Cyber Bullying Detection using Machine Learning

Jai Mehta
Jai.mehta@somaiya.edu
K.J.Somaiya College of Engineering

Atharva Kitkaru
atharva.kitkaru@somaiya.edu
K.J.Somaiya College of Engineering

Rachit Mehta
rachit.hm@somaiya.edu
K.J.Somaiya College of Engineering

***Abstract--* Cyber bullying has been increasing at an unfathomable rate with the rise of the internet and the increasing radius of the social networks. This increasing amount of cyber bullying serves to be a threat to many people who are an active part of these social networks, especially the people using the social media websites for posting information and communication. These crimes have affected many people and have also resulted in suicides. Thus we need an efficient system to detect these bullying attacks and reduce their amount. In the past machine learning models have been constructed to address this problem, but those models have not been very successful. In this paper we will discuss the latest technologies like Naive Bayes, SVM, BERT models,CNN and LSTM that have been applied in this area and compare their results.**

Keywords-- Cyber Bullying, Social Network, Machine Learning, Naive Bayes, SVM, CNN

I. Introduction

Networking, in today's world, has become a really important and unavoidable part of our lives. With the help of these networks, we can have social exchange of information and media with anyone and at any point of time as per our convenience. Hence, these networks can also be called social networks. Millions of users have been added to these networks in recent years and its growing at a tremendous rate. At present there are approximately 3 billion social media users. Cyberbullying possesses a constant threat to these social media networks, according to the National Crime Security Council. Cyberbullying can take place on platforms like social media apps, gaming apps, mobile phones etcetera where there involves exchange of information like texts, images and videos which could serve to be embarrassing to other people. Social Media Platforms like Instagram, Facebook, Snapchat and even Youtube can serve to be a platform for cyberbullying and the worst part of this act is that it is really difficult or next to impossible to track the source of these crimes. The criminals often upload undisclosed information about someone to threaten them or for selfish gains. Most of the times, cyberbullying is targeted towards specific individuals that lead to immense embarrassment, depression, anxiety, injures their reputation and even leads to suicide in the worst cases.

As the radius of these networks increases, there needs to be measures to regulate the interaction of humans with strangers to avoid the acts of cyberbullying. Although it is difficult to trace the source of cyberbullying, many social media sites have tried to include methods to bring these acts under regulation. For example social media site Instagram has the option to report a posted image, video or information to the authorities. Once the image is reported it undergoes inspections and is taken down from the website if there is any violation found. Few sites have also included the method to permit exchange of information and media only between people we follow or we are friends with.

In the presence of cyberbullying, we need an efficient cyberbullying detection system, which can distinguish and detect the bullying media from other media, thereby reducing the number of cyber bullying cases. These systems need to be personalized that can have permissions to personal texts and media, for extraction of meaningful data to classify them as bullying or non-bullying. To create such efficient systems, artificial intelligence, machine learning and deep learning models are used. There is an imbalance in the data on which the models have to be trained, since there is a high amount of non-bullying data compared to the bullying data present. Also, the models should be trained on all kinds of data text, image, video etc... Hence, initially the data is analysed and the imbalance is worked upon and further explored, later the models are trained on the well explored datasets and finally the models are fine tuned to achieve high accuracy of detection. In the paper, we will discuss different approaches for identifying cyberbullying by utilizing ML techniques like Naive Bayes, SVM, BERT models, CNN and LSTM.

II. Related Work

There have been several trials on building this system in the past using the Machine Learning based approach. Natural Language Processing was used, that included N-Grams and Term Frequency- Inverse Document Frequency to extract features from the media present on the social networks, which are trained using classifiers like Support Vector Machines or Naïve Bayes. Besides this approach, there was a Lexicon approach too, that used to identify the sentiments behind the text. But, the machine learning approach was preferred over the Lexicon approach.

