CYBERBULLYING DETECTION USING ENSEMBLE METHOD

Saranyanath K P[1] and Wei Shi[2] and Jean-Pierre Corriveau[1]

[1]School of Computer Science, Carleton University, Ottawa, Canada
[2]School of Information Technology, Carleton University, Ottawa, Canada

ABSTRACT

Cyberbullying is a form of bullying that occurs across social media platforms using electronic messages. This paper proposes three approaches and five models to identify cyberbullying on a generated social media dataset derived from multiple online platforms. Our initial approach consists in enhancing Support Vector Machines. Our second approach is based on DistilBERT, a lighter and faster Transformer model than BERT. Staking the first three models we obtain two more ensemble models. Contrasting the ensemble models with the three others, we observe that the ensemble models outperform the base model concerning all evaluation metrics except precision. While the highest accuracy, of 89.6%, was obtained using an ensemble model, we achieved the lowest accuracy, at 85.53% on the SVM model. The DistilBERT model exhibited the highest precision, at 91.17%. The model developed using the different granularity of features outperformed the simple TF-IDF.

KEYWORDS

Machine Learning, Natural Language Processing, Support Vector Machine, DistilBERT, Cyberbullying.

1. INTRODUCTION

The emergence of Internet and various multimedia applications has enabled the communication over social-media platforms. The number of users accessing such applications is increasing rapidly. This has resulted in bullying general or specific users and user groups, either knowingly or unknowingly. The abuses resulting from cyberbullying can cause psychological harm to the target users and groups [1].

Cyberbullying is defined as 'an aggressive, intentional act carried out by a group or individual, using electronic forms of contact, repeatedly and over time against a victim who cannot easily defend him or herself [2]. Sending vulgar messages, posting private information without an individual's consent, frequently sending offensive messages, online gossip spreading, cyberstalking etc can be considered as actions that could be termed as Cyberbullying. Studies show that about half of American teenagers have experienced cyberbullying and victims often have psychiatric and psychosomatic disorders. 8% teens have reported some form of cyberbullying among the total reported 19% bullying cases.

Cyberbullying can take place using any type of data. Text-based cyberbullying can be defined as the act of cyberbullying using texts for sending bullying messages or posts. To identify text-based cyberbullying, text classification plays a prominent role. A classification example of email involves categorising them into spam or non-spam, bullying or non-bullying. The data

David C. Wyld et al. (Eds): ARIA, SIPR, SOFEA, CSEN, DSML, NLP, EDTECH, NCWC - 2022
pp. 75-94, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121507

classification can be achieved using classification algorithms like Naive Bayes, SVM, Neural networks and NLP.

Due to an increase in the volume of data being shared over the social media platforms, it is tedious to implement a manual approach to cyberbullying detection. Hence, machine learning models for text-based cyberbullying detection can be used as an initial mechanism to reduce the manual efforts in reviewing the content [3]. The count, density and value of offensive words can be used as features to detect cyberbullying messages. Instead of actual textual features, few works have promoted the usage of complementary information that would supplement textual cyberbullying detection. The history of user's activities, location, user personalities and emotions were considered.

Several models have been developed and modified to date using many of the state-of-the-art technologies in identifying and preventing cyberbullying detection. These models were developed using machine learning and deep learning algorithms, which have the capability of learning human data. Supervised machine learning algorithms were used to classify online harassment on MySpace and Slashdot datasets, to compare the performance of various classifiers on binary and multi-classification problems using Naive Bayes (NB), SVM on Youtube comments [4]. These algorithms can be used to identify cyberbullying by combining it with the labelled data. The existing works utilizes the capabilities of only one of the machine learning, deep learning, or word embeddings techniques. We focus on combining the approaches to leverage the capabilities and to improve the performance. A list of contributions is summarized below:

1. We perform Cyberbullying detection using SVM, DistilBERT and Stacked ensemble model on our newly generated social media dataset.
2. We conduct an empirical evaluation on different levels of granularity of feature extraction methods in TF-IDF such as Word, Character and N-gram sequencing on SVM model.
3. We perform and present the results of a comparative evaluation of the five developed models in terms of various evaluation metrics. The sets of models evaluated are:

   (a) Traditional SVM model implemented using TF-IDF for Words.
   (b) An improved SVM model proposed by Sharma et al.[5] combined with the tokens of Word, Character and N-gram in TF-IDF for feature extraction.
   (c) DistilBERT model with classification layer on top.
   (d) Stacked ensemble model by combining the base models explained in (a), (b) and the DistilBERT model in (c).

4. We present a detailed analysis on impact of these models on cyberbullying detection. In summary:

   (a) The traditional SVM model with TF-IDF for words yields the worst accuracy and SVM model with different tokens of TF-IDF (i.e., Words, characters, and N-gram) yield accuracies similar to that of the DistilBERT model.
   (b) The DistilBERT model yields the best precision.
   (c) The ensemble models outperform all individual base models. Furthermore, when using combined tokens of TF-IDF with SVM and DistilBERT embeddings, we achieve an accuracy of 89.6%.

The rest of the paper is organized as follows: we briefly introduce the background in Section 2 and review the related work in Section 3. In Section 4, we present the data pre-processing steps

and explain the details on the models developed. We report on the analysis of our obtained results in Section 5 and make the conclusions in Section 6.

2. BACKGROUND

In this Section, we provide an insight to the technical information on the methodologies which are relevant for the text classification approaches. The major topics discussed in Section 2.1 is Feature extraction. Section 2.2 describes the TF-IDF used in traditional machine learning algorithms. The DistilBERT is described in Section 2.3.

2.1. Feature Extraction

Feature extraction is a method by which raw data, of any formats such as text, image, video is transformed into an acceptable internal representation or a feature vector from which any learning sub-system such as a classifier, can identify input patterns [6]. Feature extraction is considered a critical step in cyberbullying for text classification [7]. The basis of an enormous amount of text processing is the text feature extraction, in which the text information is extracted to represent a text message [6]. An important factor in classifying texts, according to the machine learning models is to digitize them [8]. The machine learning classifiers are trained using the numerical format of the input data. By applying various feature extraction techniques, every text information needs to be converted into a numerical representation. The dimension of a feature space is reduced by means of feature extraction[9]. Redundant and uncorrelated information will be deleted through feature extraction. The reduction of features will assist in improving the accuracy of the algorithms and hence speeds up the processing time. Text feature extraction directly influences the accuracy of text classification. The text feature extraction is based on the vector space model, and the text is observed as a dot in N-dimensional space. The common methods of text feature extraction are Filtration, Fusion, Mapping.

2.1.1. Filtering Method

Filtering method faster and is suitable for extensive text feature extraction. Filtration of text feature extraction comprises of word frequency, information gain, and mutual information method [9].

