**DEEP LEARNING SOLUTIONS FOR LANGUAGE-RELATED PROBLEMS**

**Abstract**

Due to the increase of cases of improper language use on social media platforms, there is the need to have designs that can detect the use of improper language. This paper focuses on deep learning models, including CNN, LSTM, and ‘BiLSTM’ with Attention to classify the text as offensive. The data set was pre-processed and tokenized besides being padded as it was used for training the model. Again, ‘BiLSTM’ with Attention was superior and had the best results on the test set compared to others for its better context capture ability. LSTM learned the sequential dependencies of the words but did not learn bidirectionally like CNN and it also successfully detected the explicit words. As much as there is an imbalance of classes, it is clear that the attention mechanisms improve the overall classification performance. It stresses the need for promoting a more complex level of architectures that will make content moderation possible. Using transfer learning and data augmentation in the next work, it is possible to achieve better detection in low-resource languages and to check the effectiveness of the methods in various languages and contexts in the future.

# 1. Introduction:

This is because with the improvement in technology, there is an increase in the usage of improper language on social media, thus calling for artificial intelligence methods that can detect them. This paper aims to discuss deep learning in language-related issues, specifically in classifying the Offensive dataset into two categories, decent and rude language. Machine learning models, for example, Random Forest (RF) and Support Vector Machine (SVM), have also been used to predict toxic tweets, however, accurate and context-based models are transformers. For instance, the phrases including the words that can be considered even itches – such as “Go back to where you came from” are tricky and need a high level of differentiation between sarcasm and aggressiveness. Using deep learning architectures in this research, the intensity of automation in moderating, reduction of bias, and improvement of real-time detection will be achieved. It was found out that NN has a higher performance over traditional models for subtle language processing as a way of showing that content moderation in online environments can be made effective and scalable.

# 2. Related Work:

## 2.1 Introduction to Offensive Language Detection

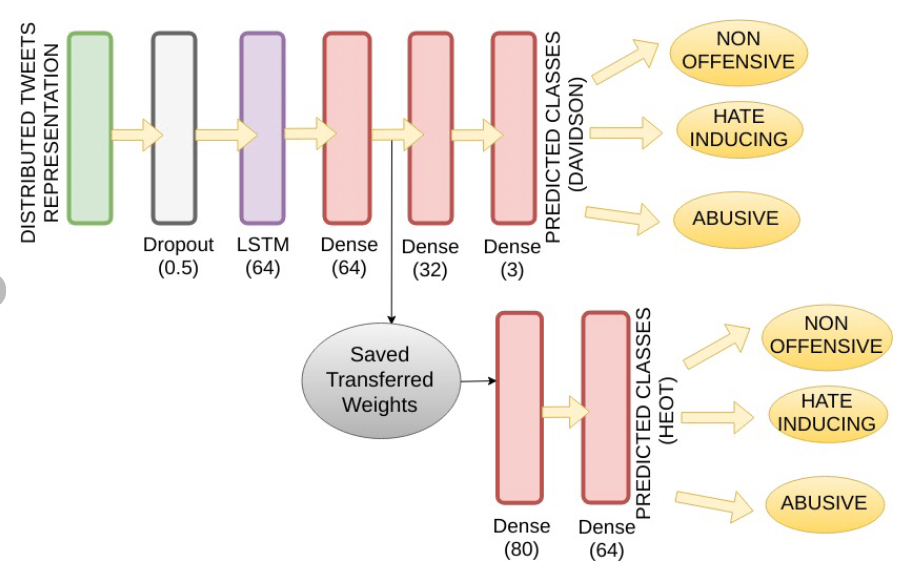
The use of inappropriate language continues to rise across social media platforms, which has consequently boosted efforts into finding ways of approaching the problem, using Deep learning. The traditional machine learning techniques have some effectiveness in the pre-processing language tasks, but are not efficient in handling sarcasm and contextual changes. Transformer-based models and RNNs have been among the most successful methods developed in the past years for abusive content detection. This section presents a literature review that discusses ten related works that address deep learning methods for detecting objectionable language, in a multilingual context, feature selection, and model fine-tuning.

