Tiako P.F. (ed) *Competitive Tools, Techniques, and Methods*. Chronicle of Computing. OkIP. CAIS24#12
© 2024 Oklahoma International Publishing https://doi.org/10.55432/978-1-6692-0007-9_3

Text Toxicity Level Detection using Deep Contextualized Embedding models

Omar Elgendy[a], Ali Bou Nassif[a], Bassel Soudan[a]

[a]*University of Sharjah, Sharjah, United Arab Emirates*

ABSTRACT

Toxic text is a critical aspect of social media, particularly in today's digital landscape. With the spread of online communication, it has become increasingly easy for individuals to spread harmful or offensive content. Toxic texts include the spread of misinformation, the promotion of hate speech, bullying, and the erosion of trust in online communities. Text toxicity detection algorithms can help to identify and mitigate these negative effects by automatically flagging potentially harmful content. This allows social media platforms to intervene and take appropriate action, such as removing the content or warning the user. Usually, social media platforms offer a reporting strategy which acts after a human decision is made. However, social media now requires an automated system to do this task. In this work, we proposed a Deep learning Regression model to predict the toxicity level in text. Additionally, we fine-tuned multiple Bert models for this task. Our work was evaluated using Mean Square Error, Root Mean Square Error and Mean Absolute Error compared to the testing set of the data and we got for the base model MSE of 0.562, RMSE of 0.750 and MAE of 0.364 but for BERT we got MSE 0.403, RMSE 0.635 and MAE 0.232.

Keywords: Toxic Text, Deep Learning, Bert, Natural Language Processing

I. INTRODUCTION

Spam and toxic text messages impact our society in a multitude of detrimental ways. They congest our communication lines with undesired and frequently false information, making it challenging for individuals to locate the information they require. This consequence of misleading content can lead to confusion and dissatisfaction, disrupting the flow of accurate information and undermining trust in online communication channels. As a result, interpersonal relationships may suffer, and trust within communities can be eroded.

Those who are exposed to these messages may experience significant emotional distress, including feelings of anxiety, tension, and even sadness. The continuous exposure to harmful and deceptive content can create a pervasive sense of unease and mistrust, further exacerbating mental health issues. Additionally, the disinformation that can be conveyed by these texts might result in the propagation of damaging ideas and beliefs (Garlapati, Malisetty, & Narayanan, 2022). This spread of false information can have far-reaching consequences, influencing public opinion and behavior in harmful ways. Additionally adding to the general contamination of our digital surroundings are texts that are toxic or spam. They frequently require massive volumes of data to be transmitted and stored, which can put a burden on our digital infrastructure. There is a negative environmental impact as a result of the increased demand on servers and data centers, which raises energy consumption and resource exploitation. Moreover, the false information propagated by these malicious writings could have a significant impact on society as a whole. By spreading negative beliefs and attitudes, they can aid in the division of society and the radicalization of particular groups. This polarization has the potential to widen rifts in society, making it more challenging to come to an agreement on crucial matters and creating an atmosphere of animosity and discord.

In the worst-case scenarios, these spam messages might even promote unlawful actions like fraud, scamming, and cyberbullying or inspire violence. The anonymity and reach of the internet allow evil actors to more readily prey on the vulnerable and carry out harmful activities. Text poisoning can be used to more easily identify and remove comments that are threatening, offensive, or detrimental in any other way. By proactively recognizing and minimizing the impact of dangerous content, online platforms can prevent it from spreading and protect users from potential harm by implementing advanced toxicity detection algorithms. Online platforms may ensure that users can communicate in a friendly and polite atmosphere by identifying and flagging toxic comments. This not only fosters a more positive and inclusive online environment but also helps maintain the integrity of the platform. Determining harmful comments can also assist in shielding vulnerable individuals from online abuse and harassment (Zaheri, Leath, & Stroud, 2020). For instance, automated detection systems can alert moderators to potentially abusive content, allowing for timely intervention and support for affected users.

Furthermore, putting automatic text toxicity detection tools in place can help human moderators handle some of the workload. Conventional content moderation techniques can be labor- and time-intensive because they mostly rely on user reporting and manual evaluation. Platforms are able to improve speed and efficacy in the moderating process by using advanced machine learning models and natural language processing techniques to streamline the process.