1. Word frequency: Word frequency is defined as the number of times a word appears in a text. To reduce the dimensionality of feature space using feature selection, words whose frequencies which are less than a certain threshold are deleted. The deletion criteria are based on a hypothesis that words with small frequencies will have a less impact on filtration. In terms of information retrieval, the words with less frequency of occurrences may have more information. Thus, it may be unseemly to remove the words only based on the word frequency.
2. Mutual information: MI (mutual information) is a commonly used method for mutuality in the analysis of computational linguistics models. MI helps to retrieve the differentiation of features. MI represents the relationships between information and the statistical measurement of correlation of two random variables. MI helps to create a table of association of words from a large corpus. If a feature belongs to a class, it is said to have largest amount of mutual information. A drawback of MI is that the score is regulated by the marginal probabilities of words.
3. Information gain: IG (information gain) is employed in machine learning to measure whether a known feature appears in a text of a certain applicable topic and the prediction rate of the information on the topic. The features that occur frequently in positive or

negative samples can be obtained by computing IG. The IG is computed on each feature based on the training data and deletes those features which has small information gain, and the remaining features are ranked in descending order based on the IG.

2.1.2. Mapping Method

1. Latent Semantic Index: Mapping has been used in text classification and has shown to achieve good results [9]. The commonly used mapping methods is LSI (latent semantic index). LSI is an algebraic model introduced in 1988 by S.T. Dumais. LSI reduces the dimensionality of text vectors by extracting and employing the latent semantic structure between words and texts. The mapping is achieved through SVD (singular value decomposition) of item or document matrix. LSI can be used in text classification, information extraction, information filtering.

2. Least squares mapping method is based on centre vector and least squares. The clustered centre vectors reflect the structures of raw data, whereas SVC did not consider these structures.

2.2. TF-IDF

TF-IDF is a combination of TF and IDF (Term Frequency and Inverse document frequency). The TF-IDF score indicates the relative importance of a specific term in any dataset[7]. The TF-IDF algorithm is based on word statistics for text feature extraction. TF-IDF is used to vectorize the input [1]. The model considers only the expression of words, that are similar in all texts. The TF-IDF is a commonly used feature extraction technique in text detection. A TF-IDF vector can be generated using different tokens such as words, characters, and n-grams.

- Word TF-IDF: Matrix representation of TF-IDF scores of words
- N-gram TF-IDF: Matrix representation of TF-IDF scores of n-grams, where n-grams are the combination of “n” words
- Char TF-IDF: Matrix representation of TF-IDF scores of character-level ngrams

2.3. Distil BERT

DistilBERT can be defined as a distilled version of BERT in which a compression technique termed as “Knowledge Distillation” is performed on a larger model- BERT to train a smaller model and to reproduce the behaviour of actual BERT model [10]. The actual larger model is termed as “the teacher” and the compact model is termed as “the student” in distillation mechanism. The architecture of DistilBERT is same as that of the transformer architecture, BERT, but to reduce the model size, a smaller number of layers is used. The token type embeddings and pooler are removed from DistilBERT (which BERT uses for the next sentence classification task). The Batch size was also changed from original BERT that led to an increase in performance. DistilBERT has relied on the same training data as that of BERT model. Three training losses was taken into consideration for DistilBERT namely Distillation loss, Masked Language Modelling loss (from the MLM training task) and Cosine embedding loss (to align the directions of the student and teacher hidden states vectors).

Triple losses ensure the DistilBERT model learns properly and has efficient transfer of knowledge. The distilled model has about half the total number of parameters of BERT base and retains 97% of BERT’s performances on language understanding capabilities. The DistilBERT model is 60% faster, and the model size was reduced by 40% when compared to the BERT model, has been constantly faster. The Parameter count of different pretrained language models is

depicted in Figure 2.4. The similarity in performances on various downstream tasks performed by DistilBERT and BERT was also validated. DistilBERT requires only a small computational training budget, while maintaining the flexibility of larger models. The DistilBERT models are small enough to run on platforms such as on mobile devices.

3. LITERATURE REVIEW

This Section provides a review of the existing literature on various text classification methodologies on different domains. Section 3.1 describes an overview of different datasets on which text classification has been performed. The machine learning algorithms implemented for text classification are discussed in Section 3.2. Section 3.3 highlights the works that have used Feature extraction techniques for classification.

3.1. Various Social Datasets

Engaging with the online platforms, people use social networks as a prominent way for expressing their opinion about an issue or presenting their experiences about an experienced product or service from a company. The data posted on these networks make users potentially vulnerable or abusive, which results in cyberbullying. Instagram, Twitter, Youtube are the commonly used social media platforms. The datasets are usually collected by crawling the target social media using its Application Programming Interface (API). The commonly used datasets for cyberbullying detection are described below.

Raisi et al. [11] described Twitter as one of the public-facing social media platforms with high frequency of cyberbullying. To the best of our knowledge, Twitter is the most available source in the field of Natural Language Processing (NLP) for researchers since a large portion of reviewed papers have benefited from Twitter contents. One of the reasons that this social media is popular among researchers to check their proposed algorithm is that registered users can broadcast short posts (280 character per post) which are mostly textual posts providing a direct to the point source of data. Moreover, people can tweet on Twitter in different languages, so datasets for other languages than English may also be achieved through Twitter. Twitter daily use is increasing rapidly. Muneer et al. [12] mentioned that this platform raised many issues due to misunderstanding regarding the concept of freedom of speech meaning the users share their unfiltered opinion even if they have offensive contents. Thus, this platform is considered as a vital data source in the field of cyberbullying detection.

Instagram dataset is a mix modal dataset that contains text, video, and photo at the same time. It seems that Instagram dataset is not suitable for employing NLP techniques, but it is worth mentioning that NLP is not limited to text analysis. However, there are several info-graphic posts which can be analysed using text analysis and image processing. Although there are rules on Instagram for reporting the abusive and harsh posts, the posts' comments are good place for cyberbullying.

The Ask.fm is a question and answering social network where users can ask their burning question, anonymously or publicly. This social network became the largest QA network in the world in 2017 [13]. A subsample of Ask.fm dataset was used for evaluating the weak supervision model. They filtered the dataset by removing anonymous users' question-answers and the posts that contained only "thanks" word. Samghabadi et al. [13] collected a dataset which contained the full history of question-answer pairs for 3K users.

YouTube is an online video sharing social media. Although this social media is a suitable platform for sharing tutorials and informative videos, it is an open environment that each user can share different kinds of video with harsh contents such as racism videos, porn videos, and so on. This makes YouTube as a good source for researchers to evaluate their detection models on. Bruwaene et al. [14] used two datasets for evaluating their model. They chose about 11,000 posts from VISR dataset which is a dataset from SafeToNet application, an application for parents to control their children's account in different social media. This dataset contains randomly chosen posts from six social media including YouTube. It has 7188 posts from total 603,379 posts. A hashtag collection and then crawled YouTube using the list of hashtags to download posts which are related to selected hashtags.

Wikipedia is a well-known, free content, and multilingual encyclopaedia. Volunteers can edit the texts using wiki-based system. The editors can share their opinion and discuss about improvements of articles in an environment named Wikipedia Talk pages. These pages are associated with each article in the form of "Talk: Article's name". Editors post their messages as new thread and other can share their view about the issue. These threads may be a potential environment for cyberbullying between the editors. Existing works used the Wikipedia talk pages dataset which were collected by Wulczyn et al. [15]. The dataset was gathered by processing the public dump of full history of English Wikipedia. The corpus contains 63M comments from talk pages for the articles dating 2004-2015. The labelled dataset has about 14000 comments which is labelled as personal attacks. Gada et al. [16] used the Toxic Comment Classification Challenge dataset which is a Wikipedia comments dataset labelled by human for toxic behaviour. The dataset has around 1.6M rows.

