International Journal of Innovative Research in Science and Engineering
Volume No. 12, Issue No. 05, Year 2026
ISSN : 2454-9665

Detection of Cyberbullying on Social Media Using Machine learning Techniques

Mr.M.Arun Kumar,Assistant professor[1]
A.L.Suvarchala[2],B.Yaswanth[3],G.Nithin[4],G.Bhargavi[5]
Department of Computer Science and Engineering
Tirumala Engineering College

Abstract— Cyberbullying is a major problem encountered on internet that affects teenagers and also adults.It has lead to mishappenings like suicide and depression.Regulation of content on Social media platorms has become a growing need. The following study uses data from two different forms of cyberbullying, hate speech tweets from Twittter and comments based on personal attacks from Wikipedia forums to build a model based on detection of Cyberbullying in text data using Natural Language Processing and Machine learning. Three methods for Feature extraction and four classifiers are studied to outline the best approach. For Tweet data the model provides accuracies above 90% and for Wikipedia data it gives accuracies above 80%.

Keywords—Cyberbullying, Hate speech, Personal attacks, Machine learning, Feature extraction, Google, Wikipedia

I. INTRODUCTION

Now more than ever technology has become an integral part of our life. With the evolution of the internet. Social media is trending these days. But as all the other things misusers will pop out sometimes late sometime early but there will be for sure.Now Cyberbullying is common these days.

Sites for social networking are excellent tools for communication within individuals. Use of social networking has become widespread over the years, though, in general people find immoral and unethical ways of negative stuff. We see this happening between teens or sometimes between young adults. One of the negative stuffs they do is bullying each other over the internet.In online environment we cannot easily said that whether someone is saying something just for fun or there may be other intention of him. Often,with just a joke,"or don't take it so seriously," they'll laugh it off.Cyberbullying is the use of technology to harass, threaten, embarrass, or target another person. Often this internet fight results into real life threats for some individual. Some people have turned to suicide. It is necessary to stop such activities at the beginning. Any actions could be taken to avoid this for example if an individual's tweet/post is found offensive then maybe his/her account can be terminated or suspended for a particular period.

So, what is cyberbullying??

Cyberbullying is harassment, threatening, embarrassing or targeting someone for the purpose of having fun or even by well-planned means

II. BACKGROUND

Researches on Cyberbullying Incidents show that 11.4% of 720 young peoples surveyed in the NCT DELHI were victims of cyberbullying in a 2018 survey by Child Right and You, an NGO in India, and almost half of them did not even mention it to their teachers, parents or guardians. 22.8% aged 13-18 who used the internet for around 3 hours a day were vulnerable to Cyberbullying while 28% of people who use internet more than 4 hours a day were victims. There are so many other reports suggested us that the impact of Cyberbullying is affecting badly the peoples and children between age of 13 to 20 face so many difficulties in terms of health, mental fitness and their decision making capability in any work. Researchers suggest that every country should have to take this matter seriously and try to find solution. In 2016 an incident called Blue Whale Challenge led to lots of child suicides in Russia and other countries . It was a game that spread over different social networks and it was a relationship between an administrator and a participant. For fifty days certain tasks are given to participants . Initially they are easy like waking up at 4:30 AM or watching a horror movie . But later they escalated to self harm which let to suicides. The administrators were found later to be children between ages 12-14.

III. LITERATURE SURVEY

Lot of research have been done to find possible solutions to detect Cyberbullying on social networking sites.Ting,I-