## 2.2 Deep Learning for Offensive Language Classification

This paper specifically focuses on such deep learning models that include BERT and other variants coming under the transformer model. In the literature, Wadud *et al.* (2023) recently proposed Deep-BERT for detecting the presence of offending posts in multiple languages using transfer learning. It reveals the fact that using social media datasets to fine-tune the pre-trained models achieves higher accuracy than traditional approaches of machine learning classifiers. Likewise, Alotaibi *et al.* (2021) design a multi-channel deep learning system to detect cyberbullying with multi-Neural networks for the text. They paid much attention to the fact that integrating CNNs and LSTMs could enhance the significance of the classification model by overcoming the weaknesses inherent in individual architectures.

## 2.2 LSTM-Based Approaches for Offensive Text Detection

LSTM networks have been used extensively in text classification due to their ability to model long dependencies. Wadud *et al* (2022) developed LSTM-BOOST, which is a combination of LSTM and boosting to boost the classification results.



**Figure 1: LSTM based model for tweet classification**

(Source: Kapoor *et al*.,2019)

This work showed that there is value in combining TEL with DNN as a method for improving the generality of the models. Similarly, Ahmed *et al.* (2021) also used a deep neural network (DNN) for cyberbullying detection in Bangla with the help of LSTMs to process such textual data. The study was centered on the need to collect data from specific languages, and how there is a difference when it comes to language that is deemed as being offensive.

## 2.3 Transformer-Based Models in Offensive Language Detection

Offensive language identification has been revolutionized with the aid of contextualized embeddings through the use of transformers, especially BERT architectures. Chakravarthi *et al.* (2022) developed Dravidiancodemix, the dataset for offense language detection in code-mixed text and showed the efficiency of the transformers in the Multilingual environment. This work sheds light on the fact that fine-tuning on the code-mixed datasets improves accuracy of classification to a significant extent. The following authors have also used transformer-based models to classify hate speech in sports discussion. In the same year, Vujičić Stanković and Mladenović (2024) employed transformer-based models to do analysis and classification of hate speech on social media. In their study, they mapped how the fine-tuning in domain makes the model perform better, thus the versatility of the deep learning solutions.

## 2.4 Advances in Preprocessing and Feature Engineering

This paper aims to investigate how effective preprocessing and feature extraction methods affect the given deep learning models. In their study, Mullah and Zainon (2021) discussed various existing approaches for hate speech detection and focused on the preprocessing methods, which are tokenization, stemming, and embeddings. According to them, such word embeddings as Word2Vec and FastText would enrich defence profiles and optimise deep learning performance on such texts. Dewani *et al.* (2021) experimented with preprocessing techniques on Roman Urdu, which is a low resource language and proved that preprocessing according to the specific language can enhance the results. It established the significance of tuning the kind of preprocessing based on the general language used in the data set.

## 2.5 Cyberbullying Detection with Neural Networks

Deep learning models have also been very helpful in cyberbullying detection, as the models of different architectures work in different scenarios. CNNs, RNNs and the combined models have been studied by Neelakandan *et al.* (2022) in the context of deep learning-based cyberbullying detection. In their study, they discovered that transformer-based architectures surpassed the deep learning models, and now it is better to use transformer pre-trained language models for higher accuracy. Similary, Akhter *et al.* (2022) studied a comparative performance of traditional ML and Deep learning in detecting abusive language in social media. The study concludes that precision and retrial pointed out that CNN-LSTMs were new classifiers and have higher precision and retrial compared to other deep learning models, rather than being scalable and robust in real-world applications.

## 2.6 Literature Gap

However, there are still gaps in deep learning for CMC for the following reasons. There is a trend of high resource languages, and only a few works are addressed on low resource and code-mixed languages. In any case, although transformer models enhance contextual knowledge, they cannot be applied for real-time use due to their computational resources consumption. In addition, few works combine features including images or voice data with text-based models for improved detection. Regarding the hybrid architectures, Neelakandan *et al.* (2022) mentioned that they could enhance the classification but the appropriate integration of deep learning models for such forms is still a research issue.