3.2. Cyberbullying Detection Using Machine Learning Algorithms

In this section, the machine learning algorithms which mostly used in cyberbullying detection are reviewed. The recent literature accounts the use of different machine learning and deep learning algorithms for detecting the hate speech, harsh contents including the pornography and abusive languages.

The most popular machine learning in text classification is linear SVM as the most text analysis problems are linearly separable. Moreover, the significant characteristic of SVM is that it can be learnable with any number of features. Thus, as the texts have lots of features, this algorithm is appropriate. choice for their classification problems. Hani et al. [17] compared two supervised machine learning algorithms which are SVM and CNN on two different types of features namely Term Frequency- Inverse Document Frequency (TFIDF) and Sentiment Analysis features. Like other approaches, they aimed to have a machine learning model for detecting the harassments in a text data, so their model followed the three main steps: pre-processing, feature extraction, and classification in which they used Support Vector Machine (SVM) as the machine learning algorithm. Besides the TFIDF features, they used N-Gram as the feature extraction method and for the sentiment features, they used Text Blob Library which is a pre-trained model on movie reviews. The results showed that SVM gets highest accuracy in 4-Gram while NN gets highest accuracy in 3-Gram. However, in average of n-Gram, NN works better than SVM. Kumar Sharma et al. [5] experimented different methods to identify bully content in a text and find the best classifier in this way. Among the four classifiers that they used SVM was the second one in terms of AUC score. Soni et al. [18] instead of doing research on only text data, they implemented an audio-visual-textual cyberbullying detection platform. They used 5 different machine learning algorithms including SVM for detecting cyberbullying in audio, visual, and textual features. The results showed that the proposed approach which applied the machine learning algorithms on multi modal features (Audio+Visual+Textual) compared to applying proposed approach on all comments achieved about 2.75% decrease in F1 score. The lowest F1

score in all features belongs to SVM, which means this algorithm is not suitable for multi modal cyber bullying detection. As a hot topic in this field is detection of bullies in different languages. Leon-Paredes et al. Authors of [19] developed an online prevention tool for detecting cyberbullying in Spanish language. They used three different classifiers based on the characteristic of algorithms namely Naïve Bayes, SVM, and Logistic Regression on three different size of dataset which are small corpus, medium corpus, and large corpus. They measured accuracy, average precision, and F1 score as the evaluation metrics for a total 90 executions. The results showed that the average precision of the detection was between 80% to 91%, however, SVM got the best accuracy of 93% on the medium corpus at the training rate of 10%. In addition, Nurrahmi et al. Authors of [20] proposed a cyberbullying actors detection system based on the reliability analysis of the users for notifying them about their offensive content in Indonesian language. They classified the tweets based on normal behaviour and abnormal behaviour and then used the number of bully and non-bully tweets for each user to calculate the probability of user's behaviour so that they can user this probability in finding the reliability of the user. They categorized the users in four groups based on the probability of their behaviour: if the probability is less than 50% then user is normal, and if the probability is equal and more than 50% then the user lies under bullying actors. Their web-based tool used SVM as one of the two machine learning algorithms and tried it using two techniques, linear and RBF, to recognized whether the dataset is fitted to linear function or non-linear function. The results showed that SVM got higher F1 score than KNN algorithm, and between linear kernel or RBF kernel in SVM, the RBF with C=4 achieved the highest F1 score.

3.3. Feature Extraction on Cyberbullying Detection

The usage of social media platforms such as Facebook, WhatsApp, Twitter, and Instagram had increased over the past years [21]. A huge amount of data is transferred through these platforms among users which also includes obfuscated content and hateful words. The data contributing to cyberbullying could be of different formats such as text, images, and videos. Every dataset is comprised of features, which could be considered as variables. The data analysis, prediction, and classification are dependent on these features. The accuracy of any machine learning algorithm relies upon the features that have been used for training the models. The datasets are expanding with various features in the cyberworld, and this increases the challenge of selecting features for prediction. The quality of a dataset can be improved by optimizing features and hence feature extraction plays a vital role, as it helps in defining complex datasets with a reduced number of features. The feature extraction methods play an important role in improving the accuracy of the different Machine learning algorithms used to identify cyberbullying. The performance of cyberbullying detection using classifiers could be improved by using text-based features instead of non-text-based features such as image and network graph [2].

The different data types contribute different features that are used for cyberbullying prediction. A major classification of features includes content, user, sentiment, and network-based features [5]. Feature extraction methods implemented on any dataset depend on the data type. The content-based features could be further classified as profanity, negativity, and subtlety [22]. Negativity and Profanity seems to appear among most of the cyberbullying instances [23]. Special features could be further used to predict the label that includes Sexuality, Intelligence, and Race.

The most identified cyberbullying involves the usage of text data types irrespective of the social media platforms. Text data types consist of the negative connotation, profane words, context related to minority races, physical characteristics, religion [3]. The textual features help in improving the analysis of cyberbullying content includes the density of inappropriate words, number of special characters such as question mark and exclamation, the density of upper-case letters, number of smileys and part of speech tags. A combination of features was identified to

detect cyberbullying in Youtube comments [11]. Online user-based features, cyberbullying-specific features, content-based features were used to identify cyberbullying in social network videos that include Youtube user comments.

TF-IDF is a commonly used feature extraction technique in text detection. Dinakar et al. In [23], the authors used TF-IDF on multiple machine learning algorithms to compare the accuracy of cyberbullying detection on Form spring and Youtube datasets. The feature extraction method was used in predicting the accuracy of the model generated using SVM on MySpace, Slashdot, Kongregate by Raisi et al. [11]. The accuracy prediction of cyberbullying detection on Turkish language was performed by obtaining TF-IDF properties [8]. As an initial approach to get the baseline model, Gada et al. [16] used TF-IDF on simple classification techniques. Among different feature engineering techniques carried out in the early detection of cyberbullying, the lexical features were weighted using TF-IDF [13]. The TF-IDF vectors generated using different levels of input tokens such as Word TF-IDF, N-gram TF-IDF and Char TF-IDF was used by Chen et al. [7] to compare their HANCD model with baseline models such as KNN, Random Forest, Naive Bayesian, XGBoost and Logistic Regression. Chen et al. [7] identified that TF-IDF vectors was more effective when compared to the pre-trained word embedding technique, Glove. Sharma et al. [5] created a Machine learning model by extracting all the feature vector sets and stacked them to a single feature set. Word and characters were taken as token for TF-IDF feature extraction. Features were extracted using TF-IDF along with sentiment analysis to design the cyberbullying detection model designed by Hani et al. [17]. Analysis of tweets to identify bully and non-bully tweets were performed using TF-IDF vectorization. TF-IDF is a simple and proven method in text classification [7].

DistilBERT pre-trained language model is built by leveraging the knowledge distillation on BERT models. The DistilBERT models are lighter and has a faster inference time. This recently released pre-trained language model is getting popular and researchers are working to exploit its capabilities on various downstream tasks.