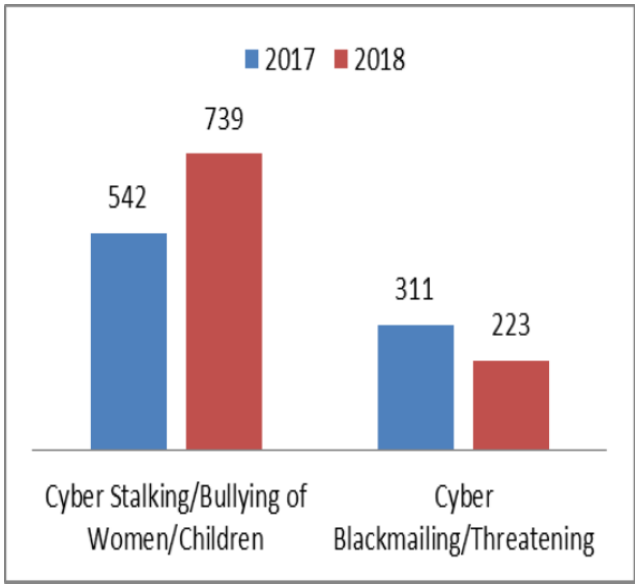

Fig. 1. Cyberbullying cases in India 2017-2018

Hsien[1] used an approach using keyword matching, opinion mining and social network analysis and got a precision of 0.79 and recall of 0.71 from datasets from four websites.Patxi Gal´an-Garc´ıa et al.[2] proposed a hypothesis that a troll(one who cyberbullies) on a social networking sites under a fake profile always has a real profile to check how other see the fake profile. They proposed a Machine learning approach to determine such profiles. The identification process studied some profiles which has some kind of close relation to them. The method used was to select profiles for study, acquire information of tweets, select features to be used from profiles and using ML to find the author of tweets. 1900 tweets were used belonging to 19 different profiles. It had an accuracy of 68% for identifying author. Later it was used in a Case Study in a school in Spain where out of some suspected students for Cyberbullying the real owner of a profile had to be found and the method worked in the case. The following method still has some shortcomings. For example a case where trolling account doesnt have a real account to fool such systems or experts who can change writing styles and behaviours so that no patterns are found . For changing writing styles more efficient algorithms will be needed.

Mangaonkar et al. [3] proposed a collaborative detection method where there are multiple detection nodes connected to each other where each nodes uses either different or same algorithm and data and results were combined to produce results. P. Zhou et al.[4] suggested a B-LSTM technique based on concentration. Banerjee et al.[5]. used KNN with new embeddings to get an precision of 93%.

Kelly Reynolds, April Kontostathis and Lynne Edwards[6] propose a Formpring(A forum for anonymous questions-answers) dataset which gives recall of 78.5% using Machine learning Algorithms and oversampling due to imbalance in cyberbullying posts Jaideep Yadav, Kumar and Chauhan [7] used a latest language model developed by google called BERST which generates contextual embeddings for classification. The model gave a F1 score of 0.94 on form spring data and 0.81 on Wikipedia data.Maral Dadvar and Kai Eckert[8] trained deep neural networks on Twitter,Wikipedia and Formspring datasets and used the model on Youtube dataset for the same and achieved F1 score of 0.97 using Bidirectional Long Short-Term Memory(BLSTM) model.Sweta Agrawal and Amit Awekar [9] used similar same datasets for training Deep Neural Networks but one of its key focus is swear words and their use as features for the task. They determined how the vocabulary for such models varies across various Social Media Platforms.Yasin N. Silva,Christopher Rich and Deborah Hall[10] built BullyBlocker,a mobile application that informs parents of cyberbullying activities against their child on Facebook which counted warning signs and vulnerability factors to calculate a value to measure probability of being bullied

IV. Proposed Methodology

Cyberbullying detection is solved in this project as a binary classification problem where we are detecting two majors form of Cyberbullying: hate speech on Twitter and Personal attacks on Wikipedia and classifying them as containing Cyberbullying or not.

Fig. 2 describes the methodology used for solving the problem which is applied on both the datasets.

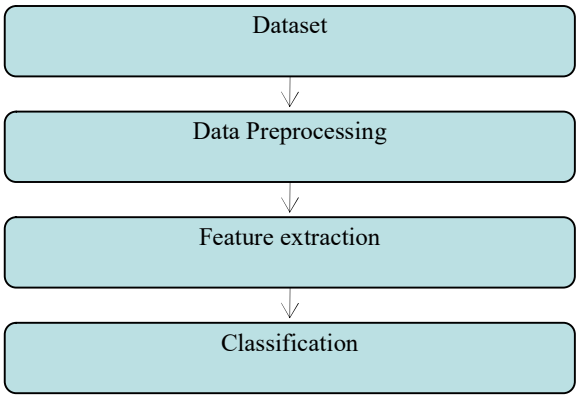

Fig. 2. Methodology

V. Dataset

A. Twitter Dataset

The Twitter Dataset is combined from two datasets containing hate speech :

- Hate Speech Twitter Dataset by Waseem, Zeerak and Hovy, Dirk[11] which contains 17000 tweets labelled for sexism or racism. The tweets are mined using the annotations .5900 tweets are lost due to accounts being deactivated or tweet deleted.
- Hate Speech Language Dataset by Davidson, Thomas and Warmsley, Dana and Macy, Michael and Weber, Ingmar[12].It contained 25000 tweets obtained by crowdsourcing.

