i+k} , and so on is used to calculate the vector concatenation of the i th window.

$$x_i = [w_i, w_{i+1}, \dots, w_{i+k}] \in R^{k \times d} \quad (1)$$

After applying the Convolution-filter to each window, the scalar values for each i th window are revealed, and they are as follows:

$$r_i = g(x_i \cdot u) \in R \quad (2)$$

Adding more filters $u_i, u_l, \dots$ which is represented as a vector multiplied by a matrix U plus an extra bias term (b):

$$r_i = g(x_i \cdot U + b) \quad (3)$$

where $r_i \in \mathbb{R}^l$, $x_i \in \mathbb{R}^{k \times d}$, $U \in \mathbb{R}^{k \cdot d \times l}$, and $b \in \mathbb{R}^l$

In this study, the convolutional layers produce a tensor after receiving a series of embedding vectors from the embedding layer. In the convolutional layer, the embeddings are passed through 100 1-Dimensional (Conv1D) CNN filters with a size of 256. The next step was to apply global Max Pooling specifically for each feature map that was produced via the filter. This helps identify whether an n-gram feature exists in the text that matches that feature, allowing learning a wider range of features.

Long Short-Term Memory Architecture

A recurrent neural network called the LSTM component was created to identify sequential dependencies in text data. The sequential processing of the text input enables the modeling of long-range connections and linkages within the data. The LSTM in question is a specific kind of RNN that serves as a memory cell. The LSTM structure used input, output, and forget gates. The structure additionally consists of further levels (activation layers for the input and output). Long-term dependencies and the vanishing gradient of the neural network are resolved by the LSTM (Alwehaibi & Roy, 2017). When processing sequence words in textual data, it is incredibly effective (Collobert et al., 2011).

$X(t)$ and $h(t)$ represent the input and output of the LSTM model at time t , as shown in Figure 2, which illustrates the LSTM structure. The output of the hidden layer is represented by $h(t-1)$ and the historical data as of time $t-1$ is represented by $c(t-1)$, respectively. For each forget-gate, input-gate, and output-gate at time t , the corresponding representations are f_t , i_t , and o_t . The new data that was introduced during the transformation is represented by both c_t and $c(t)$, which stand for the updated information at time t . The structure of LSTM is shown in Figure 2.

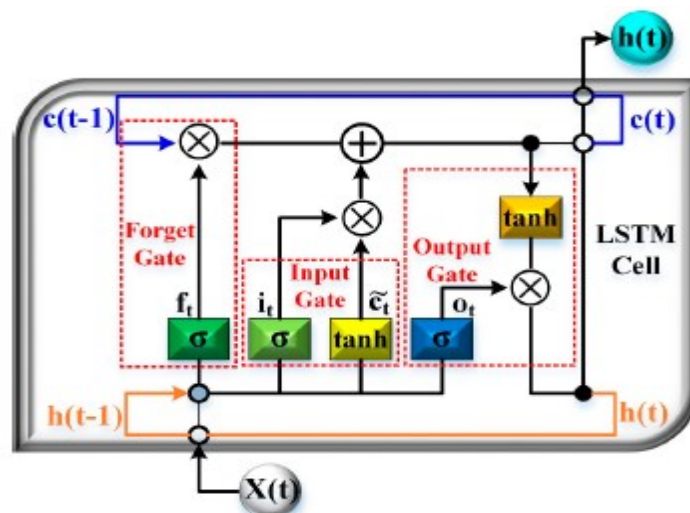


Figure 2: Network structure of LSTM (Chen et al., 2020)

Chen et al. (2020) state that the calculations performed within the LSTM's internal network structure can be completed in several steps by taking Figure 2 into consideration:

Step 1: At the forget-gate, the data of $X(t)$ and $h(t-1)$ are filtered using::

$$f_t = \delta(W_f X(t) + U_f h(t-1) + b_f) \quad (4)$$

Where W_f , U_f , and b_f represent the cycle-weight, input-weight, and the bias of the forget-gate respectively. Also, the logistic sigmoid function is represented by δ .