Both the wellbeing of the individual and the cohesion of society are seriously threatened by the spread of spam and harmful text messages. A holistic strategy that incorporates

Presented at the 2024 OkIP International Conference
on Automated and Intelligent Systems (CAIS)
in Oklahoma City, OK, USA, and Online, on October 3, 2024

community involvement, education, and technology innovation is needed to address this issue. We may endeavor to create a more secure and encouraging digital environment for all by raising awareness of the risks connected to toxic messages and encouraging an attitude of decency and accountability online.

Natural language processing (NLP) is an artificial intelligence, computer science, and linguistics discipline that focuses on how computers and human languages interact. To help computers communicate and comprehend human language in a natural and intuitive way, it entails creating models and algorithms that can process, analyze, and produce human language. The potential of natural language processing (NLP) to revolutionize how we engage with technology and facilitate information access and machine communication is what makes NLP so important. NLP applications help organizations process vast amounts of text data to make better decisions by, among other things, increasing the accuracy of language translation and speech recognition Information retrieval, sentiment analysis, and text summarization are just a few of the many tasks that fall under NLP's purview. The efficacy and efficiency with which we interact with digital content is improved by these tasks. While fake news detection aids in identifying and halting the spread of false information, emotion detection from text can be utilized in customer care to better understand and respond to consumer moods. Lately, the capabilities of natural language processing have been greatly enhanced with the introduction of deep learning models, especially Transformers. Since their introduction in 2017, transformers have completely changed the field by enabling more precise and contextual language processing (Nassif, Elnagar, Elgendy, & Afadar, 2022). These models are excellent at several NLP tasks, including language modeling, text production, and machine translation.

The BERT model, which stands for Bidirectional Encoder Representations from Transformers, is among the most prominent models in this area. For particular NLP applications, BERT was intended to pretrain deep bidirectional representations from unlabeled text and then fine-tune using labeled data. With the help of this method, BERT is able to comprehend a word's context by looking at the words that surround it, which results in more precise and complex language processing (Devlin, Chang, Lee, & Google, 2021).

NLP is an important and quickly expanding field that improves human capacity to communicate with technology. Its extensive potential is demonstrated by its applications in voice recognition, language translation, emotion detection, and fake news detection. NLP has advanced even further since the introduction of models like Transformers and BERT, which have made it a vital tool in the fields of machine learning and artificial intelligence.

In this study, we utilize Jigsaw Regression Based Data to train two deep learning models, a basic Deep Neural Network, and a Fine-tuned Bert base model, for text toxicity detection. Through a comparative analysis of their performance, measured by the mean absolute error and root mean squared error, we aim to identify the most efficient model for this crucial regression task.

II. RELATED WORK

In this section, some of the previous work that was done for the same application will be discussed here. Since not too much work was done in toxic texts. The compared work is for both toxic texts and Cyber bullying.

In this paper (Maslej-Krešňáková, Sarnovský, & Jacková, 2022), the authors explore the use of data augmentation techniques in the detection of antisocial behavior on online platforms. Data augmentation is used to overcome issues related to the lack of data or class imbalance, by generating artificial data samples to improve the volume of the training set or balance the target distribution. The authors focus on the class imbalance problem and apply data augmentation techniques (EDA) to two problems: fake news and toxic comments classification. EDA techniques are found to be taskdependent, with certain limitations resulting from the data they are applied on. The authors train convolutional neural networks classifiers and compare their performance on the original and EDA-extended datasets. EDA techniques prove to be useful for the fake news dataset, boosting the F1 score by 0.1, but are not suitable for the toxic comment's dataset, improving performance only marginally.

Authors in (Georgakopoulos, Tasoulis, Vrahatis, & Plagianakos, 2018), presented a method for identifying toxic online comments using convolutional neural networks (CNNs). The authors compared the performance of CNNs to the traditional bag-of-words approach for text analysis, using a selection of algorithms known to be effective in text classification. The results showed that CNNs with word embeddings (CNNfi) improved toxic comment classification, outperforming the other methods in terms of accuracy and specificity, while having a relatively low false discovery rate (FDR). These results suggest that CNNfi is a promising approach for this task. The mean values for accuracy, specificity, and false discovery rate (FDR) are 0.912, 0.917, and 0.083, respectively, for the convolutional neural networks (CNNs) with word embeddings (CNNfi) approach. For the CNNs with random word embeddings (CNNrand) approach, the mean values are 0.895, 0.906, and 0.092.