This gives total 35787 tweets for the task distribution for which is shown in Fig. 3.For the following dataset, 70 percent(25,050) of this dataset is used as training data and 30 percent as testing data(10,737) .

B. Wikipedia Dataset

The Wikipedia dataset by Wulczyn, Thain and Dixon[13] contains 1M comments labelled for Personal attacks.For the analysis 40000 comments are used from the dataset from which 13000 comments are labelled as Cyberbullying due to personal attack. These comments are extracted from conversations between editors of pages on Wikipedia labelled by 10 annotators via Crowd Flower. For this dataset the same split(70 percent i.e 28000 to training data and 30 percent i.e 12000 to testing data) is used.Fig. 4 shows its distribution

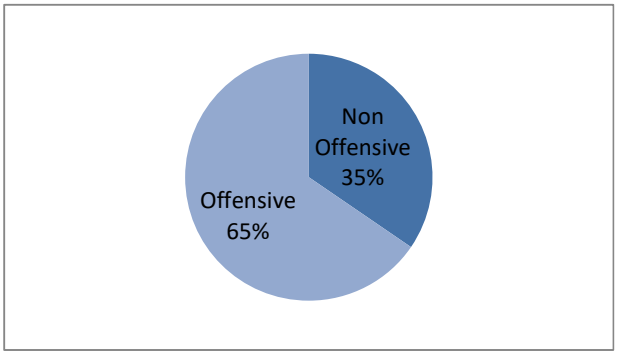

Fig. 3. Distribution of Tweets in Twitter Dataset

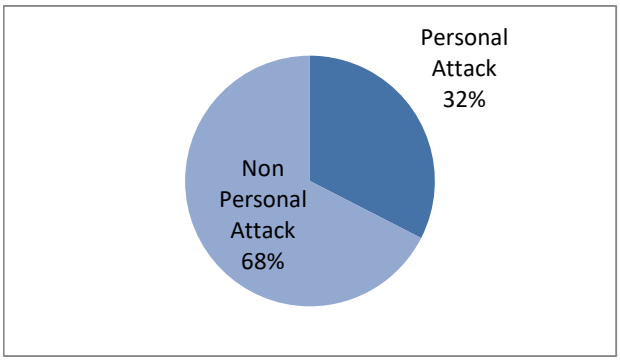

Fig. 4. Distribution of Comments for Wikipedia Dataset

VI. Data Preprocessing

Fig. 5 shows a data processing pipeline used for both the datasets. First all text data are converted to lowercase.Then some words like "what's" or "can't" are converted to "what is" or "can not".Also,all the punctuations are removed using the string library..Then following Natural Language Processing techniques are used using Natural Language Toolkit:

- Tokenization: In tokenization we split raw text into meaningful words or tokens. For example, the text "we will do it" can be tokenized into 'we', 'will', 'do', 'it'. Tokenization can be done into words called word tokenization or sentences called sentence tokenization. Tokenization has many more variants but in the project we use Regex Tokenizer. In regex tokenizer tokens are decided based on rule which in the case is a regular expression. Tokens matching the following regular expression are chosen Eg For the regular expression '\w+' all the alphanumeric tokens are extracted.
- Stemming: Stemming is the process of converting a word into a root word or stem. Eg for three words 'eating' 'eats' 'eaten' the stem is 'eat'. Since all three branch words of root 'eat' represent the same thing it should be recognized as similar. NLTK offers 4 types of stemmers: Porter Stemmer, Lancaster Stemmer, Snowball Stemmer and Regexp Stemmer. The following project uses PorterStemmer.
- Stop word Removal: Stop words are words that do not add any meaning to a sentence eg. some stop words for english language are: what, is, at, a etc. These words are irrelevant and can be removed. NLTK contains a list of english stop words which can be used to filter out all the tweets. Stop words are often removed from the text data when we train deep learning and Machine learning models since the information they provide is irrelevant to the model and helps in improving performance.