Step 2: The input-gate makes the decisions about which values need to be updated at this moment and which new data needs to be added at this stage:

$$i_t = \delta(W_i X(t) + U_i h(t-1) + b_i) \quad (5)$$

$$c_t = \tanh(W_c X(t) + U_c h(t-1) + b_c) \quad (6)$$

Where W_i , U_i , and b_i represent the cycle-weight, input-weight, and the bias of the input-gate, W_c , U_c , and b_c represent the cycle-weight, input-weight, and the bias of the cell respectively.

Step 3: To update the internal memory cell-state $c(t)$, the input-gate and forget-gate information is merged using:

$$c(t) = f_t \oplus c(t-1) + i_t \oplus c_t \quad (7)$$

where $\oplus$ is a concatenation operator

Step 4: The information output of the internal memory cell is controlled by the output-gate o_t as:

$$o_t = \delta(W_o X(t) + U_o h(t-1) + b_o) \quad (8)$$

Where W_o , U_o , and b_o represent the cycle-weight, input-weight, and the bias of the output-gate respectively.

Step 5: Finally, the activation function and output-gate are utilized to ascertain the LSTM network's output:

$$h(t) = o_t \oplus \tanh(c(t)) \quad (9)$$

The Hybrid CNN and LSTM Deep Learning Architecture

The embedding layer has been developed using the pre-trained Glove word embedding from Figure 3 below. Each word is encoded into a unique vector by the word embedding layer. Then a dropout layer was employed to prevent the tendency toward overfitting. Additionally, an LSTM structure was fed the higher-level phrase representations via CNN to generate a word representation. Word embeddings were fed into the CNN model, which used a 1-D matrix to generate a variety of feature maps. The LSTM model receives the encoded feature maps as input after performing convolution and pooling processes.

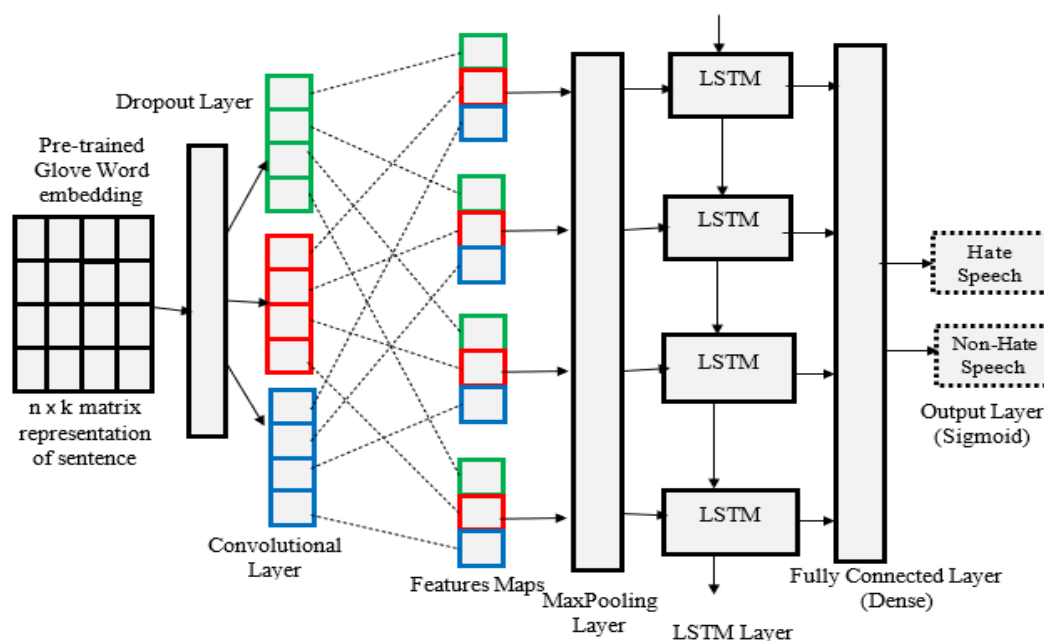


Figure 3: Designed hybrid CNN-LSTM model for detection of hate speech

An expression at the sentence level is the result of the long-term dependencies that the LSTM acquires. Following the feeding of the sentence-level representation into a fully connected network (Dense layer), the classification outcome is displayed via the SoftMax output. The long-term dependencies that the LSTM model records and the local encoded information that the CNN model retrieves are combined in the hybrid CNN-LSTM model. The final output from the LSTM layer is aggregated into a single matrix, which is received by a fully connected layer, which determines whether the result can be considered as hate sentiment or not.