The authors in (Chakrabarty, 2020) presented a machine learning model to detect toxic comments in online environments. The model was trained on a dataset of labeled comments, and its performance was evaluated using two metrics: mean validation accuracy and absolute validation accuracy. Mean validation accuracy is the average accuracy of the model across all classes or types of toxicity. In other words, it measures the overall performance of the model in correctly identifying toxic comments from non-toxic comments. Absolute validation accuracy, on the other hand, measures the performance of the model for each individual class or type of toxicity. This metric provides a more detailed view of the model's performance, allowing for the identification of any potential weaknesses or areas for improvement. The authors report a mean validation accuracy of 98.08% and an absolute validation accuracy of 91.64%. This suggests that the model can identify toxic comments with a high degree of accuracy, although there may be some differences in performance for different types of toxicity.

In (Mounir et al., 2019), authors proposes a supervised machine learning approach for detecting and preventing cyberbullying. The approach uses several classifiers, including a neural network, to train and recognize bullying actions. Evaluation of the proposed approach on a cyberbullying dataset showed that the neural network performed better, achieving an accuracy of 92.8%, compared to 90.3% for a support vector machine. The neural network also outperformed other classifiers in similar work on the same dataset.

This paper (Dalvi, Chavan, & Halbe, 2020), presents a machine learning model for detecting and preventing cyberbullying on Twitter. The authors use two classifiers, Support Vector Machine (SVM) and Naïve Bayes, to train and test the model. They find that both classifiers can detect true positives with reasonable accuracy (71.25% for Naïve Bayes and 52.70% for SVM), with SVM outperforming Naive Bayes on the same dataset. The authors also use the Twitter API to fetch tweets and pass them to the model for detecting bullying. Limitations of the study include the lack of a comprehensive dataset and the limited scope of the study, which only focuses on detecting bullying on Twitter.

The study in (Nalini & Sheela, 2015) proposes a powerful technique for identifying cyberbullying on Twitter that combines text mining, latent feature extraction via Latent Dirichlet Allocation (LDA), and weighted features with a negative word lexicon. The authors provide a statistical detection strategy that combines a text classification algorithm with a graph model to identify active cyberbullies and victims. Along with relevant findings, the report also offers a thorough description of Twitter and its function in cyberbullying. According to the study, the proposed method performed better than the baseline method on the Twitter dataset, with precision, recall, and F-1 measures of 0.87, 0.96, and 0.97, respectively. It ends with recommendations for enhancing the identification of cyberbullying by lowering false positives and negatives.

Authors in (Nurrahmi & Nurjanah, 2018) offers a technique that uses text categorization and user trustworthiness analysis to find cyberbullying on Indonesian Twitter. The technique, which combined SVM and KNN to detect cyberbullying in text, outperformed KNN with an F1-score of 67% for SVM. To extract features from tweets, eight rules were developed. These rules helped identify 257 "Normal Users," 45 "Harmful Bullying Actors," 53 "Bullying Actors," and 6 "Prospective Bullying Actors." The method was successful in identifying cyberbullying actors and harmful tweets despite some issues with casual language and POS Taggers and the need for more thorough feature extraction. Future advancements will focus on feature extraction refinement and the use of an improved Indonesian POS Tagger tool. The study offers a cutting-edge strategy for stopping cyberbullying, especially in the setting of Indonesia.

The work in (Shah, Vidyavihar, Chopdekar, & Somaiya, 2020) focuses on using machine learning approaches for Twitter cyberbullying detection, a severe issue that has a global impact on mental health problems. The manuscript investigates various machine learning models to accurately categorize tweets using a dataset made up equally of bully and non-bully tweets. The most successful classifier was determined to be logistic regression, which had 91% precision, 94% recall, and a 93% F1-score. Natural Language Processing methods, such as lemmatization and Term Frequency-Inverse Document Frequency (TF-IDF), were used to clean and analyze the data. The long-term objective is to put in place a system that automatically finds and removes bullying-related information, shielding users from negative encounters. India is ranked third in the world for cyberbullying, highlighting the need for this study. The study finds that machine learning is an effective method for combating cyberbullying.

