Cyberbullying Detection – A Solution with Voting Classifier

Ali Shahidi
Department of Electrical and Software Engineering, University of Calgary
Calgary, Alberta, Canada
+1 (403) 437 - 4734
ali.shahidi@ucalgary.ca

Ahmadreza Nazari
Department of Electrical and Software Engineering, University of Calgary
Calgary, Alberta, Canada.
+1 (825) 994 - 2599
ahmadreza.nazari@ucalgary.ca

ABSTRACT
Social media has become an inseparable part of human life. With this, the use of electronic communication to bully a person, cyberbullying, has also grown significantly. Uncovering the detrimental effects of cyberbullying, especially on adolescents, taking immediate action seems to be necessary. Many countries have tried to address this rising problem by imposing restrictive laws and declaring it a criminal act. This idea will not suffice, considering the scale of social communication on the internet and the anonymity of the users. Hence, research on automated cyberbullying detection has surged over the past decade. One of the most popular social media vulnerable to cyberbullies is Twitter. In this paper, we perform extensive experiments with different algorithms including deep learning techniques. We have proposed a solution that stands up to some state-of-the-art algorithms. We have created a dataset containing 49,134 unique tweets by merging three different datasets. By evaluating multiple data mining and supervised machine learning techniques to select the best subsystems for the system, we reached an F1-Score of ~93.

General Terms
Algorithms, Measurement, Performance, Design.

Keywords
Cyberbullying, Twitter, Data Mining, Machine Learning

1. INTRODUCTION
With the rapid growth of social media, cyberbullying, which is considered a cybercrime, is increasing, especially during the COVID-19 pandemic [1]. Cyberbullying is the act of harassment of others through social media, emails, text messaging, etc. It can be known or anonymous, and it happens twenty-four hours a day, seven days a week. Cyberbullying is a serious issue since it may affect adolescents' mental health and well-being [1], [2]. Therefore, it is crucial to detect it in real-time. One of the biggest platforms prone to cyberbullying is Twitter, a microblogging social media, which users can broadcast their daily activities [3]. Twitter had over 206 million monetizable daily active users worldwide in the second quarter of 2021 [4]. This number of users, Twitter's nature, anonymity, and having access to it from anywhere by anyone could bring some new trends to the community, one of which is cyberbullying. Detecting cyberbullies' posts and comments manually is impossible. Most of the twitter's contents are texts. Therefore, with the text mining process, we can detect cyberbullying automatically by finding patterns in the text [5].

In this project, we aim to try different text mining and deep learning models, Long Short-Term Memory, to increase the accuracy and F1-score of the related works. Furthermore, we combine different datasets to create the bigger one.

The rest of the paper is organized as follows: In section 2, we go through some related works that have been done in this field, section 3 explains the current research gap which we want to cover, section 4 describes our proposed method in detail. Section 5 describes the experiment and presents the results from this experiment.

2. Related Work
Several approaches to the topic of cyberbullying detection have been investigated in the literature. M. Gada et al. [6] used a hybrid solution with combining the LSTM, as the first layer, and CNN architectures and achieved 95.2% accuracy, however, their F1-Score for this model is only 0.44 and this means the data in their dataset mostly is biased to one of the classes. C. Iwendi et al. [7] have worked on BLSTM and LSTM to detect cyberbullies, and their best-proposed model is Bidirectional LSTM (BLSTM) with an accuracy of 82.18% and an F1-Score of 0.88. They utilized different data pre-processing techniques, including Stemming, Lemmatization, and stop words. Riadi et al. [5] used a Naïve Bayes classifier that performs data processing in numeric data using Bayesian probability. This technique has the learning ability, so it works better than the rule-based technique (if-then). On the other hand, this algorithm does not consider the dependency of the words in one sentence. In 2017 P. Badjatiya et al. [8] achieved a 0.93 F1-Score by combining LTSM, Random Embedding, and GBDTs. Z. Zhang et al. [9] propose a new architecture, Convolutional Neural Networks + LSTM, and try their model on a different dataset. Their F1-Score for each dataset varies around 89 to 94.1. However, they did not concatenate the datasets and tried their model separately.

3. Research Gap
Even though research on cyberbullying detection has been extensive, current systems still lack accuracy and generalizability. To improve on current systems, we decided to use a combination of different datasets so that our system does not rely on one specific dataset. Also, we propose a solution that utilizes multiple algorithms to label a tweet, voting classifier. The details of the system are described in the next section.

4. Methodology
The goal is to design a system to classify the tweets into two categories: offensive, and non-offensive. We break down the project into different steps as follow:

1. Find and/or prepare the suitable dataset
2. Pre-process and clean the data
3. Build and train the classifier

4.1 Dataset

We went through multiple datasets and based on their diversity and publicity we chose three of them to use in this project.