Hybrid of CNN and LSTM Model Training

The proposed model was trained, validated, and tested on a laptop computer with Intel processor Core i3-4005U CPU at 1.70GHz & 1.70GHz, 4 GB RAM. The computer is equipped with Microsoft Windows 10 operating system, Python on Jupiter notebook. The training was done based on supervised learning and stopped at 10 epochs due to computational resource limitations. The experiment was done in two tasks (Task 1 and Task 2). Task 1 used the D1 dataset and Task 2 used the D2 dataset for training, validating, and testing.

The Python programming language TensorFlow (Abadi, 2016) was utilized for the backend construction of the hybrid model, which was trained and evaluated on a Jupyter Notebook. The front end of the model was created using the Keras library (Chollet, 2015). The first three of the four hidden layers that the researcher employed were turned on using the ReLU activation function. Using the sigmoid activation function, the output layer was turned on. For the output layer, two neurons were used, despite the dataset having two classes. The binary_crossentropy loss function, the sigmoid activation function utilizing the Adam

optimizer, the verbose rate at 1, the dropout at 0.2, and the batch size of 128 were all employed.

The original datasets were separated into train and test sets using the dataset splits found in the Python Sklearn package. The datasets were divided into train and test sets with a percentage of 75 and 25, respectively, and then randomly shuffled. The split approach of the datasets was selected over more intricate cross-validation techniques such as k-fold validation due to computational constraints.

Method of Model Analysis

The method of analysis for this classifier is the Precision, Recall, Accuracy, F1-measure, and confusion matrix. The equations are given as (Kapil & Ekbal, 2020):

Precision is the ratio of the total number of correctly identified texts as hateful over the total number of hateful texts in the corpus.

$$Precision = \frac{TP}{(TP + FP)} \quad (10)$$

Recall the sensitivity as to how much the classifier can identify actual hate texts as Hate.

$$Recall = \frac{TP}{(TP + FN)} \quad (11)$$

Accuracy is the proportion of texts that have been correctly classified as being either hateful or not over all the correctly and incorrectly classified texts.

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \quad (12)$$

F1- measure is the balance between precision and recall. Thus, is presented as:

$$macro_F1 = \frac{1}{C} \sum_{C=1}^C Fi \quad (13)$$

$$weighted_F1 = \frac{1}{N} \sum_{i=1}^n Fi \cdot Ni \quad (14)$$

$$Error\ Rate = \frac{FP + FN}{(TP + TN + FP + FN)} \quad (15)$$

Where: TP (True Positive), TN (True Negative), FP (False Positive), FN (False Negative), C (total number of classes), N (total number of samples per class), $N = \sum_{i=1}^n Ni$, $Fi = 2 \times Precision \times Recall$.

Results and Discussion

To evaluate the effectiveness of the proposed Hybrid CNN and LSTM deep learning architecture, a series of experiments were conducted using the two datasets (D1 and D2) dataset. The model was trained on a diverse range of hateful and non-hateful tweets and stop stopped at 10 epochs due to computational resource limitation. Evaluation metrics such as accuracy, precision, recall, and F1-scores were employed for the analysis to assess its performance.

For hate sentiment recognition in Task 1, Figure 4 shows the performance of the hybrid model training and validation based on accuracy.