The authors in (Di Capua, Di Nardo, & Petrosino, 2016) suggests a non-supervised method for automatically detecting cyberbullying on social media sites like Twitter, YouTube, and Formspring. This method makes use of machine learning and natural language processing. The methodology is based on a Growing Hierarchical Self-Organizing Map (GHSOM), which identifies the semantic and communicational behavior of suspected cyberbullies. The model makes use of a variety of aspects, such as syntactic, semantic, sentimental, and social network-based elements. On the datasets, the authors used K-fold cross-validation, and they compared the outcomes to supervised learning models. With a precision of 0.60, accuracy of 0.69, recall of 0.94, and F1 score of 0.74 on YouTube data, the system performed admirably. Additionally, a genuine Twitter data stream was tested qualitatively. To increase the effectiveness of the system, the article emphasizes the need for additional developments in areas like sarcasm detection. This innovative strategy might serve as a foundation for creating practical monitoring programs to stop cyberbullying.

In this research, we built a basic deep learning model and utilized BERT for text toxicity level detection as a regression deep learning program. We used an opensource dataset and evaluated our model using MSE, RMSE and MAE. Our work outcome is a toxicity level instead of categorical output.

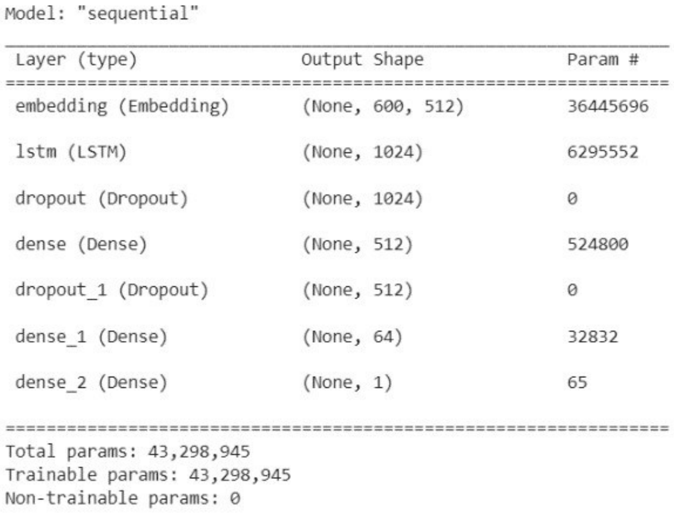
III. METHODOLOGY

A. *Dataset*

The dataset used in this work is Jigsaw Regression Based Data1 available publicly in Kaggle. The dataset is 3GB and it includes three CSV files with pre-processed and balanced text data and float data on toxicity. It also includes word embedding files in 100D and 256D. These files were created using the Continuous bag of words (CBOW) approach with genism in Python. The dataset output is a regression output which means a number indicates how toxic is the text. Here is a quick summary of some basic statistical data. Data split into 80% training and 20% testing. During the training process, we proposed two different models for achieving this task. The first model was a basic Deep Neural Network and the second was Fine-tuned Bert base model. We will be showing the architecture of each one.

B. *Base model:*

This deep learning model is a sequential model with six layers. The first layer is an embedding layer. The second layer is an LSTM layer. The third and fifth layers are dropout layers. The fourth layer is a dense layer. The sixth layer is a dense layer. The final layer is a dense layer with one output unit since it's a regression problem.

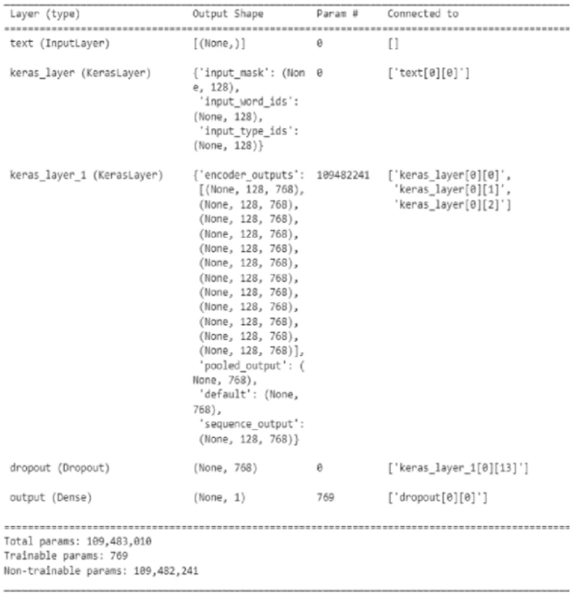


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 600, 512)          36445696

 lstm (LSTM)                 (None, 1024)              6295552

 dropout (Dropout)           (None, 1024)              0

 dense (Dense)               (None, 512)               524800

 dropout_1 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 64)                32832