1. The first dataset contains 24,784 tweets with 3 different labels: hate speech, offensive, and neither. To fit this dataset for our use case, we consider hate speech as offensive. This distribution of the tweets in this dataset is about 83% offensive tweets, and 17% non-offensive [10].
2. The second dataset has 8,798 tweets, with offensive or non-offensive labels. 68% of the tweets are non-offensive, and about 32% of them are offensive [11].
3. The third dataset includes 16,852 tweets and is labeled with Sexism, Racism, and None. For this project, Sexism and Racism as considered offensive, and None is considered non-offensive. There has been a lot of work on this dataset [12], however, only about 32% of this dataset is labeled offensive.

To achieve the best results, we decided to concatenate all three datasets to not only increase the number of tweets but also to have a balanced distribution of the data among the two categories, offensive and non-offensive. Since we need to combine all three datasets, first we merge all offensive labels including sexism, racism, hate speech into one category, offensive. After removing additional columns, null data, mapping the columns to the labels, and removing duplicate tweets, we created a larger dataset that contains 49,134 tweets. 57% of tweets are offensive and 43% of them are non-offensive.

4.2 Data Pre-Processing

The next step in the text classification problem is to clean data and prepare it for making the best use of it. Figure 1 depicts all data pre-processing techniques used in this project.

4.2.1 Text Cleaning

To have clean data, we need to reduce the sentences to the words that will provide us with useful information in our problem. Punctuations, numerical and emojis do not have any useful information for detecting the cyberbullies' tweets. Furthermore, using capital letters is the same as using the lower ones. Therefore, after converting all the words in the data set to lower case, we eliminate all symbols, such as emojis, numerical, and punctuation.

4.2.2 Stop Words

The dataset contains a lot of words that can be counted as noisy data when we use text classification algorithms. We call these stop words. Stop words are words that are very common in the English language and, they do not have any specific meaning such as I, me, you, etc. Therefore, they do not provide any useful pieces of information. Hence, in the second step, we delete stop words from the dataset.

4.2.3 Lemmatization

Also, we use the Lemmatization technique which is a process of grouping together the inflected forms of a word so they can be analyzed as a single item to reduce the word to its lemma. For example, eat, eating, ate are all forms of the word eat, therefore eat is the lemma of all these words.

4.2.4 Vectorizing

Furthermore, for feeding our data to the models, we need to create a matrix of all the words in the dataset. For each tweet, we have all the words in all the tweets and the number of occurrences in that tweet, in this case, each word could be seen as a different feature. To increase the change of convergence and avoid overfitting we need to limit the number of features. For this purpose, we only consider the X word with the most occurring in the dataset.

4.2.5 Tokenizing

For feeding the data to the LSTM model, we should use different methods for tokenizing the sentence to get the best result. Since we want to use the data for the deep learning models, we cannot give a vector of words as an input because these models only accept the numerical values as an input. Hence, we must develop a way to map each word to a number. For this purpose, we used the Tokenizer function. This function will assign a unique index to every word. After that, we map each word to its index in the dataset. Now, instead of having a list of words for each tweet, we have a list of indexes for them. We have 39,519 unique words in our dataset. However, we only consider X most frequent words.

Additionally, we must also do the same thing for the labels list; instead of having offensive and normal tweets, we use one-hot-encoding to represent the labels.

Furthermore, we cannot have different input sizes for each tweet for these models. So, first, we use a maximum limit of 20 for the number of words in each tweet.

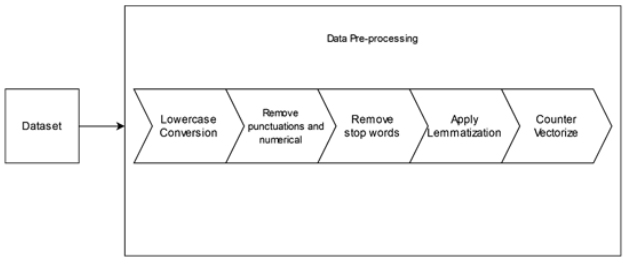

Figure 1 Data Pre-Processing Steps

4.2.6 Train, Validation, Test Splits

We use a 70-15-15 percent split for a train set, validation set, and test set respectively. For evaluating the model correctly, it is better if the train, validation, and test set all have the same distribution. For feeding our data to the data mining classifiers, we only need train and test sets which we split into two subsets, the train set containing 80 percent of the data and the test set containing 20 percent of it.

4.3 Models

We have managed to utilize seven different models to build a classifier. The following is the summary of them:

1. Multimodal Naive Bayes: This model calculates the probability of each tag based on the words in each tweet and its output is the tag with the highest probability. One of the advantages of this model is it is highly scalable and can easily handle a large dataset. However, it is mostly used for textual data and cannot predict numerical value.