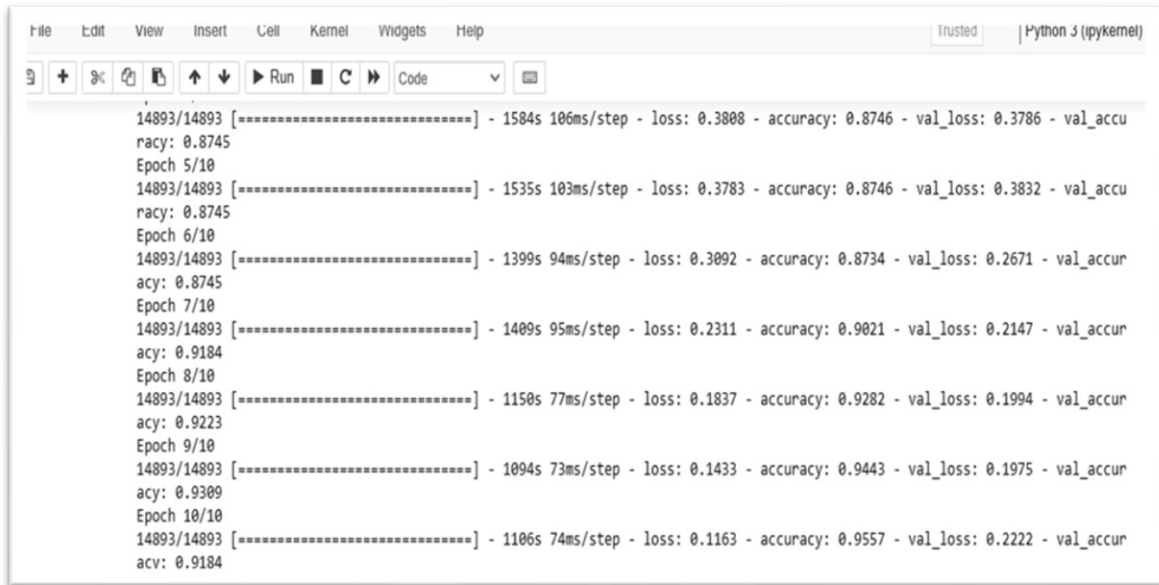


Figure 4: Training and validation performance of the hybrid model using D1 dataset

To identify hate sentiment on Twitter (Task 1), Figure 4 displays the results of training and validation. It can be shown that the hybrid model was trained and terminated at epoch 10 with a 95.57% training accuracy and a 0.1163 training loss, as well as a 91.84% validation accuracy and a 0.2222 validation loss. This showed that, when applied to the supplied D1 dataset, the model accurately identified hateful emotion with a 92% detection accuracy. Similar to this, Figure 5 displays the results of the hybrid model that was trained and validated using a CrowdFlower (D2) hate speech dataset that was obtained from the public domain.



Figure 5: Training and validation performance of the hybrid model using D2 dataset

Using the D2 dataset for both training and validation of the hybrid model, Figure 5 reveals a performance of 96.74% training accuracy and a 0.0915 loss. Furthermore, the validation accuracy was 91.35% with a loss of 0.2121. Thus, based on these results, it can be concluded that the hybrid model for identifying hateful sentiment fits well with an existing dataset that is available to the public (Lakhotia, 2020).

Analysis of Hybrid CNN and LSTM Model

After the models were trained using samples from the training sets, the efficiency of the model was evaluated by applying the trained model to the test unseen dataset. Figures 6(a) and 7(a) show the assessment results from the hybrid CNN-LSTM model, taking macro-F1 and weighted-F1 into consideration. Furthermore, each task's confusion matrix is shown in Figures 6(b) and 7(b).

	precision	recall	f1-score	support
0	0.96	0.95	0.95	4342
1	0.67	0.70	0.68	623
accuracy			0.92	4965
macro avg	0.81	0.83	0.82	4965
weighted avg	0.92	0.92	0.92	4965

Figure 6(a): Task 1 evaluation results from the hybrid CNN and LSTM model

		Predicted	
		Positive (+)	Negative (-)
Actual	Positive (+)	4122 (TP)	220 (FN)
	Negative (-)	185 (FP)	438 (TN)

Figure 6(b): Task 1 confusion matrix for the hybrid CNN and LSTM model result

The evaluation of the hybrid model in Task 1 to identify hateful sentiment using the D1 dataset was shown in Figure 6(a). The hybrid model performed well, as seen by the results in Figure 6(a), where the weighted-F1 and macro-F1 values were 0.92 and 0.82, respectively. The fact that the standard value of F1 is 1, and the hybrid model macro-F1 and weighted-F1 values are close to 1. Furthermore, the model performs well on the D1 dataset, as shown by the Error Rate (ERR) for Task 1 from the confusion matrix in Figure 6(b) of 0.0815.