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 43,298,945
Trainable params: 43,298,945
Non-trainable params: 0
```

Table 1: Base model architecture

C. *Bert model:*

This is a deep learning model that uses BERT to encode text input and produces a single output. It has an input layer, a preprocessing layer, and an encoder layer, followed by a dropout layer and a dense output layer. The purpose of the dropout layer is to prevent overfitting.



```
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 text (InputLayer)              [(None,)]            0           []

 keras_layer (KerasLayer)       {'input_mask': (Non  0           ['text[0][0]']
                                e, 128),
                                 'input_word_ids':
                                (None, 128),
                                 'input_type_ids':
                                (None, 128)}

 keras_layer_1 (KerasLayer)     {'encoder_outputs':  109482241   ['keras_layer[0][0]',
                                 [(None, 128, 768),               'keras_layer[0][1]',
                                 (None, 128, 768),                'keras_layer[0][2]']
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768),
                                 (None, 128, 768)],
                                 'pooled_output': (
                                None, 768),
                                 'default': (None,
                                768),
                                 'sequence_output':
                                 (None, 128, 768)}

 dropout (Dropout)              (None, 768)          0           ['keras_layer_1[0][13]']

 output (Dense)                 (None, 1)            769         ['dropout[0][0]']

==================================================================================================
Total params: 109,483,010
Trainable params: 769
Non-trainable params: 109,482,241
__________________________________________________________________________________________________
```

Table 2: Bert model architecture

C. *Training*

The training process uses Adam optimizer with a learning rate of 0.001. The model is trained for 25 epochs. with a batch size of 64. It uses the root mean squared error (RMSE) as the loss function and evaluates the model using the mean absolute error (MAE), mean squared error (MSE), and RMSE metrics.

D. *Results*

The two deep learning models were evaluated using three primary metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). Each metric provides a unique perspective on the models' performance.

Mean Squared Error (MSE): MSE is calculated as the average of the squared differences between the predicted and actual values. It's a commonly used regression loss function, and the lower the MSE, the better the model. In mathematical terms, for n observations, below is the MSE equation:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Equation 1: Mean Square Error

In Equation 1, Yi represents the actual values and Ŷi represents the predicted values. In the results, the Base model had an MSE of 0.562 while the BERT model achieved an MSE of 0.403, demonstrating better accuracy.

[1] https://www.kaggle.com/datasets/nkitgupta/jigsawregression-based-data

Root Mean Squared Error (RMSE): RMSE is simply the square root of the MSE. It is especially useful because it is in the same units as the response variable and hence provides an interpretable measure of error

$$\text{RMSE} = \sqrt{MSE}$$

Equation 2: Root Mean Square Error

The RMSE of the Base model was 0.750 while the BERT model achieved an RMSE of 0.635. Again, the BERT model outperformed, indicating it made smaller errors on average.

Mean Absolute Error (MAE): The MAE calculates the average of the absolute differences between the predicted and actual values. It is less sensitive to outliers compared to the MSE and provides a linear measure of errors. Below is the equation of MAE.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} | y_i - \hat{y}_i |$$

Equation 3: Mean Absolute Error

In this case, the Base model had an MAE of 0.364, while the BERT model achieved an MAE of 0.232. In all three metrics, the BERT model achieved lower error rates than the Base model, indicating that it is more accurate in detecting text toxicity. Although both models performed well, the BERT model was demonstrably superior in terms of these evaluation metrics. In the table below we summarize all our results.

Table 3: Results comparison

Model	MSE	RMSE	MAE
Base	0.562	0.750	0.364
BERT	0.403	0.635	0.232

IV. DISCUSSION AND CONCLUSION