## 2.7 Summary

In the case of detecting OWL, deeper systems have proved effective with transformer models performing better than previous models. This way, LSTMs and CNNs improve classification and, at the same time, advanced preprocessing increases accuracy. However, identifying languages that are multilingual or belong to low resource languages is still difficult. Developing hybrid architectures is beneficial, although the use of multi-modal approaches still lacks optimization when it comes to the ensemble of deep learning models.

# 3. Method:

The main approach used in this study can be described as the use of deep learning for text classification in detecting offensive language. Some of them include, data cleansing, converting raw text into vector form, selecting model, training model, model evaluation, and comparison of different architectures of the models. Three deep learning models were used, namely the Convolutional Neural Network (CNN), Long Short-Term Memory network (LSTM) and Bidirectional LSTM with Attention. The popular approach for exercising this type of reporting is laid out as follows:

**Data Preprocessing**

The dataset used in this work was collected from social media, and this means that a good deal of pre-processing was done to furnish quality input to the deep learning models. This was followed by a preprocessing of the text data by converting it to lower case, as this enhances equality. The process of tokenization was carried out with the help of the ‘Tokenizer’ function, which belongs to the ‘keras.preprocessing.text’ of ‘TensorFlow’, and for out-of-vocabulary tokens, a placeholder token was used. This was followed by the tokenized sequences, of which the sequences were truncated to 100 tokens as the input to all the models is of the same length.

The dataset was in the form of labelled instances, in which one of the outputs was either offensive or non-offensive. The first step was to preprocess the data, where categorical data was encoded using ‘LabelEncoder’, so that “YES”, which is a category, was represented by a numerical value of 1, while “NO”, which is also a category, was represented by a numerical value of 0. This was done to take care of the fact that the model must be trained with categorical outputs.

**Model Architectures**

Thus, this study attempted to compare and contrast three different architectures:

***Convolutional Neural Network (CNN):***

Local relationship in the given text data can be captured through convolutional filters in CNNs. The CNN model comprised an embedding layer with input size 10000 and embedding size 128, next, it was followed by two ‘Conv1D’ with filters 128 and the kernel size of 5. In order to down sample, max pooling layers namely ‘MaxPooling1D’ were applied. In the situation of classification, the final layer is fully connected with a ‘SoftMax’ activation function. The optimizer employed for the model was Adam, while the loss function that was used was the sparse categorical cross entropy.

***Long Short-Term Memory (LSTM) Network:***

LSTMs are used in the text classification as these models can handle sequential data and the long-term dependencies amongst them. LSTM started with an embedding layer, just like CNN, then an LSTM layer having 128 units in it. ‘DropOut’ operation of dimension one was used to avoid overfitting where the neurons of the layer are temporarily shut down during training. The model ended with dense layers, which were ‘Dense (64)’, followed by a dropout layer of dropout rate 0.5 to drop the substantial unit by 50 percent in anti-overfitting mechanisms to enhance the accuracy of the model, and the last layer, which was a ‘SoftMax’ layer, which classified the output.

***Bidirectional LSTM (BiLSTM) with Attention:***

Compared with the traditional ‘LSTMs’, ‘BiLSTMs’ are applied to learn the forward and backward sequences to make it easier to learn the context of the given sequences. The ‘BiLSTM’ model had an embedding layer, then Bidirectional Layer LSTM ‘(128, return\_sequences=True)’. An attempt to apply attention and focus on the most important elements within text and filter the rest was made with the help of ‘TensorFlow’s’ Attention module. The architecture of the model was then being completed with the ‘GlobalAveragePooling1D’ layer, additional dense layers for classification, and the SoftMax layer at the end.