Later on, the scientists chose Neural Networks (Convolutional and Recurrent) methods and deep learning in modelling of the language. Long Short-Term Memory and Convolutional Neural Network models were constructed to deal with the issues in cyberbullying detection. Few methods were also designed on the bases of personal data like the followers and friends of a person for bullying detection. These methods achieved an overall accuracy of approximately 86%. In deep learning with the increase in the number of layers a problem of vanishing gradient arises. Research has revealed that when these results are combined with gradient boosted decision trees, the accuracy will increase considerably.

There are other areas of research in cyberbullying too, like the detection of Rumours on Social Media Platforms. Researchers have been able to design algorithm that would detect rumours on the basis of four major factors:

- The source of the rumour
- The date of upload
- The location of upload
- The original place of content

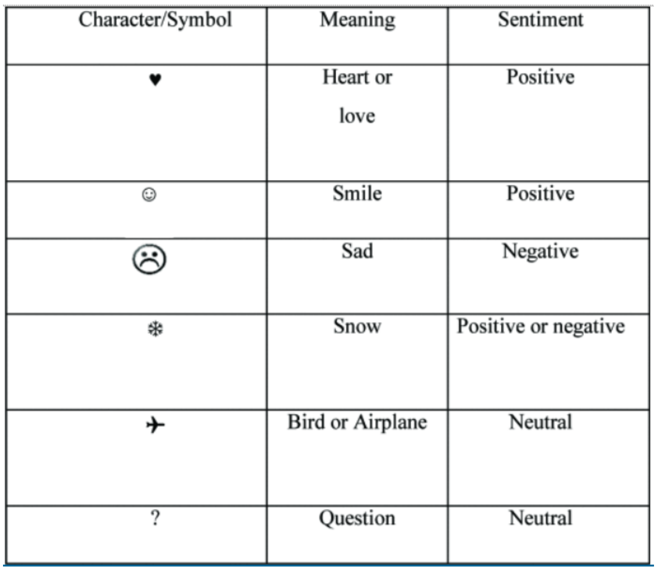
The very famous example supporting this research would be the implementation of rumour detection in the social media app Whatsapp, which detects the messages that may be rumours and are spreading quickly through the network. These messages are highlighted with a tag that they may be fake and have been forwarded a couple of times. Even the application Instagram has a team that detects the presence of fake information and takes it down. In-fact Instagram, has a dedicated team to aid this detection of rumours.

III. Methodology

A. Pre-Processing of Data

Twitter has noisy data which consists of emoticons, folksonomies, slangs, censored words. short message texts.It becomes crucial to remove this noise. General steps in preprocessing are as follows:

- Converting uppercase letters to lowercase
- Removal of URLS
- Removal of punctuations and hashtags..
- Appropriate conversion of symbols/emoticons into text.



Character/Symbol	Meaning	Sentiment
♥	Heart or love	Positive
☺	Smile	Positive
☹	Sad	Negative
❄	Snow	Positive or negative
✈	Bird or Airplane	Neutral
?	Question	Neutral

Fig 1: Sample conversion of symbols and their corresponding meanings

- Removal of stop words(and, or, how, but).
- Word stemming into root form i.e. removal of prefixes and suffixes.

Tokenization: Words are transformed into feature vectors using a dictionary of features. The index of the word in the vocabulary is linked to the number of times it occurs in the training dataset.

B. Feature extraction and preparation of Feature vector

Machine learning-based cyberbullying detection has 2steps: Numerical Representation Learning for Tweets and Classification. Each tweet is converted into a fixed length vector. This constitutes the feature vector space.

Word embeddings: Word embeddings are Words represented as numeric vectors. For example, “The capital of California is Sacramento.”, the word “capital” can be encoded as [0 1 0 0 0 0] and Sacramento can be encoded as [0 0 0 0 0 1]. These vectors can be represented in vector space and can be used for learning.