	precision	recall	f1-score	support
0	0.90	0.55	0.68	833
1	0.92	0.99	0.95	4124
accuracy			0.91	4957
macro avg	0.91	0.77	0.82	4957
weighted avg	0.91	0.91	0.90	4957

Figure 7(a): Task 2 evaluation results from the hybrid CNN and LSTM model

		<u>Predicted</u>	
		Positive (+)	Negative (-)
<u>Actual</u>	Positive (+)	456 (TP)	377 (FN)
	Negative (-)	52 (FP)	4072 (TN)

Figure 7(b): Task 2 confusion matrix from the hybrid CNN and LSTM model result

Figure 7(a) is an evaluation result from the hybrid model to detect hateful sentiment in Task 2 by utilizing a publicly available dataset (D2). The result revealed that the hybrid model has proven efficient with a macro-F1 0.82 and weighted-F1 0.90 for the detection of hateful sentiment. Also, the ERR is 0.0865 considering Figure 7(b), this shows that the model has performed well on the D2 dataset since the F1-scores are close to 1 and the error rate is negligible.

Comparison of the Hybrid Model to the Chosen Baseline Models

To assess the efficiency of the proposed hybrid CNN and LSTM model, the results were compared to a subset of current findings published in several literature sources.

Table 2: Comparison of the hybrid model to the baseline model

Author(s)	Dataset	Model	macro-F1 score	weighted-F1 score
Lakhotia (2020)	D2	LR +TF-IDF	0.82	0.89
Kapil and Ekbal (2020)	D2	CNN+GRU	0.73	0.90
Task 1	D1	Hybrid CNN+LSTM+Glove pre-train Word embeddings	0.82	0.92
Task 2	D2	Hybrid CNN+LSTM+Glove pre-train Word embeddings	0.82	0.90

The baseline models were selected by taking into account the datasets used, the method of training the models, validation, and the evaluation metrics are displayed in Table 2. The baseline models were trained, validated, and evaluated using the same D2 datasets as used in Task 2 of this work and supervised learning techniques were employed, also the baselines used the same F1-score as the evaluation metric.

Discussion of Findings

The proposed hybrid CNN and LSTM deep learning models with Glove pre-trained word embeddings have proven good performance in terms of accuracy to detect hateful sentiment in both Tasks 1 and 2 carried out in this research, which achieved better performance than the baselines models. Based on the revelation from Task 1 using the D1 dataset, the hybrid model could detect instances of hateful sentiment with great accuracy of 92% and ERR of 0.0815 since the same Twitter dataset was used for the model's training and testing. Likewise, to determine the hybrid model's efficiency on an existing dataset in the public domain (Lakhotia, 2020), the model was trained and tested using a D1-CrowdFlower dataset to detect instances of hateful sentiment (Task 2). The result of this task shows great performance with an accuracy of 91% and ERR of 0.0865.

Considering Table 2, Task 1 results show the model generated macro-F1 of 0.82 and weighted-F1 of 0.90 to identify hateful sentiment on Twitter using the D1 dataset. The developed model used in Task 1 has outperformed the baseline models Lokhatia (2020) with 3% weighted-F1, it also outperformed Kapil and Ekbal (2020) with 9% macro-F1 and 2% weighted-F1 as seen in Table 2. The result in Task 2 generated 0.82 macro F1-score, and 0.90 weighted F1-score, which is higher than the baseline models. The Task 2 result outperformed Lakhotia (2020) on the same D2-CrowdFlower dataset with 1% weighted F1-score, also Kapil and Ekbal (2020) with 9% macro F1-score.

The experimental results demonstrated the superior performance of the Hybrid CNN and LSTM deep learning architecture in comparison to existing methods for hateful sentiment

analysis on Twitter. The model exhibited higher accuracy and F1-scores, signifying its ability to effectively detect and classify hateful sentiments. The combination of CNNs and LSTMs in the architecture proved to be a potent strategy for addressing the complex nature of hate sentiments on social media.