**Model Training and Evaluation**

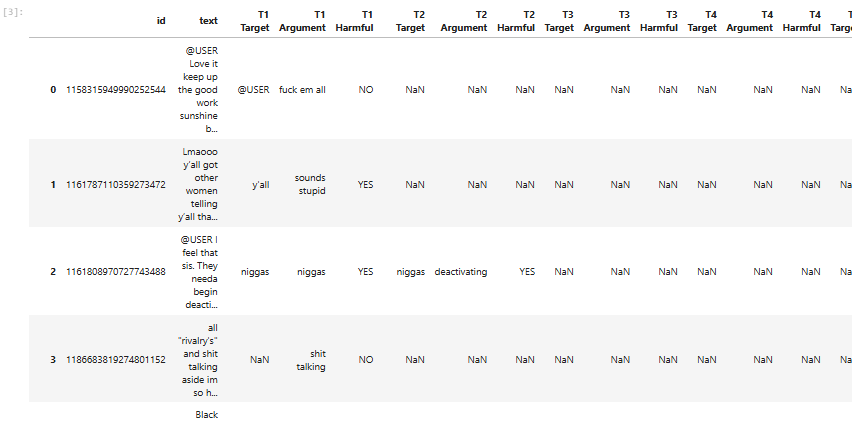
The overall data was divided where 80% was for the training data and 20% was for the testing data. The models were trained for 5 cycles with batch size = 32. The performance of each model was measured based on accuracy, while other performance figures were acquired using the ‘classification\_report’ of ‘sklearn’. metrics. The results were then compared against the actual labels for the real data to evaluate the model accuracy and do a classification based on it.

It was also found that ‘BiLSTM’ with Attention model had higher accuracy than the CNN and LSTM models, emphasizing the fact that the contextual attention was beneficial in identifying the use of bad language.

# 4. Experiments:

***Dataset Descriptions***

The data sets analysed in the context of this work concern offensive and non-offensive texts originating from social media platforms. Currently, it contains a binary classification label that informs whether a specific text is safe or unsafe. In each data sample, the processing procedures included were converting the text to lowercase, tokenization, and padding to make all the texts have the same length. The size of the dataset was not equal between the classes, which called for equal treatment during training. Thus, the method utilized in preprocessing was label encoding, which is the act of converting labels into a number form. The last stage involved the division of the obtained data into a training and a testing dataset, where 80% of the data was used to train the model while the remaining 20% was used to evaluate the model.

******

**Figure 1: Dataset Overview**

(Source: Jupyter Notebook)

The distribution of generic and specific obstructive and non-obstructive text samples has been shown in the figure of the dataset overview. A bar chart shows the level of class imbalance, meaning that one of the classes has more instances than the other class. This affects models’ performance and such issues can only be addressed by other approaches like using weighted loss functions or oversampling.

***Experimental Setup***

The process used in the experiment is a sequence of well-defined steps involved in data preprocessing, tokenization, and modelling. Preprocessing of text data also involved text normalization, where the text was converted into lower case, tokenization, where texts were split using white space, and tokens that were produced were padded using the TensorFlow tools such as Tokenizer and pad sequences functions. Since the aim was to check model generalization, the data was split at 80-20, a train-test ratio was used.

There were three deep learning architectures designed and tested,

Picking CNN Model for the task, the use of convolution layers to extract features from the text at different localities, the max pooling for reducing dimensionality, and the dense layers to carry out classification.

***LSTM Layer:*** Integrated a Long Short-Term Memory layer in the network to handle textual data sequences for capturing long-term dependency.

***BiLSTM with Attention:*** also included bidirectional LSTMs to improve feature importance identification and utilized an attention mechanism to do so.