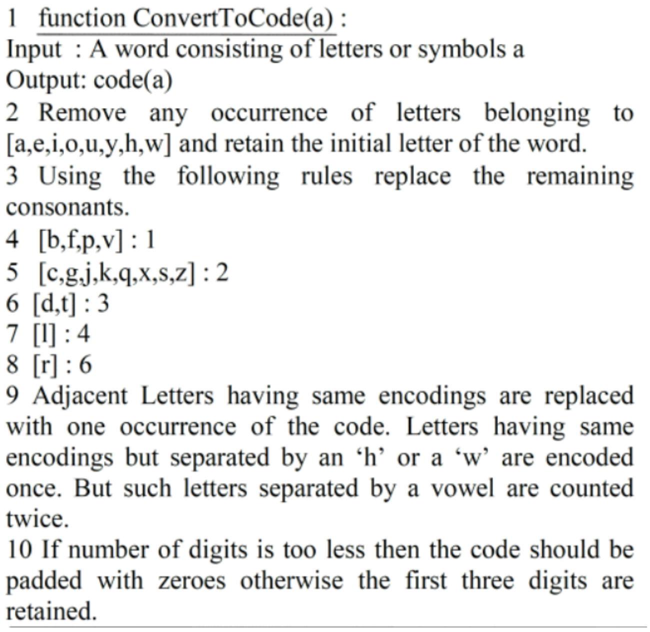
There are two types of word embeddings:

- Frequency based embeddings
- Prediction based embeddings

Frequency based embeddings are deterministic in their approach, they do not use any prior information.2 frequency based methods are: Count Vectorization and TF-IDF Vectorization. However, these methods proved to be limited in their word representations

WordToVec embeddings can draw deeper insights like word analogies and word similarities than frequency based embeddings.

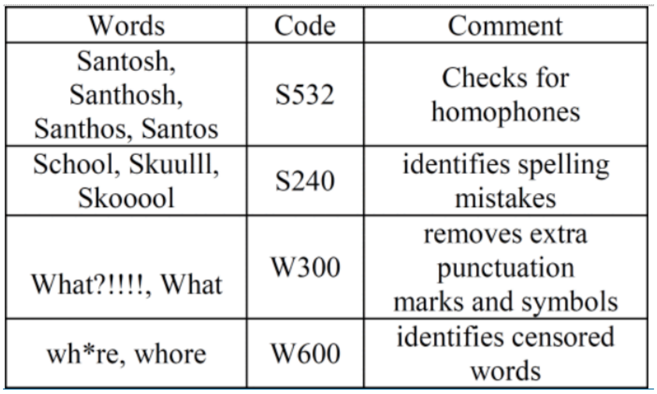
A Bag-of-Phonetic-Codes Representation:



```
1 function ConvertToCode(a) :
Input  : A word consisting of letters or symbols a
Output: code(a)
2 Remove any occurrence of letters belonging to
[a,e,i,o,u,y,h,w] and retain the initial letter of the word.
3 Using the following rules replace the remaining
consonants.
4 [b,f,p,v] : 1
5 [c,g,j,k,q,x,s,z] : 2
6 [d,t] : 3
7 [l] : 4
8 [r] : 6
9 Adjacent Letters having same encodings are replaced
with one occurrence of the code. Letters having same
encodings but separated by an 'h' or a 'w' are encoded
once. But such letters separated by a vowel are counted
twice.
10 If number of digits is too less then the code should be
padded with zeroes otherwise the first three digits are
retained.
```

Fig 2: Soundex Algorithm

In this method, Soundex algorithm as shown in Fig 2. is used to create vocabulary which is a multi-set of phonetic codes. Each tweet is represented as a vector.. The number of unique phonetic codes in the tweet correlates to the number of items in the vector representing a tweet. Words in a tweet are scored, and the results are displayed in the representation at the appropriate area. After generation of respective scores of words, the result will be a 2-D matrix of size M X N, where M is the number of tweets in the training dataset and N is the number of unique phonetic codes in the vocabulary.