VII. Feature Extraction

Feature extraction is important for Natural Language Processing.Texta data can not be classified by classifiers therefore they need to be converted to numerical data. Each document (tweet or comment in this case) can be written as a vector and those vectors can be used for classification. The following project studies three Feature extraction methods:

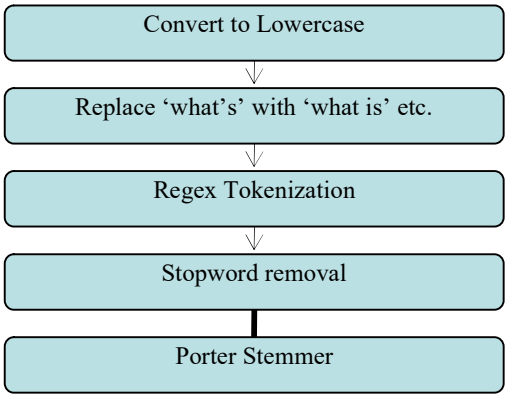

Fig. 5. Data Preprocessing Pipeline

Bag of Words, TF-IDF and Word2Vec.

A. Bag of Words model

The BoW that is bag of words model is a simple method of extracting features from documents that uses occurrence of words within a document. Bag of Words model has two important parts:

- A vocabulary of words(tokens) derived from all documents
- A way of measuring all these words as features in each document

It is referred to as 'bag' because the model only concerns with the word rather than its order of occurrence in the document. The intuition for this method is that similar documents have similar words in them.

The Bag of Words model uses the following procedure:A vocabulary is designed from all the documents. The vocabulary may consist of all words (tokens) in all documents or some top frequency tokens e.g. top 10 features with max occurrences in the corpus. Also features can be extracted for vocabulary in multiple forms based on number of words used per feature. e.g. for the sentence 'This was the best ever'.

- Unigram model where single words are used eg 'this', 'was', 'the', 'best', 'ever' are the features for the corpus.
- Bigram model uses two words at a time for a feature e.g. 'this was', 'was the', 'the best','best ever' are features for the corpus.
- N-gram model is the generalised model where n can be 1,2, 3,... or even more than one value of N can be possible eg. extracting all unigram and bigram features

When the vocabulary is designed what is left is to transform all the documents based on the vocabulary using a way of measuring features. Generally, two types are used, first is a binary one where features are 1 or 0 depending on whether they exist in a document or not. But it does might not work on some sentences. e.g There is a difference between 'very very good' and 'just good'. Therefore we can use the second method i.e frequencies of features in a

documents.Bag of words is a simple but quite effective method for sentiment analysis[14] but has certain limitations. It does not consider context or ordering of words which can make a lot of difference in some cases. Also, Vocabulary design becomes difficult in large datasets due to increase in number of features.

e.g 'Is it interesting' has a different meaning than 'It is interesting'.

B. TF-IDF Model

Tf-Idf method is similar to the bag of words model since it uses the same way to create a vocabulary to get its features. TF-IDF addresses a problem not seen much in the corpus, but is important for better extraction of features. The value of Tf-Idf increases with the increase in frequency of a word in same document and decreases with decrease in frequency of documents that have the word in the corpus. It has two elements, which are

- Term frequency(Tf) is a calculation of frequency of a word in a document. It is measured as chance of finding a text word inside a document. It is measured as the frequency of a word Wi appearing in a document Rj, divided by total words in document Rj

$$tf(W_i, R_j) = \frac{No.of\ times\ W_i\ appears\ \ in\ R_j}{Total\ \ no.of\ documents\ \ in\ R_j} \quad (1)$$

- Inverse document frequency (Idf) shows how frequent or rare a word is throughout the corpus. It is used to identify rare words in corpus. Idf value is higher for rarer words. IDF is getting by dividing the complete number of Words in document D in the corpus by the number of Words in files that consist the term t, and then calculating log value of resulting.

$$idf(d, D) = \log \frac{|D|}{\{d \in D : t \in D\}} \quad (2)$$

In above equation, $|D|$ denotes no of documents in the corpus and denominator term denotes number of documents which have the word t. Sometimes 1 is added to denominator to ensure there is no division by zero.

$$TfIdf(t, d, D) = tf(t, d) * idf(d, D) \quad (3)$$

The high TF-IDF means that word is frequent in a document but rare in the corpus making it more useful as a feature. A low or close to 0 TF-IDF means that these words almost occurs in all document making it less useful as a feature. TF-IDF solves some of the major issues in Bag of Words model thus making it more efficient.