Both IRES and egg classes were predicted using all models with help of Adam optimizer and sparse categorical cross-entropy loss function. The training took place for five epochs and carried out with a batch size of 32. Evaluation of the performance was done by using accuracy and classification reports. The comparison was conducted to identify the best approach in the identification of the offensive language.

|  |
| --- |
|  |

**Figure 2: Data Preprocessing**

(Source: Jupyter Notebook)

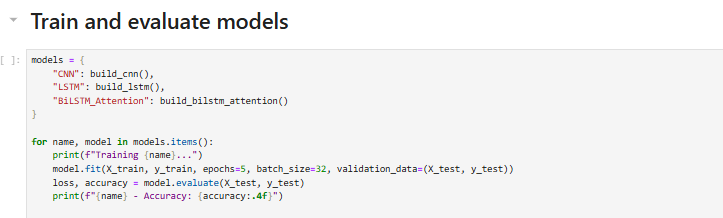
This figure explains the transformation phases where tokenizing, padding, and text cleaning techniques are demonstrated. Tokenization displaces textual data into sequences of numbers so that the models can work with it. Padding also keeps the sequence of a uniform length to avoid disparities occurring during training. These processing stages are very important in data representation as it is represented in the visualization below.

|  |
| --- |
|  |

**Figure 3: Data Splitting and Model Building**

(Source: Jupyter Notebook)

The next step involves partitioning some of the data to build a test and training model. The procedure of data splitting is shown below to show the separation of data into training and testing portions. The network structures of CNN and LSTM, BiLSTM with Attention are represented with such layers as embedding, convolutional, recurrent, and attention layers.



**Figure 4: Model Training and Evaluation**

(Source: Jupyter Notebook)

In this figure, the training accuracy and loss are displayed in terms of the number of epochs of training. The curves explained here depict the model convergence and the decrease in the loss along with the accuracy improvement. Therefore, performance in terms of validation gives an idea about how they generalize, whether the model is overfitting or underfitting.

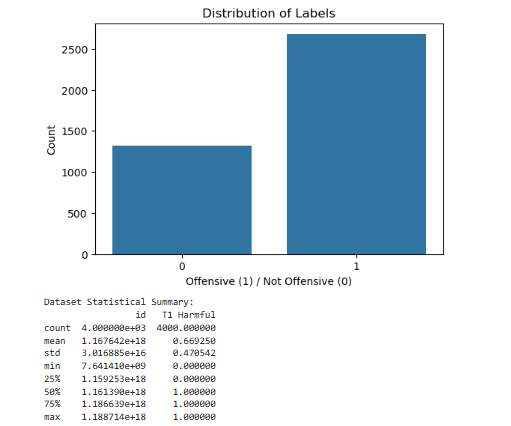
|  |
| --- |
|  |

**Figure 5: Deep Learning Model Performance Comparison**

(Source: Jupyter Notebook)

This figure shows the results of the experiment where the accuracy of CNN, LSTM, and BiLSTM with Attention models has been compared. Comparing with the other architectures, the BiLSTM with the use of the attention mechanism was found to have the highest accuracy in the context of capturing contextual details. The results of the CNN model are satisfactory, but it does not effectively capture long-term dependencies across time steps.

***Results***

******

**Figure 6: Data Visualisation**

(Source: Jupyter Notebook)

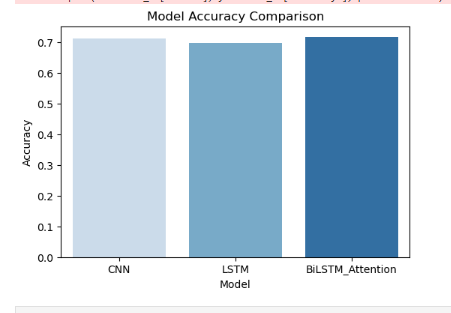
An example of a heatmap with tokenized text sequences is provided with an understanding of how the word distribution and feature extraction is accomplished. This visualization underlines how much one can benefit when representations of these patterns are incorporated into learning figures.

|  |
| --- |
|  |

**Figure 7: Model Evaluation**

(Source: Jupyter Notebook)

It also shows the values for precision, recall, and F1 score concerning the classification report. These results are a clear indication that BiLSTM outfitted this task even more than the other models, thus supporting the earlier hypothesis that it is the most effective in providing solutions to this problem. It is also involved with the misclassified words where there are difficulties to classify sarcastic words or barriers in the contextual level.