Herath et al. [1] developed and evaluated a cyberbullying classification model using DistilBERT and state-of-the-art NLP technology. The dataset collected from Twitter for the SemEval 2019-Task 5 (HatEval) challenge was utilized for the study. The addressed problem in this challenge was to identify cyberbullying against Women and Immigrants. To identify cyberbullying, three classification models, each built on DistilBERT along with a classification layer was developed. The three models were built by changing the ratio of positive and negative classes as explained below:

1. Model A: Training data was imbalanced, and majority class was positive.
2. Model B: Training data was imbalanced, and majority class was negative.
3. Model C: Training data was balanced.

All the three models mentioned above were ensembled using a Simple Voting Classifier to predict the results. This ensemble model achieved a result of 0.41% F1-Score.

Ratnayaka et al. [24] implemented DistilBERT in identifying cyberbullying detection through role modelling. Ask.fm dataset was utilized to categorize the participant roles into victim and harasser, which is a multi-class classification problem. The evaluation of cyberbullying classification was done based on the model developed by Herath et al.[1] as explained above, where in three models where ensembled in which each model was fed with a training dataset in which the majority class was positive, negative, and balanced. The Twitter dataset was used to evaluate this model, in which the tweets were categorized into “Offensive” and “Not Offensive”. This ensemble model achieved an accuracy of 0.906 on F1 score.

4. OUR PROPOSED CYBERBULLYING DETECTION APPROACHES

In the following section, we first present the data processing steps performed. Then in subsection 4.2 we present the modified SVM models. The two SVM models were developed using different tokens of TF-IDF vectors. The proposed DistilBERT-based model is presented in subsection 4.3. The Ensemble models of stacking the base models are explained in detail in subsection 4.4.

4.1. Data Pre-processing

Data pre-processing plays a major role in developing any machine learning model, as the model performance relies on the data input. It is an important step in cleaning the data before feeding them to any model, to avoid any error during training. The NLTK library is commonly used to perform the pre-processing tasks such as tokenization, lemmatization removing stop words and unwanted characters, stemming the raw data. The type of data pre-processing required depends on the task for which the models are developed. The blank rows were removed, and the text case was converted to lower case. This was followed by the tokenization, word-stemming, and lemmatization process. The stop words were not removed in our cyberbullying detection task, because the important indicators for cyberbullying detection could be the second and third nouns.

4.2. Applying Two Different Length of Feature Extraction Tokens on SVM

Two different feature extraction tokens were used to implement the enhanced SVM model. The models differed in terms of TF-IDF vectors fed into the classifier. The SVM Model 1 utilized the TF-IDF for words, and the SVM Model 2 was built using the TF-IDF vectors of Words, Characters, and N-gram.

4.3. Applying Word Embeddings: DistilBERT

The raw data was pre-processed. The processed data saved to a data frame. The data is split into Training Set, which constitutes 80% of the entire dataset and Testing set, that contains 20% of the remaining data. The DistilBERT Tokenizer and DistilBERT model is loaded. The Training dataset is fed to the DistilBERT Tokenizer. The Tokenizer converts the raw data into a format that DistilBERT can process. The DistilBERT Tokenizer performs below actions to prepare the input to the model:

1. Tokenizer transforms the sentence's words into an array of DistilBERT tokens.
2. Adds a special starting token ([CLS] token) to the above generated sequence.
3. Adds the necessary padding to have a unique size for all sentences (we used the maximum length value as 32).

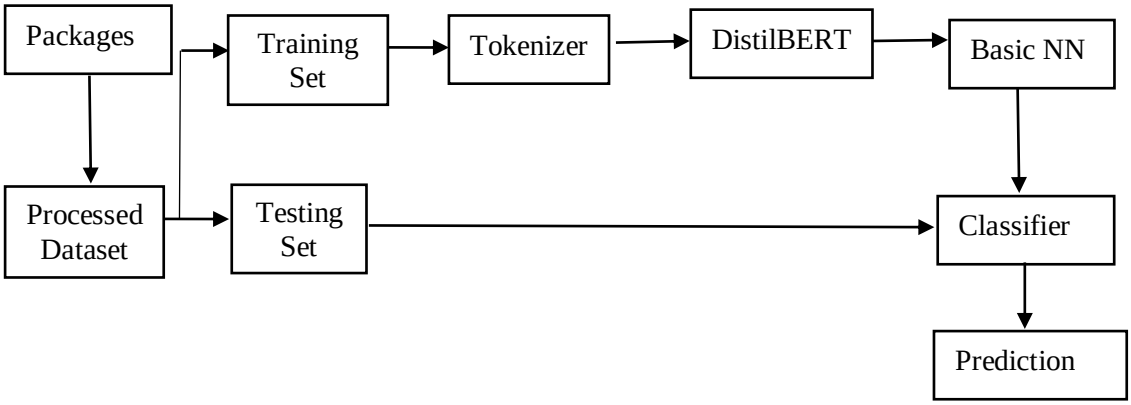

Figure 1. The DistilBERT Model

The output from the DistilBERT Tokenizer contains input IDS, Attention masks and Special Tokens. This is fed to the DistilBERT fine-tuned model. The Trained DistilBERT model was used to generate the sentence embeddings. The output of this model is a vector of length 768 (default length).

To utilize this output from the pre-trained DistilBERT embedding model for cyberbullying detection, a basic neural network architecture with Dense and Dropout layers is implemented. This layer gets the input from the DistilBERT transformer and produces a vector, that is used for predictions in classification tasks. The model was trained for 3 epochs. Adam was used as the optimizer for the model. Since the samples belong to exactly one class, the Sparse Categorical Cross entropy is used to estimate the loss calculation. The block diagram of the DistilBERT Model developed is illustrated in Figure 1.

4.4. Stacked Ensemble Models

We have developed two models that are based on two different approaches: the enhanced SVM is based on the textual features of the data, and the DistilBERT word embeddings is based on the ability of language understanding capabilities of NLP transformers. These heterogeneous models are combined using the Stacking Ensemble method for the classification task. In the ensemble model, a meta-learning classification algorithm is used to combine the predictions from the two base models, SVM model and DistilBERT model. Since stacking model has the ability to exploit the potential of various well-performing models, sit was chosen to make predictions on the cyberbullying detection task, expected to exhibit better performance than the individual base models.

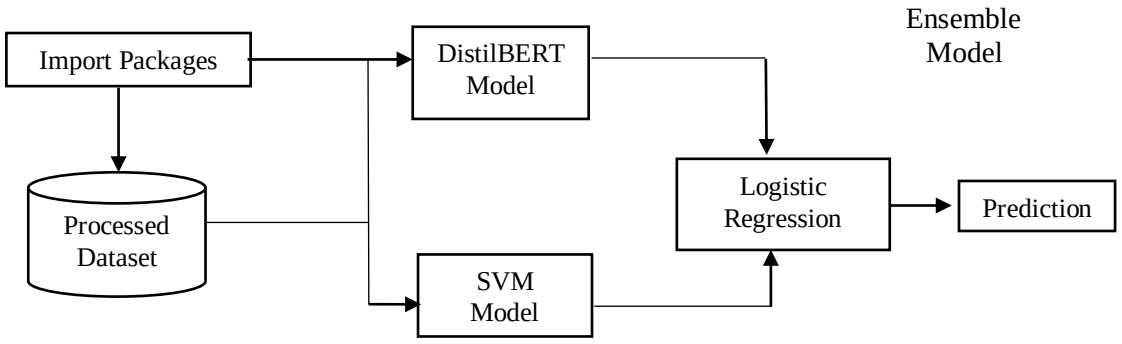

Figure 2. The General Ensemble Model Architecture

Figure 2 represents the general stacked model architecture. Cyber-bullying detection is a binary classification problem, and the input features are independent. Hence, the Logistic Regression model is used as a meta-model for classification of cyber-bullying content.