C. Word2Vec

Word2Vec[15] is a Feature extraction method that uses word embeddings which was developed in 2013 by Google. It is used to represent word in vector form. This can be used to find similarity between words as two similar words have smaller angle between their vectors or cosine of angle between them is close to 1

$$sim(A, B) = \cos(\theta) = \frac{A.B}{\|A\|.\|B\|} \quad (4)$$

In (4) A and B are word vectors and θ is angle between both vectors. Word2Vec is a neural network method that uses that uses this as an approach to train the mode land construct word embeddings. There are two methods for the construction of the word embeddings:

- Common Bag of Words Model(CBOW): Common Bag of Words model takes as input of multiple words and predicts the word based on the context. Input can be one word or multiple words. A soft max is used at output. CBOW uses negative log likelihood and is more probabilistic rather than deterministic.
- Skip Gram Model: The skip gram model is just the reverse of CBOW model in which multiple context words are predicted using a single input word.Here the total number of words represented by X are predicted using the neural network. CBOW model takes a mean of context of input words but two semantics can be clicked for a single word. i.e. two vector of Apple can be predicted. First is for the firm Apple and next is Apple as a fruit.

Both of these methods use forward and back propagation to train the neural networks and find the best parameters. For each document then a feature vector can be created by concatenating and combining all word vectors in that document. Combination of word vectors can be done by summation or by averaging all word vectors. Selection between the both is based on data.

VIII. CLASSIFICATION

After getting feature vector for the training data by fitting it on the Feature extraction methods above, testing data is transformed using the same scheme without fitting it on the vectorizers or training it on the word2vec model. Using the training data following classifiers will be trained and tested on.

A. Support Vector Machine(SVM)

This theorem is basically used to plot a hyperplane that creates a boundry between data points in number of features (N)-dimensional space.To optimize the margin value hinge function is one of best loss function for this.Linear SVM is used in the following case which is optimum for linearly seperable data. In case of 0 misclassification, i.e. the class of data point is accurately predicted by our model, we only have to change the gradient from the regularisation arguments.

In case of misclassification, i.e. our model makes a mistake in our data point's class prediction, we add the reduction with the gradient update regularisation.

B. Logistic Regression

It is a classification model and not a regression model.The probabilistic function used to model the ouput of problem is sigmoid function

$$sig(x) = \frac{1}{\{1 + \exp(-x)\}} \quad (5)$$

$$A=LT+C \quad (6)$$

$$T(x) = sig\ (A) \quad (7)$$

In (7) T(x) is hypothesis function for our classifier,L is weights derived by classifier,C is bias derived by classifier and T is feature vector(input).If h(x)>0.5 then class is 1 else class is 0. Since sigmoid lies between 1 and 0 it is ideal for classification.

C. *Random Forest*

A random forest consists of many individual decision trees which individually predict a class forgiven query points and the class with maximum votes is the final result.Decision Tree is a building block for random forest which provides a predicition by decision rules learned from feature vectors.An ensemble of these uncorrelated trees provide a more accurate decision for classification or regression.

D. *Multi Layered Perceptron*

Multi Layered Perceptrons are the Artificial Neural Networks containing at least 3 layers: one input, one output and at least one hidden layer. Each node has a activation value calculated using an activation function in a process called forward propagation and back propagation is used to train the weight used in the neural networks. It is generally used when data is linearly non seperable. Activation functions used can be relu or sigmoid. Sigmoid function is similar to the tanh function which is hyperbolic in nature between -1 and 1. Relu is defined as f(x)=max(0,x). Multi Layered Perceptrons can be created and trained using Keras Framework.