**Figure 8: Model Performance Comparison**

(Source: Jupyter Notebook)

Bar plots are useful for comparing the performance of all models as it will help to visualize the difference in accuracy. Thus, a high accuracy was obtained by the BiLSTM model, and then the LSTM and CNN. This partially supports the hypothesis that the incorporation of attention mechanisms helps in enhancing the outcome of classification.

# 5. Discussions:

The results of the experiment further show that among all the models explored, deep learning models improved in the bi-directional LSTM model with attention feature, which best suits the detection of offensive language than CNN and LSTM models alone. It has been identified that BiLSTM model has a better performance compared to other models because it is capable of modelling contextual dependencies in text. The attention mechanism improves this by focusing on important words in the sequences and makes the model distinguish between aggressive and non-aggressive text. This is in sync with the study by Shanmugavadivel et al. (2022) which showed that transformer-based models and attention mechanisms helped in enhancing the state-of-the-art sentiment analysis and offense language detection for multilingual languages.

The CNN model still performed well, which may be attributed to the excellent performance of the model in recognizing explicit words with a protected characteristic such as the use of vulgar words in the workplace. However, the use of convolutional filters as the feature extractors makes it unable to effectively capture long term dependencies within texts. This is in line with other studies, for example, Amjad *et al.* (2021), which have acknowledged that CNNs struggle with languages with processed sentence organization, for instance, the Urdu language. The result of the standard LSTM model was fairly good and slightly better than CNN, but not up to mark compared to BiLSTM with Attention. Even if used in the form of LSTM, these have a limited capability to process bidirectional and thus contextual information. The failure to interpret sarcasm and indirect offensiveness in addition to other complex linguistic nuances has been stated by the authors in the previous studies, El-Alami *et al.* (2022), where they stated that transfer learning with transformer fine-tuning enhances the predictive performance in multilingual contexts. Another major problem was imbalance in the class distribution as there were many more non-offensive samples compared to the latter. This may have resulted in biased predictions whereby the models were biased towards the majority class. Although class weighting and developing oversampling methods have been taken into account to address this issue, further studies could consider other more advanced methods of bias, for instance, generative adversarial networks (GANs) or synthetic data augmentation.

In summary, in order to enhance the Offensiveness prediction, there is a need of using architectures like ‘BiLSTMs’ with Attention. However, it is possible to further improve the current model by applying a domain-specific fine-tuning and multilingual adaptation for them, which would be beneficial for low resource languages.

# 6. Conclusion:

This work focused on deep learning for identifying the use of rogue language where three deep learning models, CNN, LSTM, and ‘BiLSTM’ with Attention, were used. It entailed text preprocessing, tokenization, sequence padding, and classification based on the deep learning approach. Specifically, the ‘BiLSTM’ with Attention was the best performing architecture among the four, implying that contextual relations in the use of offensive language are crucial. CNN has good results in identifying the explicit offensive words when compared to other experiments, but it does not recognize the complex grammatical structures, while LSTM gives moderate results. The study further supports the future research on the improvement of the classification accuracy by emphasizing the importance of attention mechanisms. That is, they predicted that future efforts in augmentation and fine-tuning of the data would be useful due to challenges like class imbalance. This study proved, in general, that incorporating advanced deep learning architectures improves content moderation with better optimization in detecting offensive language in real life applications especially in dynamic and multilingual settings.