5. EXPERIMENTAL RESULTS ANALYSIS

5.1. Evaluation Metrics

The potential of any model can be evaluated using few metrics which helps in determining the ability of a model to differentiate texts as cyberbullying or not. To analyse the performance of models, it is important to examine the assessment metrics. The evaluation of models was performed based on various parameters such as Accuracy, Precision, Recall and F1-measure from the confusion matrix. Confusion matrix can be used to measure the performance of any machine learning classification problem.

The Accuracy of a model can be defined as the ratio of the number of correct predictions against the total number of predictions made. The Accuracy can be estimated using below formula.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

The Precision of a model is determined as the proportion of predicted positive cases to the total predicted positives. It helps us to calculate the ratio of relevant data among true positive (TP) and false positive (FP) data belonging to a specific class.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall can be defined as the proportion of Real Positive cases that are correctly Predicted Positive.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-Score is the weighted average of Precision and Recall. F1 Score is calculated using below formula. F1-score helps to combine precision and recall into a single measure.

$$\text{F1Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

5.2. Impact of Feature Extraction on SVM models

Muneer et al. [12] performed a comparative analysis of various machine learning models for cyberbullying detection on twitter dataset. The dataset used was relatively smaller in size when compared to other works and was similar to the smaller dataset used in our work. The work employed the TF-IDF vectorization for feature extraction as applied in this thesis. The SVM model developed by utilizing TF-IDF features exhibited a lower accuracy of 67.13% and a precision of 0.67. These metrics were much lower when compared to our results, where in an accuracy of 85.53% and precision value of 0.86 was achieved. Salminen et al. [25] conducted an analysis of different classifiers using a combined dataset which was extracted from different social media platforms such as Youtube, Twitter, and Wikipedia. The generated dataset had a

class imbalance of 1:4, in which most of the data samples had non-cyberbullying content. Due to the similarity in the generation of dataset and the class imbalance exhibited in the study, a comparison of results was performed using the results obtained from this paper. The study was done using different stand-alone feature extraction methods such as TF-IDF and BOW, word embedding techniques such as Word2Vec and BERT, simple features such as punctuation and use of upper-case characters etc in combination with individual ML algorithms such as LR, NB, SVM, XGBoost etc. F1-score was used as an evaluation measure in this study. SVM model exhibited an F1-score of 64.8% with TF-IDF vectorization, which was clearly less compared to the results obtained in our study, where we obtained an F1-score of 71.48%.

In addition to the above comparisons, due to the similarity in the dataset features and source of extracted data, the performance of SVM model was compared with the machine learning model developed by Sharma et al. [5]. The dataset was extracted from sources such as UCI, Twitter and Kaggle. The extracted dataset was pre-processed and labelled resulting in a final set with columns Date, Comment and Label. This is similar to the dataset used for this work, though we have limited features and our dataset had contents extracted from Twitter.
Two SVM models were implemented using different tokens of extracted TF-IDF vectors to understand the impact of feature selection on traditional SVM models. The initial model was based on the simple TF-IDF word tokens which was fed to the SVM model. The second model was built by using the various tokens of words, characters, and N-grams of TF-IDF vectors into the SVM model.

5.2.1. Evaluating Different Feature Extraction Token Sizes on SVM Models

Different N-gram word tokens were tested on the SVM-TF-IDF model. The N-gram range chosen was between 1 and 7. We identified that, with an increase in the word n-grams, the accuracy was decreasing. The model performed better when the N-gram was set as (1,1) and resulted in an accuracy of 85.53%. The corresponding accuracies of different N-grams are listed in Table 1.

Table 1. Comparison of different word tokens.

Word N-gram	Accuracy (in %)
1	85.53
2	85.28
3	85.22
4	85.39
5	85.33
6	85.37
7	85.13

An analysis was done by changing both the word and character tokens using N-gram. A unigram character token was also used by default in addition to the other two tokens. The word and character tokens were tested for different N-gram values within the range of 1 to 7 to determine the impact of the increase in token size on accuracy. The accuracy was dropping when both the word and character token sizes were increased simultaneously. The comparison of accuracies on different word and character tokens is illustrated in Table 2.

Table 2. Comparison of different word and character tokens.

Word N-gram	Character N-gram	Character Unigram	Accuracy (in %)
2	2	1	87.03
3	3	1	88.02
4	4	1	87.83
5	5	1	87.56
6	6	1	87.26
7	7	1	86.86

Hence, to understand the impact of "Character" tokens, the unigram of word token was considered, and the experiment was performed by changing only the "Character" token sizes. The unigram of character token was used in combination with the word unigram and character N-gram. The results obtained from changing the token sizes are illustrated in Table 3. The best accuracy was achieved when the N-gram was set as (1,5) for the character token. Hence the feature vector for the base model was created using a combination of unigram character, unigram word token, and an N-gram of (1,5) for character tokens.

Table 3. Comparison of different character tokens, word and character unigram.

Word Unigram	Character N-gram	Character Unigram	Accuracy (in %)
1	2	1	88.02
1	3	1	88.04
1	4	1	88.05
1	5	1	88.1
1	6	1	88.03
1	7	1	88.06

The accuracy of the SVM Model 1 achieved was 85.53%. This accuracy was achieved by implementing the TF-IDF vectorization on words. The model exhibited a better Recall and F1 score was achieved for Class 0 data. The accuracy of the SVM Model 2 achieved was 88.1%. This accuracy was achieved by implementing the TF-IDF vectorization on words, character and N-gram tokens. The Evaluation metrics comparison of both SVM models is illustrated in Table 4. Figure 3 represents the Confusion matrix of SVM models 1 and 2 respectively.

Table 4. Comparison of two SVM models.

Model	Parameters			
	Accuracy	Precision	Recall	F1-Score
SVM: TF-IDF of Words	85.53	82.05	63.33	71.48
SVM: TF-IDF of Words, Characters and N-gram	88.1	84.38	71.71	77.53

The increase in accuracy of the SVM model 2 can be attributed towards the combination of different granularity of features. The results also showed a drastic increase in the Recall and F1scores. Thus, combining different tokens prove to perform better on traditional SVM algorithms, than relying on a single feature set.