IX. Experiments and Results

Google colab was used for the experiments. For each classifier the following parameters were evaluated on the test sets

- Accuracy(A): is defined as no of correct predictions divided by total number of predictions.

$$A = \frac{True\ positives}{Size\ of\ dataset} \quad (8)$$

- Precision(P): Out of all the positive predictions by the classifier how many are actually positive.

$$P = \frac{True\ positives}{True\ positives + False\ positives} \quad (9)$$

- Recall(R): Out of all the positive inputs how many were predicted positive

$$R = \frac{True\ positives}{True\ positives + False\ negatives} \quad (10)$$

- F-measure(F): Calculates HM (harmonic mean) of precision and helps in comparison of both precision and recall.

$$F = \frac{2 \times P \times R}{P + R} \quad (11)$$

The following configurations are used for the feature selection methods for both datasets used:

- Bag of Words Model: Top 10000 features out of Unigrams, Bigram and Trigram features were selected based on frequency.
- TF-IDF Model: Same as Bag of words model
- For the word2vec model both the skips-gram and Common Bag of Words(CBOW) model were trained. 200 features from both models were combined to get 400 features for each word embedding and for each document summation was used to generate document vector. word2vec was trained on the training sets for 30 epochs with a window of 5 words.

The classifiers were loaded through sklearn library except the Multi Layer Perceptrons which were made in Keras. Two MLPs were used: one for Bag of Words and tfidf for 10000 feature input and other for 400 feature input of Word2vec. The architectures of both neural networks are shown in Fig. 6 and Fig. 7. The Classifiers used are Linear SVM (SVC), Random Forest Classifier (RF), Logistic Regression (LR) and Multi Layered Perceptron (MLP).

Tables 1 and 2 show results for Twitter and Wikipedia dataset respectively.

The Twitter dataset which contained tweets related to Hate speech show F- measures above 0.9 for all three feature selection methods. The values for Word2Vec model are a bit less but are ideal considering it used 400 features instead of other methods using 10000. TF-IDF method combined with Linear SVM gives best recall and F-measure.

For the Wikipedia dataset which contained comments with Personal attacks it shows F-measures only around 0.8 for all models. The TF-IDF with Linear SVM still get the best F-measure but Word2Vec with Multi Layered Perceptron gives better recall.

X. Conclusion

Cyber bullying across internet is dangerous and leads to mishappenings like suicides, depression etc and therefore there is a need to control its spread. Therefore cyber bullying detection is vital on social media platforms. With avaibility of more data and better classified user information

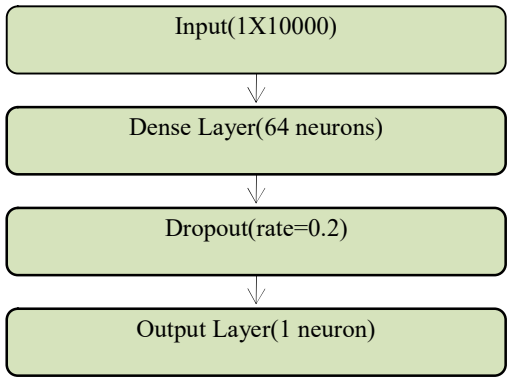

Fig. 6. Multi Layered Perceptron used for Bag of Words and TF-IDF inputs

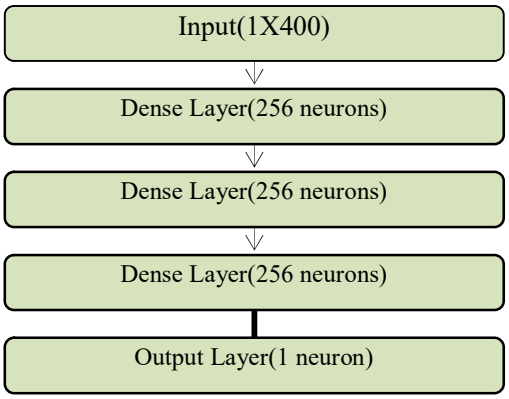

Fig. 7. Multi Layered Perceptron used for Word2Vec input

for various other forms of cyber attacks Cyberbullying detection can be used on social media websites to ban users trying to take part in such activity In this paper we proposed an architecture for detection of cyber bullying to combat the situation.We discussed the architecture for two types of data: Hate speech Data on Twitter and Personal attacks on Wikipedia.For Hate speech Natural Language Processing techniques proved effective with accuracies of over 90 percent using basic Machine learning algorithms because tweets containing Hate speech consisted of profanity which made it easily detectable.Due to this it gives better results with BoW and Tf-Idf models rather than Word2Vec models However, Personal attacks were difficult to detect through the same model because the comments generally did not use any common sentiment that could be learned however the three feature selection methods performed similarly.Word2Vec models that use context of features proved effective in both datasets giving similar results in comparatively less features when combined with Multi Layered Perceptrons.As seen by changing nature of

REFERENCES

[1] I. H. Ting, W. S. Liou, D. Liberona, S. L. Wang, and G. M. T. Bermudez, "Towards the detection of cyberbullying based on social network mining techniques," in *Proceedings of 4th International Conference on Behavioral, Economic, and Socio-Cultural Computing, BESC 2017*, 2017, vol. 2018-January, doi: 10.1109/BESC.2017.8256403.