2. K-NN: In this model, an object will be classified by a sum of the votes of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. This model is a lazy learner, which means it does not learn anything in the training period. It stores the training dataset and learns from it only at the time of making real-time predictions. And when new data is added, it does not change the accuracy of the model.

3. Decision Tree: In this model, we have a tree in which internal nodes are the features, and each branch represents the outcome of the test. In this tree, the leaves are the class label. In this model, we do not need to normalize or scale the data, on the other hand, it needs more time to train the data and small changes could affect the whole structure.

4. Logistic Regression: This model is a statical model which is used to predict the test value based on the observation of the train data.

5. Bagging Classifier: This algorithm resamples the training set and gives each of these sets to different base classifiers, after that, each classifier trains independently with a subset of the training set that has been assigned and at the end, the final prediction would be the major of the votes for that label. It causes to reduce the overfit because of the voting process.

6. Voting Classifier: Until this point, we run every model independently, now we want to have a classifier that is somehow a combination of all these models, we want to predict the output label based on the majority voting of the previous models that we ran. For this purpose, we use Voting Classifier. This method votes among the first 5 algorithms.

7. Long Short Term Memory model is one of the recurrent neural networks and can memorize important information. It means this algorithm considers the dependency of the words in one sentence. Before feeding the data to the LSTM, we decided to add an Embedding layer, and this drastically improved the results. Figure 3 illustrates the result of the training process. In this model, the learning rate is reduced in each epoch. At first, the learning rate has its maximum values to allow the model to learn faster, we reduce it in each epoch to make sure that our model will learn the optimal weights. Also, to avoid overfitting, we only evaluate our model with the best model that we have achieved which has the highest F1-Score for the validation set.

4.4 Metrics

We managed the balance of the classes in the dataset by combining different datasets, so, accuracy would be a good metric for measuring the result. On the other hand, the cost of false positives and false negatives are high, therefore, we need high precision and recall. For this purpose, one of the best choices for the evaluation of our model is using the F1-Score. Hence, we have sorted our result based on the F1-Score that we obtained from the test set.

5. Results and Discussion

To sum it up, we combined three different Twitter datasets for cyberbullying together and did data pre-processing on them, and tried different LSTM models and text mining classifiers to find our dataset's best model to detect the target tweet as either non-offensive or the offensive one in real-time. As shown in Table 1, the best accuracy belongs to Logistic Regression with an accuracy of 87.52% and the F1-Score of 0.88.

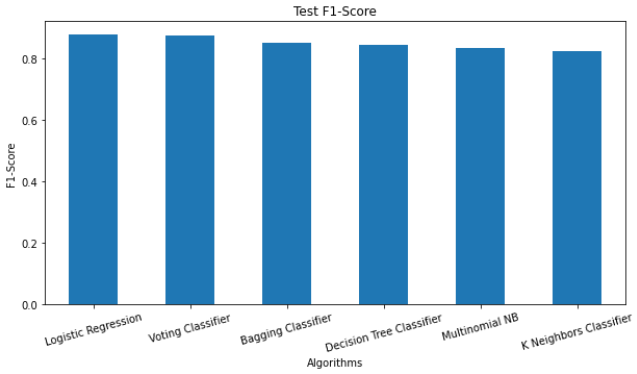

Figure 2 F1-Score for the test set including the third dataset

Table 1 Results when including the thirds dataset

Algorithm	Test Accuracy	Train Accuracy	Test F1 Score	Train F1-Score
Logistic Regression	87.52	91.85	0.889	0.92
Voting Classifier	87.55	95.73	0.888	0.96
LSTM	0.86	0.87	0.86	0.87
Bagging Classifier	85.59	98.39	0.87	0.98
Decision Tree Classifier	83.94	99.61	0.86	0.99
Multinomial NB	82.21	84.69	0.85	0.87
K Neighbors Classifier	80.79	85.13	0.81	0.85

After analyzing the results, we concluded that the third dataset categorization has some conflicts with the other two. So, we decided to remove that from the dataset and only use the other two to see the changes in our result. As you can see in Table 2, not only our accuracy gets better, but also, we achieved an F1-Score of 0.93. It should be noted that the voting classifier has been

implemented but the F1-Score of 0.93 on an extensive dataset has never been reached before.

As shown in table 2, the best accuracy belongs to the Voting Classifier system with an accuracy of 90.83 % and the F1-Score of 0.93. As you can see in Figure 3, in each epoch the validation loss is reducing and the F1-Score is increasing, which means our model is learning.

Figure 4 shows the result of different algorithms when excluding the third dataset.