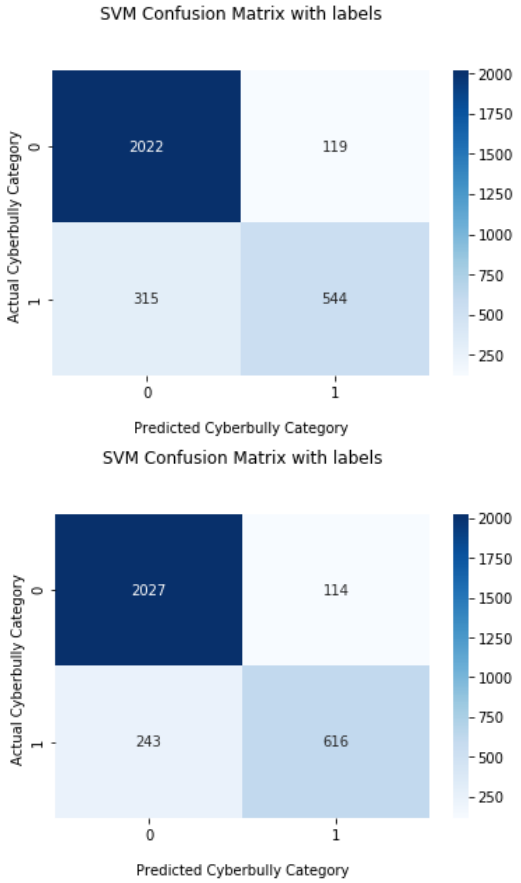

Figure 3. Confusion matrix of SVM models 1 and 2 respectively

5.2.2. Evaluating Different Length of Feature Extraction Tokens on SVM against Existing Work

The analysis of the existing work is done by comparing the developed models based on similarities such as the feature extraction method and ML algorithm.

A summary of the comparison of the related work based on TF-IDF vectors is illustrated in Table 5.

Table 5. Comparison of Related Studies.

Authors	Feature	N-gram	Classifier
Muneer et al. [12]	TF-IDF: Words	Unigram	SVM
Salminen et al. [25]	TF-IDF: Words	Unigram	SVM
Sharma et al. [5]	TF-IDF: Words, Characters & N-gram	(1,5) Character	SVM
In this paper	TF-IDF: Words	Unigram	SVM
In this paper	TF-IDF: Words, Characters & N-gram	(1,5) Character	SVM

For comparing the results with the model developed by Sharma et al. [5], the dataset used for this paper was deployed on their model which we developed based on their work. The model was developed by generating TF-IDF vectors of three types. The TF-IDF vectors of both words and characters as tokens along with an n-gram sequencing from 1 to level 5 was generated. The extracted feature vectors were stacked into a single set. This stacked set of features were divided into training and test data sets. The SVM model trained using the stacked feature set resulted in an accuracy of 88.1%. The results of the SVM Model with TF-IDF word tokens was compared with the model developed by Sharma et al. [5]. The baseline model outperformed in this scenario compared to the SVM Model with TF-IDF word tokens, which was based on only word vectors that was developed in this paper. This increase in performance could be due to the combination of different extracted feature vectors. Due to the high performance of the traditional SVM using different tokens of TFIDF such as Words, Characters and N-gram sequencing, we have chosen this as the base model instead of the simple SVM with TF-IDF for words.

5.3. Comparative Evaluation of DistilBERT Model on Cyberbullying Detection

DistilBERT pre-trained language model developed by Sanh et al. [10] is an emerging word-embedding technique, that uses lesser number of parameters when compared to the existing BERT embeddings. Researchers are implementing DistilBERT in many downstream tasks and few works has focused on using DistilBERT for Cyberbullying detection. The classification model used by Herath et al. [1] to identify cyberbullying against Women and Immigrants uses DistilBERT. Three models which were built on a Training dataset by changing the ratios of the majority classes acts as base models. The final ensemble model was built using a Simple Voting Classifier.

Since the dataset used in this research is comparatively balanced, a simple DistilBERT model was developed, with a classification layer on top. Reducing the number of DistilBERT models reduces the training time. This model provided an accuracy of 87.53% and the highest precision of 91.17%. This model was slightly better when compared to traditional SVM with TF-IDF in terms of accuracy. The Processing time of training DistilBERT model was 30 minutes for the 3 epochs. The DistilBERT exhibited the maximum training time when compared to all other models developed. The better recall and f1 scores were exhibited for class 0 data in DistilBERT models as well. In addition to that, this model outperformed in terms of Precision for bullying content. The Confusion matrix generated for the DistilBERT model is shown in Figure 4.

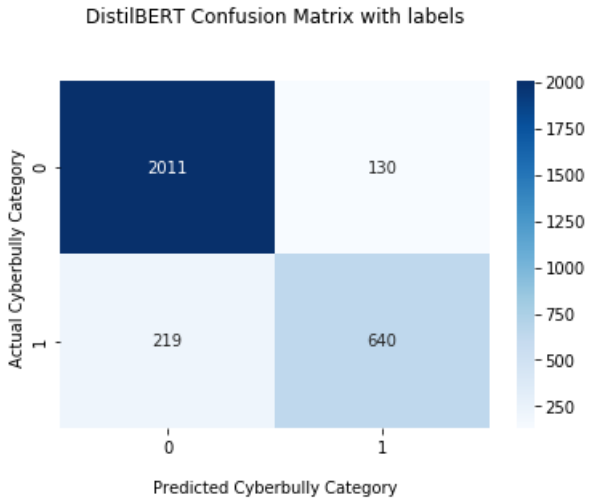

Figure 4. Plot of DistilBERT Confusion Matrix

Figure 5. Training and Validation Loss

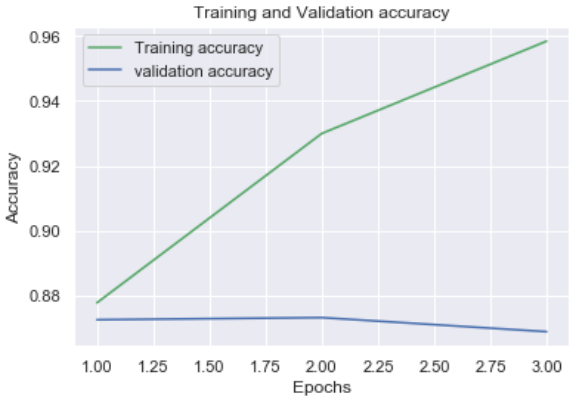

Figure 6. Training and Validation Accuracy

The training and validation loss of three epochs in DistilBERT model is plotted in Figure 5. Figure 6 represents the training and validation accuracy for the three epochs. The training and validation losses drastically reduced with the increase in the number of training epochs. The training accuracy improved significantly over the epochs, however, there was a slight dip in the validation accuracy at the end of the third epoch.

5.4. Proposed Ensemble Models

We developed and analyzed the performance of two ensemble models. The initial ensemble model is using the traditional SVM with the TF-IDF for words and the DistilBERT model. The second ensemble model is developed using the combined feature extraction levels of TF-IDF using different tokens such as word, character, and n-gram on SVM and the DistilBERT model. More details of these two ensemble models are explained below:

5.4.1. Ensemble Model 1: Ensemble Using SVM (TF-IDF for Words) and DistilBERT

This ensemble model is built using SVM and TF-IDF for words along with the DistilBERT model. It yields an accuracy of 88.3%. The Recall and F1 score of this ensemble model were much better while compared to the base SVM and DistilBERT models. The Confusion Matrix of the Stacked ensemble model 1 is shown in below Figure 7. The ensemble model 1 outperformed the base models in terms of evaluation metrics except for precision.