Words	Code	Comment
Santosh, Santhosh, Santhos, Santos	S532	Checks for homophones
School, Skuulll, Skooool	S240	identifies spelling mistakes
What?!!!!, What	W300	removes extra punctuation marks and symbols
wh*re, whore	W600	identifies censored words

Fig 3: Examples of similar words and their phonetic codes.

C. Algorithms used for classification

We have used supervised learning approach since unsupervised learning algorithms may overlap and learn to localize the text

Naive Bayes

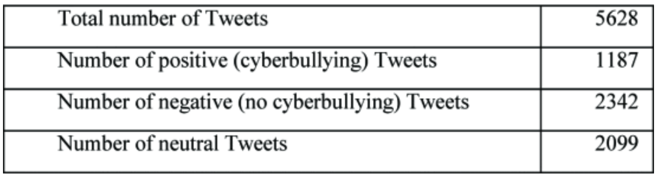
The Naïve Bayes (NB) classifier consists of a family of classifiers. They are of probabilistic nature. It assumes that features are independent of each other. The one we are referring to in this paper is Multinomial Naive Bayes Classifier. NB is reliant on the bag of words presentation of a document.NB classifier neglects words that are not frequently used and only selects the one that are most used.NB has a language modeling.Representation of the text, unigram, bigram, or n-gram are the divisions in language modelling.Further the query corresponding to specific document has a probability which is tested.

Support Vector Machine

Support-Vector Machines (SVMs) are used for classification purposes. SVM classifier tries to find the hyperplane which has the largest margin from the cluster of points of each class. Linear and Radial Basis functions are two most important models for SVM models.

D. Procedure

5628 total tweets have been collected. Fig 4 represents the distribution of tweets



Total number of Tweets	5628
Number of positive (cyberbullying) Tweets	1187
Number of negative (no cyberbullying) Tweets	2342
Number of neutral Tweets	2099

Fig 4. Distribution of tweets

The set of tweets go through various phases of cleaning, preprocessing before the experiment. Dataset is split into a 70:30 ratio for training and testing. Cross-validation is used in which 10-fold equal sized sets are produced.

Tweets with 2-gram, 3-gram, and 4-gram are used to assess and compare the 2 classifiers in terms of accuracy, recall, precision, ROC and f-measure.

E. Bert Models

Bidirectional Encoder Representations from Transformers (BERT) is machine learning technique developed by Google which is used for regression and classification purposes.

BERT Embeddings

The main activity of a BERT model is to generate word and sentence embeddings (inbuilt pooling) for input to classifiers. BERT generates a bidirectional contextual embedding that can lead to different embedding for the same word according to its meaning in the textual context.

BERT Architecture

In our bully detection model, different components/layers are employed, they are as follows:

- Input - The input layer provides the required text sentences in our dataset with the corresponding labels (0-not bully, 1-bully) to the BERT tokenization layer.
- Tokenization - in this layer, a tokenizer provided by the BERT model is used. It is used to convert the words in the text to integer tokens determined by the pre-trained vocabulary of BERT.
- BERT-Base-Model - the core component of the architecture is the BERT embedding model. We have used the "BERT-base-uncased" model which provides a pre-trained model for lowercased English language and consists of 12 layers of transformer encoders to encode the language data. The model is fine-tuned on the dataset to learn dataset-specific vocabulary and generate the corresponding embeddings. It is provided with token _ ids and corresponding attention mask as input for each sentence in the dataset. The special token [CLS] which is also the first token in the input to BERT contains the sentence embedding as a 768 sized vector which is then used to classify the sentence.