Conclusion

This research paper has introduced an innovative approach to hateful sentiment analysis on Twitter using a Hybrid CNN and LSTM deep learning architecture. The results indicate that the proposed model outperforms existing methods, offering a promising solution to the ongoing challenge of detecting hateful sentiments on social media. As hate sentiments continue to be a significant concern, the development of advanced and accurate detection techniques is crucial to fostering a safer online environment. This research contributes to the ongoing efforts to combat online hateful sentiments and offensive content.

Recommendation

More research should be carried out on social media platforms apart from twitter.

Statements and Declarations

Competing Interests: The author declares that there is no competing interest.

Funding Information: The author declares that this work has not received funding of any kind.

References

- 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*, 6, 13825-13835.
- Simon, H., Baha, B. Y., and Garba, E. J. (2022). A multi-platform approach using hybrid deep learning models for automatic detection of hate speech on social media. *Bima Journal of Science and Technology*, 6(2), 2536-6041. DOI:
- Gelashvili, T. (2018). *Hate Speech on Social Media: Implications of private regulation and governance gaps*. Master Thesis, Faculty of Law, Lund University.
- Balkin, J. M. (2004). Digital speech and democratic culture: A theory of freedom of expression for the information society. *New York University, Law Review*, 79(1).
- Chhabra, A., & Vishwakarma, D. K. (2023a). A literature survey on multimodal and multilingual automatic hate speech identification. *Multimedia Systems*, 29, 1203–1230.
- Ayo, F. E., Folorunso, O., Ibharalu, F. T., & Osinuga, I. A. (2020a). Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions. *Elsevier Computer Science Review*, 38 (100311).
- Jahan, M. S., & Oussalah, M. (2023). A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing*, 546(126232).
- Reddy, B. A. C., Chandra, G. K., Sisodia, D. S., & Anuragi, A. (2023). Balancing techniques for improving automated detection of hate speech and offensive language on social media. *In 2nd International Conference for Innovation in Technology (INOCON), Bangalore, India*, 1-8.

- Yadav, D., Sain, M. K., & Raj B. A. A. (2023). Comparative Analysis and Assessment on Different Hate Speech Detection Learning Techniques. *Journal of Algebraic Statistics*, 14(1), 29-48. ISSN:1309-345229
- Fazil, M., Khan, S., Albahlal, B. M., Alotaibi, R. M., Siddiqui, T., & Shah, M. A. (2023). Attentional Multi-Channel Convolution With Bidirectional LSTM Cell Toward Hate Speech Prediction. *IEEE Access*, 11, 16801-16811.
- Kemal, B. S., Abebe, T. U., Pendem, G.V.S.K, Krishna, T. G., & Gameda, K. A. (2023). Bilingual social media text hate speech detection for Afaan Oromo and Amharic languages using deep learning. *Journal of Namibian Studies*, 34(1), 250-281. ISSN:2197-5523 (online)
- Kar, P., & Debbarma, S. (2023). Multilingual hate speech detection sentimental analysis on social media platforms using optimal feature extraction and hybrid diagonal gated recurrent neural network. *Springer - Journal of Supercomputing*.
- Del Valle-Cano, G., Quijano-Sánchez, L., Liberatore, F., & Gómez, J. (2023). SocialHaterBERT: A dichotomous approach for automatically detecting hate speech on Twitter through textual analysis and user profiles. *Elsevier Expert Systems with Applications*, 216(11946).
- Mazari, A.C., Boudoukhani, N. & Djefal, A. (2023). BERT-based ensemble learning for multi-aspect hate speech detection. *Springer - Cluster Computing*.
- Ayo, F. E., Folorunso, O., Ibharalu, F. T., Osinuga, I. A., & Abayomi-Alli, A. (2021). A probabilistic clustering model for hate speech classification in twitter. *Elsevier Expert Systems with Applications*, 173 (114762).
- Malik, J. S., Pang, G., & Hengel, A. (2022). Deep Learning for Hate Speech Detection: A Comparative Study. *Computation and Language (cs.CL)*, 1 - 17.
- Davidson, T., Warmusley, D., Macy, M., & Weber, I. (2017). Automated Hate Speech Detection and the Problem of Offensive Language. *Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM)*, 512-515.
- Salminen, J., Hopf, M., Chowdhury, S. A., Jung, S., Almerexhi, H., & Jansen, B. J. (2020). Developing an online hate classifier for multiple social media platforms. *Human-centric Computing and Information Sci.*, 10 (1).
- Kapoor, R., Kumar, Y., Rajput, K., Shah. R. R., Kumaraguru, P., & Zimmermann, R. (2019). Mind your language: Abuse and offense detection for code-switched languages. *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, 9951-9952.
- Ahluwalia, R., Shcherbinina, E., Callow, E., Nascimento, A., & De Cock, M. (2018). Detecting misogynous tweets. *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval)*, 242-248.
- Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. *Springer*, 745-760.
- Park, J. H., & Fung, P. (2017). One-step and Two-step Classification for Abusive Language Detection on Twitter. *ALW@ACL*.
- Mujadia, V., Mishra, P., & Sharma, D. M. (2019). IIIT-Hyderabad at HASOC 2019: Hate speech detection. <http://ceur-ws.org/Vol-2517/T3-12.pdf>
- Wiedemann, G., Ruppert, E., Jindal, R., & Biemann, C. (2018). Transfer Learning from LDA to BiLSTM-CNN for offensive language detection in Twitter
- Garain, A., & Basu, A. (2019). The Titans at SemEval-2019 Task 5: Detection of hate speech against immigrants and women on Twitter. *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*. Minneapolis, Minnesota, USA, 494-497.
- Faris, H., Aljarah, I., Habib, M., & Castillo, P.A. (2020). Hate speech detection using word embedding and deep learning in the Arabic language context. *Proceedings of the 9th International Conference on*