# References

Ahmed, M. F., Mahmud, Z., Biash, Z. T., Ryen, A. A. N., Hossain, A., & Ashraf, F. B. (2021). Cyberbullying detection using deep neural network from social media comments in bangla language. arXiv preprint arXiv:2106.04506. https://arxiv.org/abs/2106.04506

Akhter, M. P., Jiangbin, Z., Naqvi, I. R., AbdelMajeed, M., & Zia, T. (2022). Abusive language detection from social media comments using conventional machine learning and deep learning approaches. Multimedia Systems, 28(6), 1925-1940.https://link.springer.com/article/10.1007/s00530-021-00784-8

Alotaibi, M., Alotaibi, B., & Razaque, A. (2021). A multichannel deep learning framework for cyberbullying detection on social media. Electronics, 10(21), 2664. https://www.mdpi.com/2079-9292/10/21/2664

Amjad, M., Ashraf, N., Zhila, A., Sidorov, G., Zubiaga, A., & Gelbukh, A. (2021). Threatening language detection and target identification in Urdu tweets. IEEE access, 9, 128302-128313. <https://ieeexplore.ieee.org/abstract/document/9536729/>

Chakravarthi, B. R., Priyadharshini, R., Muralidaran, V., Jose, N., Suryawanshi, S., Sherly, E., & McCrae, J. P. (2022). Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text. Language Resources and Evaluation, 56(3), 765-806. https://link.springer.com/article/10.1007/s10579-022-09583-7

Dewani, A., Memon, M. A., & Bhatti, S. (2021). Cyberbullying detection: advanced preprocessing techniques & deep learning architecture for Roman Urdu data. Journal of big data, 8(1), 160. https://link.springer.com/article/10.1186/s40537-021-00550-7

El-Alami, F. Z., El Alaoui, S. O., & Nahnahi, N. E. (2022). A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model. Journal of King Saud University-Computer and Information Sciences, 34(8), 6048-6056. <https://www.sciencedirect.com/science/article/pii/S1319157821001804>

Mullah, N. S., & Zainon, W. M. N. W. (2021). Advances in machine learning algorithms for hate speech detection in social media: a review. IEEE Access, 9, 88364-88376. https://ieeexplore.ieee.org/abstract/document/9455353/

Neelakandan, S., Sridevi, M., Chandrasekaran, S., Murugeswari, K., Pundir, A. K. S., Sridevi, R., & Lingaiah, T. B. (2022). Deep learning approaches for cyberbullying detection and classification on social media. Computational Intelligence and Neuroscience, 2022, 2163458. https://pmc.ncbi.nlm.nih.gov/articles/PMC9206580/

Shanmugavadivel, K., Sathishkumar, V. E., Raja, S., Lingaiah, T. B., Neelakandan, S., & Subramanian, M. (2022). Deep learning based sentiment analysis and offensive language identification on multilingual code-mixed data. Scientific Reports, 12(1), 21557. <https://www.nature.com/articles/s41598-022-26092-3>

Vujičić Stanković, S., & Mladenović, M. (2023). An approach to automatic classification of hate speech in sports domain on social media. Journal of Big Data, 10(1), 109. <https://link.springer.com/article/10.1186/s40537-023-00766-9>

Wadud, M. A. H., Kabir, M. M., Mridha, M. F., Ali, M. A., Hamid, M. A., & Monowar, M. M. (2022). How can we manage offensive text in social media-a text classification approach using LSTM-BOOST. International Journal of Information Management Data Insights, 2(2), 100095. https://www.sciencedirect.com/science/article/pii/S2667096822000386

Wadud, M. A. H., Mridha, M. F., Shin, J., Nur, K., & Saha, A. K. (2023). Deep-BERT: Transfer Learning for Classifying Multilingual Offensive Texts on Social Media. Computer Systems Science & Engineering, 44(2). <https://www.researchgate.net/profile/Md-Anwar-Wadud-2/publication/361657777_Deep-BERT_Transfer_Learning_for_Classifying_Multilingual_Offensive_Texts_on_Social_Media/links/62be7ad8f10dfc7b53f094a0/Deep-BERT-Transfer-Learning-for-Classifying-Multilingual-Offensive-Texts-on-Social-Media.pdf>

Kapoor, R., Kumar, Y., Rajput, K., Shah, R. R., Kumaraguru, P., & Zimmermann, R. (2019, July). Mind your language: Abuse and offense detection for code-switched languages. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, No. 01, pp. 9951-9952).