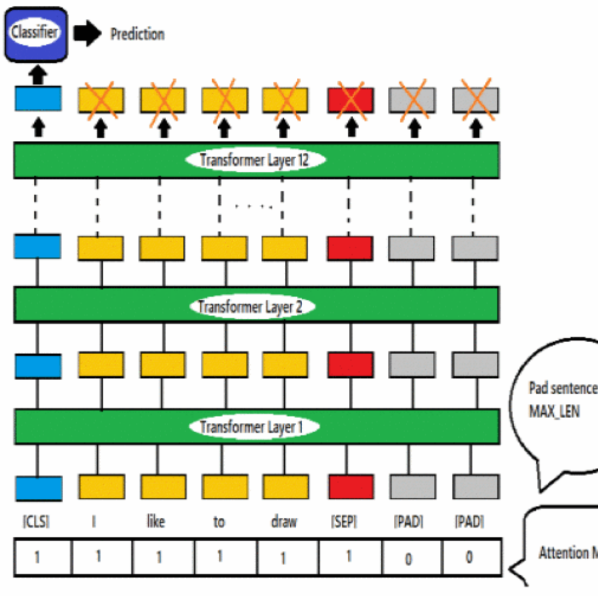


Fig 5. BERT Architecture

- Classifier Layer - this layer consists of a single linear layer of neural network which is trained on the embeddings provided as input from the output of the previous layer. It then classifies the input based on the classes specified.
- Output - output generated by the classification layer is 0or 1. Where 0 represents a non-bully sentence and 1 represents a bully sentence.

Classification using deep learning (Convolution Neural Network)

Methodology

- Convolutional filters learn relevant features from the pair of word embeddings using the 'GloVe' property that comparable words have similar cosine distances, and cosine distances are similar to dot products, and the dot product is essentially a convolution.
- Because we only slide the window in one direction, 1-D convolution is used for text.
- Padding is essential to ensure that the input and output are the same size.
- Use max-pooling to get the maximum activation value from the convolution that passes through the entire text.
- We apply more dense layers and a multi-layered perceptron, and train it for classification tasks.

GloVe: GloVe is a word vector representation acquisition algorithm that uses unsupervised learning. The GloVe model is trained on a global collection of word-word co-occurrence statistics, with the results revealing a linear structure of words in vector space.

Long Short Term Memory

The cell state which is the key of the LSTM, running down the chain, involving interactions. Gates regulate the cell's ability to add or remove info.

· • Forget gate: this gate accepts the input from a previously buried layer and outputs 0 or 1. The numbers 0 and 1 represent forgetting and remembering, respectively.

.

- Input Gate: determines what new information in the cell state should be updated.
- It is divided into two sections:
  - A sigmoid function that determines which values to update.
  - The tanh function generates a new vector of candidate values.

- Update gate: Switch from the previous cell state, ct-1, to the new cell state, ct. The new candidate values have been measured to make each state value up to date..
- The next output will be based on the condition of the filtered version cell. The first sigmoid layer determines which aspects of the cell state will be output. The cell is then ran through the tanh squish with a range of values from -1 to +1 ,further multiplying the output by sigmoid to retrieve only the sections you want.

IV. Results and Discussion

Measure		2 gram	3 gram	4 gram	Average
Accuracy	NB	82.35	81.7	81.1	82.025
	SVM	91.21	91.7	92.02	91.64
Precision	NB	78.46	78.68	78.42	78.52
	SVM	88.92	89.1	89.3	89.11
Recall	NB	77.31	79.4	79.71	78.81
	SVM	86.28	87.36	88.04	87.23
F-Measure	NB	77.88	79.04	79.06	78.66
	SVM	87.58	88.22	88.66	88.16
ROC	NB	78.61	77.9	78.03	77.9
	SVM	88.2	88.56	89.3	88.93