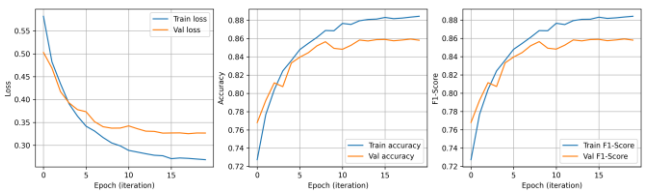

Figure 3 LSTM Results excluding the third dataset

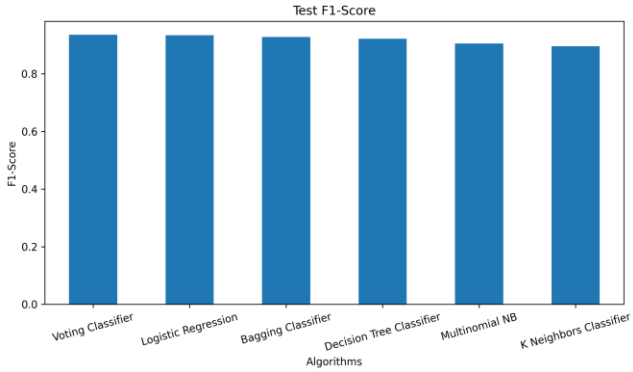

Figure 4 F1-Score for the test set excluding the third dataset

Table 2 The results of our models without the third dataset

Algorithm	Test Accuracy	Train Accuracy	Test F1 Score	Train F1-Score
Voting Classifier	90.8331	97.6420	**0.9354**	0.9835
Logistic Regression	90.6008	94.9124	0.9336	0.9642
Bagging Classifier	89.6717	98.8655	0.9278	0.9921
Decision Tree Classifier	88.2781	99.7134	0.9187	0.9980
Multinomial NB	86.0018	87.7961	0.9055	0.9171
LSTM	0.8969	0.9144	0.8970	0.9143
K Neighbors Classifier	85.5373	89.6623	0.8958	0.9259

6. ACKNOWLEDGMENTS

Our thanks to Dr. Reda Alhajj.

7. REFERENCES

[1] D. Sultan, A. Suliman, A. Toktarova, B. Omarov, S. Mamikov, and G. Beissenova, “Cyberbullying detection and prevention: Data mining in social media,” *Proc. Conflu. 2021 11th Int. Conf. Cloud Comput. Data Sci. Eng.*, pp. 338–342, Jan. 2021, doi: 10.1109/CONFLUENCE51648.2021.9377077.

[2] L. K. Watts, J. Wagner, B. Velasquez, and P. I. Behrens, “Cyberbullying in higher education: A literature review,” *Comput. Human Behav.*, vol. 69, pp. 268–274, Apr. 2017, doi: 10.1016/J.CHB.2016.12.038.

[3] L. Hon and K. Varathan, “Cyberbullying detection system on twitter,” *IJABM*, vol. 1, no. 1, pp. 1–11, 2015.

[4] “Twitter: most users by country,” *Statista Research Department*. 2021, Accessed: Dec. 21, 2021. [Online]. Available: https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/.

[5] I. Riadi, “Detection of cyberbullying on social media using data mining techniques,” *Int. J. Comput. Sci. Inf. Secur.*, vol. 15, 2017.

[6] M. Gada, K. Damania, and S. Sankhe, “Cyberbullying Detection using LSTM-CNN architecture and its applications,” *2021 Int. Conf. Comput. Commun. Informatics, ICCCI 2021*, Jan. 2021, doi: 10.1109/ICCCI50826.2021.9402412.

[7] C. Iwendi, G. Srivastava, S. Khan, and P. K. R. Maddikunta, “Cyberbullying detection solutions based on deep learning architectures,” *Multimed. Syst.*, vol. 1, pp. 1–14, Oct. 2020, doi: 10.1007/S00530-020-00701-5/TABLES/3.

[8] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, “Deep Learning for Hate Speech Detection in Tweets,” *Proc. 26th Int. Conf. World Wide Web companion*, pp. 759–760, 2017, doi: 10.1145/3041021.3054223.

[9] Z. Zhang, D. Robinson, and J. Tepper, “Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10843 LNCS, pp. 745–760, Jun. 2018, doi: 10.1007/978-3-319-93417-4_48.

[10] T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated hate speech detection and the problem of offensive language,” in *Proceedings of the International AAAI Conference on Web and Social Media*, 2017, vol. 11, no. 1.

[11] F. Elsafoury, “Cyberbullying datasets,” *Mendeley. com.[Online]. Available: https://data. mendeley. com/datasets/jf4pzyvnpj/1,[Accessed: 04-Summer-2021]*, 2020.

[12] Z. Waseem and D. Hovy, “Hateful symbols or hateful people? predictive features for hate speech detection on twitter,” in *Proceedings of the NAACL student research workshop*, 2016, pp. 88–93.