5.4.2. Ensemble Model 2: Ensemble Using SVM (TF-IDF For Words, Characters, and n-gram) And Distilbert

The accuracy of the second ensemble model developed using SVM along with the different tokens of TF-IDF vectorization such as words, characters, n-gram and the DistilBERT model is over 90%. The Confusion Matrix of the Stacked ensemble model 2 is shown in Figure 8. The increase in accuracy is due to the efficient base models. The Base model 1 has taken into consideration the different granularity of TF-IDF vectors and the word embeddings in the DistilBERT model accounted for the better performance.

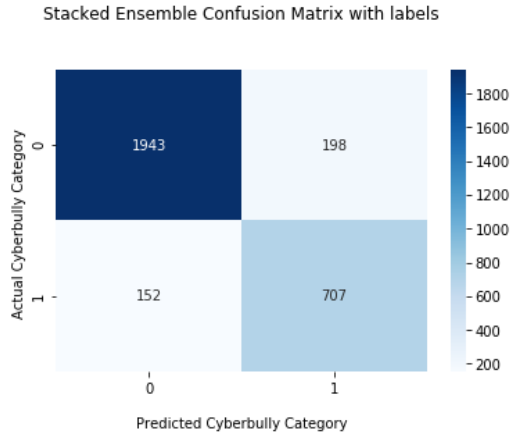

Figure 7. Ensemble model 1

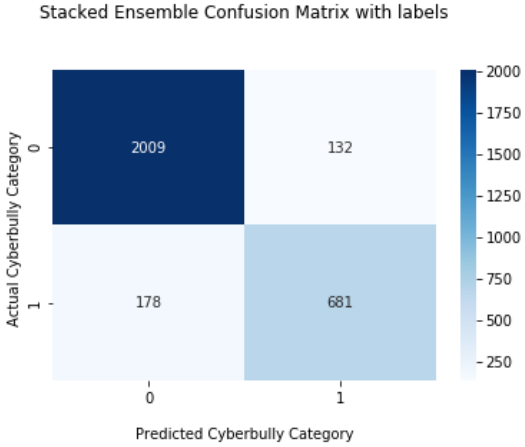

Figure 8. Ensemble model 2

5.5. Performance Comparison of All Models

A summary of the performance of all the five models in terms of various evaluation metrics is represented in Table 6.

Table 6. Model Parameters.

Model	Parameters			
	Accuracy	**Precision**	**Recall**	**F1-Score**
SVM: TF-IDF of Words	85.53	82.05	63.33	71.48
SVM: TF-IDF of Words, Characters and N-gram	88.1	84.38	71.71	77.53
DistilBERT	87.53	91.17	62.51	74.17
Ensemble: Model 1	88.3	78.12	82.30	80.15
Ensemble: Model 2	89.66	83.76	79.27	81.45

DistilBERT model exhibited the best precision when compared to all the other models, however ensemble models outperformed in terms of all other parameters. The highest accuracy of 89.66% is exhibited by the Ensemble model 2 among all other models. This model also yields the best F1-score of 81.45%. Among both SVM models developed, SVM model 2 outperforms the other

in all aspects due to the combined features fed into the model. Thus, with the increase in levels of tokens fed as features, traditional models perform better when compared to models built using a single set of features. The accuracy of this SVM model is very similar to the proposed DistilBERT model, which had inbuilt word embeddings. Thus, the addition of different tokens as features acts as a substitute for the word embeddings, while implemented on smaller datasets.

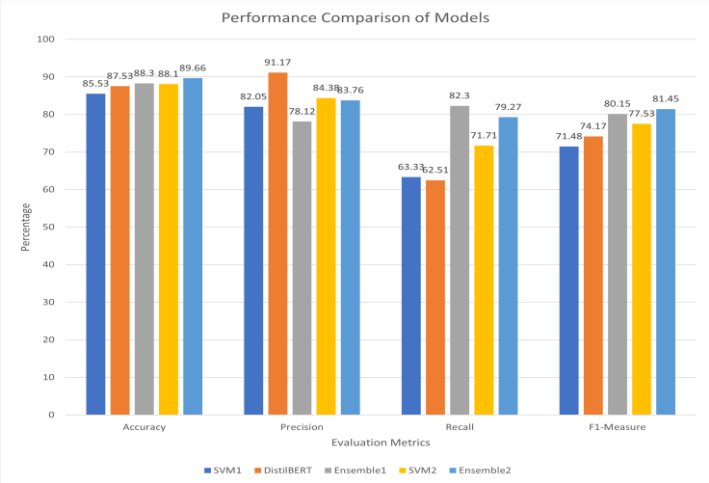

Figure 9. Performance Comparison of all models

The ensemble models can be used to develop systems that can predict the cyberbullying with better accuracy and exhibits better performance than individual base models. The Performance comparison of all the models is illustrated in Figure 9. All the models exhibited better accuracies and had slight improvements while implemented using word embeddings, the increase in features tokens and ensembling the models. We observed fluctuations in the Precision and Recall of various models. The lowest precision was demonstrated by the Ensemble model 1. However, the ensemble model 1 exhibited the highest Recall when compared to other models. Both ensemble models achieved a better F1-score than all the individual base models.

6. CONCLUSION

The impact of cyberbullying is dramatically increasing due to ease of access to Internet. This results in phycological and physical harm to victims. There are several systems available to tackle cyberbullying. This work identifies three different models for cyberbullying detection using a newly generated dataset that was extracted from the Enron email dataset, Twitter parsed data and Youtube parsed data from the Mendeley Cyberbullying dataset. These models were based on traditional machine learning algorithms and recent state-of-the-art word embeddings that consists of a single neural layer on top. We have also introduced an ensemble model using a stacking method for combining two base models which were based on completely different approaches to leverage the performance.

The evaluation of the proposed ensemble models shows good performance in cyberbullying detection. The traditional machine learning models require feature extraction techniques for better performance; however, the DistilBERT word embeddings have inbuilt tokens and do not require any explicit tokenization. The traditional SVM models were based on TF-IDF feature extraction of words and a combined TF-IDF vectors of words, characters, and N-gram. The experiment results indicated that the SVM model with the combined vectors outperformed the simple SVM-TF-IDF model. The DistilBERT exhibited the best precision of 91.17%. The

Stacked ensemble models outperformed the base models in terms of Accuracy, Recall and F1-Score. The Ensemble model using the combined vectors along with SVM and the DistilBERT model had the best accuracy of 89.6%.

REFERENCES

[1] T. Atapattu, M. Herath, G. Zhang, and K. Falkner, "Automated Detection of Cyberbullying Against Women and Immigrants and Cross-domain Adaptability," *arXiv:2012.02565 [cs]*, Dec. 2020, Accessed: Feb. 10, 2022. [Online]. Available: http://arxiv.org/abs/2012.02565

[2] A. K. Goodboy and M. M. Martin, "The personality profile of a cyberbully: Examining the Dark Triad," *Computers in Human Behavior*, vol. 49, pp. 1–4, Aug. 2015, doi: 10.1016/j.chb.2015.02.052.

[3] K. R. Purba, D. Asirvatham, and R. K. Murugesan, "A Study on the Methods to Identify and Classify Cyberbullying in Social Media," in *2018 Fourth International Conference on Advances in Computing, Communication & Automation (ICACCA)*, Subang Jaya, Malaysia, Oct. 2018, pp. 1–6. doi: 10.1109/ICACCAF.2018.8776758.