Fig 6. Results of NB vs SVM

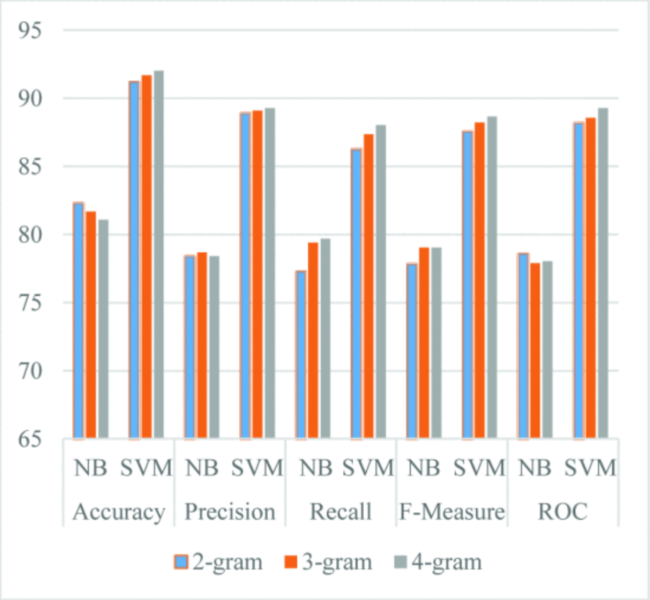


Fig 7. Graphical Comparison of SVM and NB

For both NB and SVM classifiers, Fig. 6 shows the averages of the metrics derived for the different n-grams models. In the instance of the 4-gram language model, SVM classifiers achieved an average accuracy of 92.02 percent, whereas NB classifiers had an average accuracy of 81.1 percent. In addition, in both SVM and NB classifiers, the 4-gram language model outscored all other n-gram language models in all measures. This is due to the fact that a higher n-gram increases the chance of estimation.

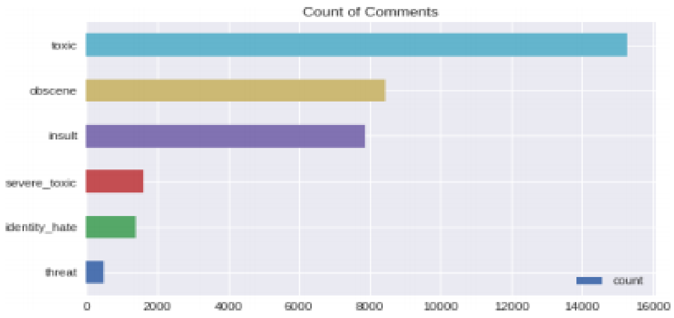
Fig. 8 represents the results obtained for the Formspring dataset for various oversampling rates using BERT model

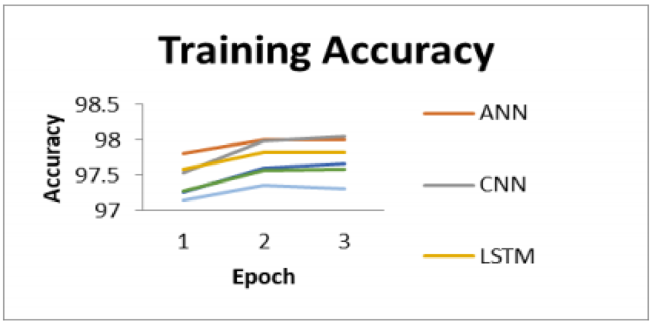
Oversampling Rate	Label	Precision	Recall	F1-Score
1 time	Bully	0.64	0.55	0.59
2 times	Bully	0.82	0.91	0.86
3 times	Bully	0.90	0.99	0.94

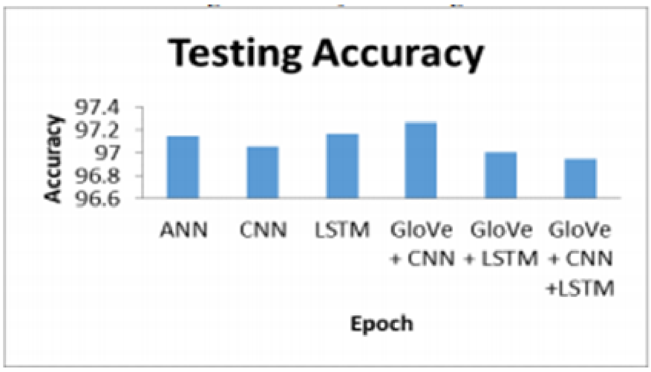
Fig. 8 Results for form spring dataset with varying oversampling rates

The results show that the performance increases with the increasing number of bully posts as it is able to learn more about the bully sentences.