Bidirectional Encoder Representations from Transformers, often known as BERT, is renowned for its exceptional competence in a variety of natural language processing applications. It makes use of context from both the text before and after, which helps it understand linguistic nuance better. BERT performed incredibly well in the context of this study, not only in traditional classification tasks but also in a regression problem as text toxicity detection. This demonstrates BERT's adaptability to various problem architectures, in this case a regression challenge, and highlights its versatility as a potent tool in the field of deep learning and machine learning.

A basic Deep Neural Network model and the well-known BERT model were both used as separate deep learning models for the task at hand. The goal was to identify text toxicity, a problem that plagues a lot of our digital conversations. By engaging in this project, we hoped to help create a more secure and civil online community. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), two often used measures for regression issues, were employed to assess the performance of these models. A lower value reflects better performance since these metrics measure the typical inaccuracy in the model predictions. Our BERT model excelled in this task, achieving an MAE of 0.232 and an RMSE of 0.635. These values signify that, on average, the model's predictions were close to the actual toxicity levels. This performance showcases the BERT model's capacity for sophisticated natural language understanding tasks and its ability to accurately predict the toxicity level of a given text.

In conclusion, the superior performance of BERT in this work further cements its standing as a leading solution for natural language processing tasks, including those involving regression problems like text toxicity detection.

REFERENCES

Garlapati, A., Malisetty, N., & Narayanan, G. (2022). Classification of Toxicity in Comments using NLP and LSTM. 8th International Conference on Advanced Computing and Communication Systems (ICACCS), 16–21. https://doi.org/10.1109/ICACCS54159.2022.9785067

Zaheri, S., Leath, J., & Stroud, D. (2020). Toxic Comment Classification. SMU Data Science Review, 3(1). Retrieved from https://scholar.smu.edu/datasciencereview/vol3/iss1/13

Nassif, A. B., Elnagar, A., Elgendy, O., & Afadar, Y. (2022). Arabic fake news detection based on deep contextualized embedding models. Neural Computing and Applications, 34(18), 16019–16032. https://doi.org/10.1007/s00521-022-07206-4

Devlin, J., Chang, M.-W., Lee, K., Google, & AI Language. (2021). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Retrieved from https://github.com/tensorflow/tensor2tensor

Maslej-Krešňáková, V., Sarnovský, M., & Jacková, J. (2022). Use of Data Augmentation Techniques in Detection of Antisocial Behavior Using Deep Learning Methods. Future Internet, 14(9). https://doi.org/10.3390/fi14090260

Georgakopoulos, S. V., Tasoulis, S. K., Vrahatis, A. G., & Plagianakos, V. P. (2018). Convolutional Neural Networks for Toxic Comment Classification. Proceedings of the 10th Hellenic Conference on Artificial Intelligence, 1–6. https://doi.org/10.1145/3200947.3208069

Chakrabarty, N. (2020). A Machine Learning Approach to Comment Toxicity Classification. Advances in Intelligent Systems and Computing, 999, 183–193. https://doi.org/10.1007/978-981-13-9042-5_16

Mounir, J. H., et al. (2019). Social Media Cyberbullying Detection using Machine Learning. International Journal of Advanced Computer Science and Applications, 10(5). https://doi.org/10.14569/IJACSA.2019.0100587

Dalvi, R. R., Chavan, S. B., & Halbe, A. (2020). Detecting A Twitter Cyberbullying Using Machine Learning. 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 297–301. https://doi.org/10.1109/ICICCS48265.2020.9120893

Nalini, K., & Sheela, L. J. (2015). Classification of tweets using text classifier to detect cyber bullying. Advances in Intelligent Systems and Computing, 338, 637–645. https://doi.org/10.1007/978-3-319-13731-5_69

Nurrahmi, H., & Nurjanah, D. (2018). Indonesian Twitter Cyberbullying Detection using Text Classification and User Credibility. 2018 International Conference on Information and Communication Technology (ICOIACT), 543–548. https://doi.org/10.1109/ICOIACT.2018.8350758

Shah, R., Vidyavihar, S., Chopdekar, R., & Somaiya, S. K. J. (2020). Machine Learning based Approach for Detection of Cyberbullying Tweets. International Journal of Computer Applications, 175(37), 975–8887. https://doi.org/10.5120/ijca2020920946

Di Capua, M., Di Nardo, E., & Petrosino, A. (2016). Unsupervised cyber bullying detection in social networks. Proceedings of the International Conference on Pattern Recognition, 0, 432–437. https://doi.org/10.1109/ICPR.2016.7899672