- Pattern Recognition Applications and Methods (ICPRAM 2020)*, 453-460. <https://doi.org/10.5220/0008954004530460>
- Naveen, K., & Kumar, C. P. (2019). An effective scheme for detecting hateful and offensive expressions on twitter. *Intl. Journal for Innovative Engineering and Mgt. Research*, 8 (8), 204-209. ISSN 2456 – 5083
- Molina-Gonzalez, M., Plaza-del-Arco, F., Martin-Valdivia, T. M., & Urena-Lopez, L.A. (2019). Ensemble learning to detect aggressiveness in Mexican Spanish tweets. *Proceedings of the Iberian Languages Evaluation Forum (IberLEF)*, 495-501.
- Kresnakova, V. M., Sarnovsky, M., & Butka, P. (2019). Deep learning methods for fake news detection. *IEEE 19th International Symposium on Computational Intelligence and Informatics and 7th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Sciences and Robotics (CINTI-MACRo)*. <https://doi.org/10.1109/cinti-macro49179.2019.9105317>
- Batista, S. D. (2018). Convolutional neural networks for text classification. <https://www.davidsbatista.net/blog/2018/03/31/SentenceClassificationConvNets/>
- Alwehaibi, A., & Roy, K. (2018). Comparison of pre-trained word vectors for Arabic text classification using deep learning approach. *17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. <https://doi.org/10.1109/icmla.2018.00239>
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(08). 2493–2537.
- Chen, C., Hua, Z., Zhang, R., Liu, G., & Wen, W. (2020). Automated arrhythmia classification based on a combination network of CNN and LSTM. *Elsevier- Biomedical Signal Processing and Control*, 57 (101819). <https://doi.org/10.1016/j.bspc.2019.101819>
- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, G. D., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI 2016) USENIX Association, Berkeley, CA, USA*, 265–283.
- Chollet, F. (2015). Deep learning for humans. *Keras-team, GitHub*. <https://github.com/fchollet/keras>.
- Kapil, P., & Ekbal, A. (2020). A deep neural network-based multi-task learning approach to hate speech detection. *Knowledge-Based Systems*, 106458. <https://doi.org/10.1016/j.knosys.2020.106458>
- Lakhotia, N. (2020). Hate speech detection in social media using python. <https://github.com/NakulLakhotia/Hate-Speech-Detection-in-Social-Media-using-Python>