Convolution Neural Networks







ANN: 98 percent training accuracy and minimal loss during loss. However, in terms of testing accuracy, the testing loss is substantial, ranking third among all models.

CNN: Training accuracy improves with each epoch, reaching 97.8% with a 5.42 percent loss. In testing, however, CNN has a lower loss than ANN, but its accuracy is poorer.

LSTM: Training Accuracy is lower than that of ANN and CNN, and the loss is larger. However, in testing, LSTM outperforms previous models and has a smaller loss than previous models.

CNN and GloVe: Because the training accuracy is lower, the loss is larger than in earlier models. During testing, the model outperforms all others, even though its loss is close to that of LSTM

GloVe & LSTM: Because the training accuracy is low, the loss is quite high. In tests, it was found to perform well in terms of accuracy and loss.

GloVe, LSTM, and CNN have the lowest accuracy and have the most loss during training and testing. All other models perform worse than this one.

V. Conclusion

In this paper, we have proposed an approach for cyber bullying detection using the following ML techniques: Naïve Bayes and Support Vector Machine. Prior to training and testing of the tweets, they undergo several preprocessing techniques like stemming, conversion of words to phonetic codes, and cleaning. Then they are tokenized and word embeddings are generated which are fed to models.

The results indicate that SVM classifiers out performs Naive Bayes in all the different scenarios.. Specifically,NB classifiers have 81.1 % accuracy. SVM classifiers achieved an average accuracy value to be 92.02%. for the 4 gram model.

Pre-trained BERT model which is based on the complex and novel deep neural network provides a new approach in cyberbullying detection. Also, it gives improved results in comparison to the previous models.

In conclusion, Glove & CNN performs the best and Glove & CNN & LSTM performs the worst in terms of training and testing, loss and accuracy.

VI. References

[1]A. Shekhar and M. Venkatesan, "A Bag-of-Phonetic-Codes Model For Cyberbullying Detection in Twitter," 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), 2018, pp. 1-7, doi: 10.1109/ICCTCT.2018.8550938.

[2] J. O. Atoum, "Cyberbullying Detection Through Sentiment Analysis," 2020 International Conference on Computational Science and Computational Intelligence (CSCI), 2020, pp. 292-297, doi: 10.1109/CSCI51800.2020.00056.

[3] J. Yadav, D. Kumar and D. Chauhan, "Cyberbullying Detection using Pre-Trained BERT Model," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 1096-1100, doi: 10.1109/ICESC48915.2020.9155700.

[4] M. Anand and R. Eswari, "Classification of Abusive Comments in Social Media using Deep Learning," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019, pp. 974-977, doi: 10.1109/ICCMC.2019.8819734.

[5] M. S. Nikhila, A. Bhalla and P. Singh, "Text Imbalance Handling and Classification for Cross-platform Cyber-crime Detection using Deep Learning," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1-7, doi: 10.1109/ICCCNT49239.2020.9225402.

[6] Desai Aditya, Kalaskar Shashank, Kumbhar Omkar, Dhumal Rashmi. Cyber Bullying Detection on Social Media using Machine Learning. ITM Web of Conferences. 2021;40:03038. doi:10.1051/itmconf/20214003038

[7] Mody A, Shah S, Pimple R, Shekokar N. Identification of Potential Cyber Bullying Tweets using Hybrid Approach in Sentiment Analysis. 2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), 2018 International Conference on. December 2018:878-881. doi:10.1109/ICEECCOT43722.2018.9001476