[2] P. Galán-García, J. G. de la Puerta, C. L. Gómez, I. Santos, and P. G. Bringas, "Supervised machine learning for the detection of troll profiles in twitter social network: Application to a real case of cyberbullying," 2014, doi: 10.1007/978-3-319-01854-6_43.

[3] A. Mangaonkar, A. Hayrapetian, and R. Raje, "Collaborative detection of cyberbullying behavior in Twitter data," 2015, doi: 10.1109/EIT.2015.7293405.

[4] R. Zhao, A. Zhou, and K. Mao, "Automatic detection of cyberbullying on social networks based on bullying features," 2016, doi: 10.1145/2833312.2849567.

[5] V. Banerjee, J. Telavane, P. Gaikwad, and P. Vartak, "Detection of Cyberbullying Using Deep Neural Network," 2019, doi: 10.1109/ICACCS.2019.8728378.

[6] K. Reynolds, A. Kontostathis, and L. Edwards, "Using machine learning to detect cyberbullying," 2011, doi: 10.1109/ICMLA.2011.152.

[7] J. Yadav, D. Kumar, and D. Chauhan, "Cyberbullying Detection using Pre-Trained BERT Model," 2020, doi: 10.1109/ICESC48915.2020.9155700.

[8] M. Dadvar and K. Eckert, "Cyberbullying Detection in Social Networks Using Deep Learning Based Models; A Reproducibility Study," *arXiv*. 2018.

[9] S. Agrawal and A. Awekar, "Deep learning for detecting cyberbullying across multiple social media platforms," *arXiv*. 2018.

[10] Y. N. Silva, C. Rich, and D. Hall, "BullyBlocker: Towards the identification of cyberbullying in social networking sites," 2016, doi: 10.1109/ASONAM.2016.7752420.

[11] Z. Waseem and D. Hovy, "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter," 2016, doi: 10.18653/v1/n16-2013.

[12] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," 2017.

[13] E. Wulczyn, N. Thain, and L. Dixon, "Ex machina: Personal attacks seen at scale," 2017, doi: 10.1145/3038912.3052591.

[14] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: a review," *Artif. Intell. Rev.*, vol. 53, no. 6, 2020, doi: 10.1007/s10462-019-09794-5.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

TABLE 1: RESULTS OF DATASET

Measure	Bag of Words				TF-IDF				WORD2VEC			
	SVC	RF	LR	MLP	SVC	RF	LR	MLP	SVC	RF	LR	MLP
Accuracy	0.906	0.914	0.921	0.903	0.920	0.914	0.917	0.911	0.890	0.867	0.894	0.898
Precision	0.935	0.947	0.959	0.930	0.949	0.949	0.952	0.937	0.935	0.886	0.935	0.932
Recall	0.920	0.921	0.920	0.922	0.927	0.918	0.920	0.927	0.894	0.915	0.901	0.918
F-measure	0.928	0.934	0.939	0.927	0.939	0.933	0.936	0.932	0.914	0.901	0.918	0.922

TABLE 2: RESULTS FOR WIKIPEDIA DATASET

Measure	Bag of Words				TF-IDF				WORD2VEC			
	SVC	RF	LR	MLP	SVC	RF	LR	MLP	SVC	RF	LR	MLP
Accuracy	0.871	0.886	0.890	0.876	0.894	0.887	0.892	0.869	0.876	0.857	0.879	0.879
Precision	0.817	0.892	0.879	0.828	0.879	0.906	0.915	0.814	0.833	0.828	0.832	0.814
Recall	0.798	0.755	0.785	0.803	0.798	0.745	0.752	0.795	0.795	0.731	0.808	0.835
F-measure	0.808	0.818	0.829	0.815	0.837	0.818	0.825	0.805	0.813	0.776	0.820	0.825