[4] W. N. Hamiza Wan Ali, M. Mohd, and F. Fauzi, "Cyberbullying Detection: An Overview," in *2018 Cyber Resilience Conference (CRC)*, Putrajaya, Malaysia, Nov. 2018, pp. 1–3. doi: 10.1109/CR.2018.8626869.

[5] H. Kumar Sharma, K. Kshitiz, and Shailendra, "NLP and Machine Learning Techniques for Detecting Insulting Comments on Social Networking Platforms," in *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, Paris, Jun. 2018, pp. 265–272. doi: 10.1109/ICACCE.2018.8441728.

[6] K. R. Talpur, S. S. Yuhaniz, N. N. B. Amir, B. Ali, and N. B. Kamaruddin, "CYBERBULLYING DETECTION: CURRENT TRENDS AND FUTURE DIRECTIONS," . *Vol.*, no. 16, p. 12, 2005.

[7] L. Cheng, R. Guo, Y. Silva, D. Hall, and H. Liu, "Hierarchical Attention Networks for Cyberbullying Detection on the Instagram Social Network," p. 10, 2019.

[8] E. C. Ates, E. Bostanci, and M. S. Güzel, "Comparative Performance of Machine Learning Algorithms in Cyberbullying Detection: Using Turkish Language Preprocessing Techniques," p. 19.

[9] H. Liang, X. Sun, Y. Sun, and Y. Gao, "Text feature extraction based on deep learning: a review," *J Wireless Com Network*, vol. 2017, no. 1, p. 211, Dec. 2017, doi: 10.1186/s13638-017-0993-1.

[10] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv:1910.01108 [cs]*, Feb. 2020, Accessed: Dec. 04, 2021. [Online]. Available: http://arxiv.org/abs/1910.01108

[11] E. Raisi and B. Huang, "Cyberbullying Identification Using Participant-Vocabulary Consistency," *arXiv:1606.08084 [cs, stat]*, Jun. 2016, Accessed: Nov. 14, 2021. [Online]. Available: http://arxiv.org/abs/1606.08084

[12] A. Muneer, "A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection on Twitter," *Future Internet*, vol. 12, no. 11, p. 187, Oct. 2020, doi: 10.3390/fi12110187.

[13] N. S. Samghabadi, A. P. L. Monroy, and T. Solorio, "Detecting Early Signs of Cyberbullying in Social Media," p. 6.

[14] D. Van Bruwaene, Q. Huang, and D. Inkpen, "A multi-platform dataset for detecting cyberbullying in social media," *Lang Resources & Evaluation*, vol. 54, no. 4, pp. 851–874, Dec. 2020, doi: 10.1007/s10579-020-09488-3.

[15] E. Wulczyn, N. Thain, and L. Dixon, "Ex Machina: Personal Attacks Seen at Scale," *arXiv:1610.08914 [cs]*, Feb. 2017, Accessed: Sep. 02, 2021. [Online]. Available: http://arxiv.org/abs/1610.08914

[16] M. Gada, K. Damania, and S. Sankhe, "Cyberbullying Detection using LSTM-CNN architecture and its applications," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, Jan. 2021, pp. 1–6. doi: 10.1109/ICCCI50826.2021.9402412.

[17] J. Hani, M. Nashaat, M. Ahmed, Z. Emad, E. Amer, and A. Mohammed, "Social Media Cyberbullying Detection using Machine Learning," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 5, 2019, doi: 10.14569/IJACSA.2019.0100587.

[18] D. Soni and V. K. Singh, "See No Evil, Hear No Evil: Audio-Visual-Textual Cyberbullying Detection," *Proc. ACM Hum.-Comput. Interact.*, vol. 2, no. CSCW, pp. 1–26, Nov. 2018, doi: 10.1145/3274433.

[19] G. A. Leon-Paredes *et al.*, "Presumptive Detection of Cyberbullying on Twitter through Natural Language Processing and Machine Learning in the Spanish Language," in *2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, Valparaiso, Chile, Nov. 2019, pp. 1–7. doi: 10.1109/CHILECON47746.2019.8987684.

[20] H. Nurrahmi and D. Nurjanah, "Indonesian Twitter Cyberbullying Detection using Text Classification and User Credibility," in *2018 International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, Mar. 2018, pp. 543–548. doi: 10.1109/ICOIACT.2018.8350758.

[21] V. Krithika and V. Priya, "A Detailed Survey On Cyberbullying in Social Networks," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, Vellore, India, Feb. 2020, pp. 1–10. doi: 10.1109/ic-ETITE47903.2020.031.

[22] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, Dec. 2014, doi: 10.1016/j.asej.2014.04.011.

[23] K. Dinakar, B. Jones, C. Havasi, H. Lieberman, and R. Picard, "Common Sense Reasoning for Detection, Prevention, and Mitigation of Cyberbullying," *ACM Trans. Interact. Intell. Syst.*, vol. 2, no. 3, pp. 1–30, Sep. 2012, doi: 10.1145/2362394.2362400.

[24] G. Rathnayake, T. Atapattu, M. Herath, G. Zhang, and K. Falkner, "Enhancing the Identification of Cyberbullying through Participant Roles," in *Proceedings of the Fourth Workshop on Online Abuse and Harms*, Online, 2020, pp. 89–94. doi: 10.18653/v1/2020.alw-1.11.

[25] J. Salminen, M. Hopf, S. A. Chowdhury, S. Jung, H. Almerekhi, and B. J. Jansen, "Developing an online hate classifier for multiple social media platforms," *Hum. Cent. Comput. Inf. Sci.*, vol. 10, no. 1, p. 1, Dec. 2020, doi: 10.1186/s13673-019-0205-6.

AUTHORS

Ms. Saranyanath is currently pursuing Masters in Computer Science at Carleton University. She holds a Bachelor of Electronics and Communication Engineering degree from Anna University India. She has 7 years of experience in Software Industry as Project Manager, Software Consultant and Business analyst. She specializes in Machine learning, Pattern recognition and Data analysis.

Dr Wei Shi is a Professor in the School of Information Technology, cross-appointed to the Department of Systems and Computer Engineering in the Faculty of Engineering & Design at Carleton University. She specializes in algorithm design and analysis in distributed environments such as Data Centres, Clouds, Mobile Agents and Actuator systems and Wireless Sensor Networks. She has also been conducting research in data privacy and Big Data analytics. She holds a Bachelor of Computer Engineering from Harbin Institute of Technology in China and received her Master's and Ph.D. in Computer Science from Carleton University in Ottawa, Canada. Dr Shi is also a Professional Engineer licensed in Ontario, Canada.

Dr. Corriveau received his Master's in Computer Science from University of Ottawa in 1984. During that time, he also worked at Nortel developing an industrial code generator. In 1986, after starting his Ph.D. at University of Toronto, he returned to Nortel becoming a founding member of the TELOS project. This tool spawn off as the ObjecTime start-up in early 1991 and eventually evolved into ROSE Real-Time, and now IBM Rational Software Architect Realtime Edition. Dr. Corriveau completed his Ph.D. in Natural Language Processing in 1991 and soon after joined the School of Computer Science at Carleton.