Spring 5-25-2021

Cyberbullying Classification based on Social Network Analysis

Anqi Wang

Cyberbullying Classification based on Social Network Analysis

A Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Anqi Wang

May 2021

The Designated Project Committee Approves the Project Titled

Cyberbullying Classification based on Social Network Analysis

by

Anqi Wang

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2021

Katerina Potika	Department of Computer Science
Mike Wu	Department of Computer Science
Robert Chun	Department of Computer Science

ABSTRACT

Cyberbullying Classification based on Social Network Analysis

by Anqi Wang

With the popularity of social media platforms such as Facebook, Twitter, and Instagram, people widely share their opinions and comments over the Internet. Extensive use of social media has also caused a lot of problems. A representative problem is Cyberbullying, which is a serious social problem, mostly among teenagers. Cyberbullying occurs when a social media user posts aggressive words or phrases to harass other users, and that leads to negatively affects on their mental and social well-being. Additionally, it may ruin the reputation of that media. We are considering the problem of detecting posts that are aggressive. Moreover, we try to detect Cyberbullies.

In this research, we study Cyberbullying as a classification problem by combining text mining techniques, and the graph of the social network relationships based on a dataset from Twitter. We create an new dataset that has more information for each tweet (post). We improve the classification accuracy by considering the additional social network features based on the user's follower list and retweet information.

Keywords: Social Network Graph, Natural Language Processing, Supervised Learning, Neural Network Model

ACKNOWLEDGMENTS

I would like to appreciate my advisor Dr.Katerina Potika for her patient guidance and support. She provided me many helpful references resources and gave me valuable suggestions for my research. Her guidance and help allowed me to complete the project smoothly. I also appreciate my committee members Dr.Mike Wu and Dr.Robert Chun for their time and valuable feedback. I would also thank San Jose State University for giving me the opportunity to gain useful knowledge.

TABLE OF CONTENTS

CHAPTER

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Cyberbullying is a kind of bullying that uses electronic methods like mobile phones or the Internet to harass and send rudely, offensive, insulting, and hateful messages to hurt other persons. The aggressive words and mean comments will be posted on social media platforms and have a negative influence on recipients' psychology, emotion, work, and study. Cyberbullying is a new kind of bullying that has many different characteristics from traditional bullying. Compared with traditional bullying, Cyberbullying can happen without the restriction of time and place. There may be more bystanders and more serious harassment because Cyberbullying can be anonymous. The development of the Internet and social media platforms such as Facebook, Instagram, and Twitter have increased Cyberbullying and made it common among teenagers. This aggressive behaviour has an impact on the victim's psychology and life, and it can also be imitated by teenagers or other people in the group. Figure

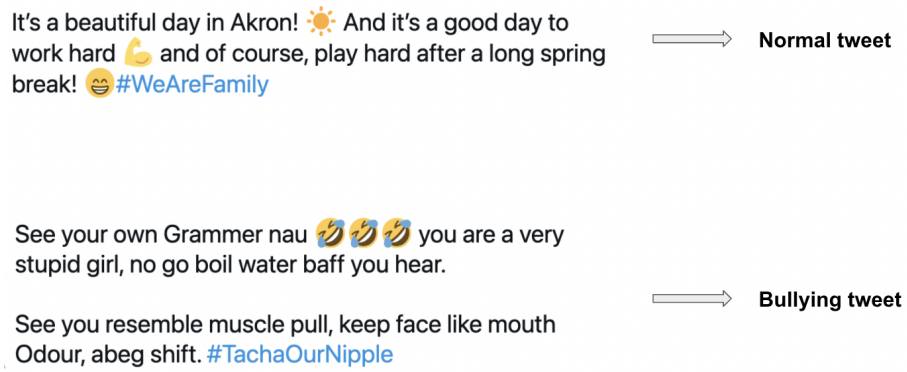

Figure 1: Example of Normal Tweet and Bullying Tweet

1 shows an example of a normal tweet and a bullying tweet. These aggressive words are hurtful and the recipient may also reply aggressive sentences, which has a bad effect on the online communication environment. Based on survey [1], more than 17%

of young people have been bullied on the Internet. Not only children and teenagers can become victims but adults also. Some people are more likely to be bullied such as minorities, women, or low-income individuals. It is important to detect Cyberbullying accurately and efficiently to stop more people from getting hurt and to create a healthy Internet environment.

Cyberbullying can be obvious and obscure. Obvious Cyberbullying contains aggressive words in messages. For obvious Cyberbullying, it can be detected by using text analysis to analyze the features of contents and words. Obscure Cyberbullying doesn't always contain obviously aggressive words but it is still hurtful. The obscure Cyberbullying message with obscure hurtful words may be spread among a group of users quickly. It is harder to detect since we can not accurately determine whether it is Cyberbullying or not, based on just the textual features. For example, sending a message with the word ''pig'' to another user may be considered Cyberbullying, because it is associated with obesity. However, in general the word ''pig'' is used for describing a type of animal. Because of the complexity of Cyberbullying, textual features may not be enough for classifying posts as bullying or normal ones, and more aspects should be considered. According to [2], some social network features, like strong ties, are indicative of having a close relationship with Cyberbullying detection. Therefore, it is reasonable to consider various social network features based on the complexity of user relationships, in order to interpret the multiple semantically meanings of words and phrases.

The detection of Cyberbullying can be considered as a classification problem. An online post can be classified as a bullying post or normal post. Currently, most solutions involve the use of text mining techniques. Many studies revise classification models by applying different machine learning methods to better detect Cyberbullying and improve performance. For example, the support vector machine(SVM) is a popular

classification model for training labeled data. However, Cyberbullying classification is more complicated than a text classification problem since it is closely related to the user and the user's community behavior. The classification accuracy can be largely affected by the obscure bullying posts if we only consider textual features. Recently, social network analysis tools are incorporated in order to consider the underlying network structure of users and their relationships. Our task aims to better classify bullying posts from the Twitter platform and improve the performance of various classification models. By exploring the relationship between the social network community structure and bullying behaviors, and constructing a graph that models this relationship, we propose a new method to classify Cyberbullying. To represent the social network relationships, a community graph is constructed based on the follower connections between users. We start by creating a new dataset based on the baseline dataset in [3], that has more information for each tweet (post): the user IDs, tweet content, retweets user list, list of followers, and the class . Additionally, we utilize retweet information to represent the strength between each users that is reflected on the graph as a weight between nodes (users).

In this work, we study the Cyberbullying classification problem based on textual feature and social network relationship features. Figure 2 shows the workflow of our project. The first phase focuses on the textual features. In order to analyze the textual features of the post, we utilize machine learning models, as well as other text analysis methods such as CNN and LSTM, which are popular neural network algorithms in the text analysis area. The second approach analyzes social network relationships and are added as a key feature. We construct a graph to represent relationships between users based on the follower and retweet information and is added as an extra feature. Additionally, we propose an algorithm that classifies not only posts but also users based on the social network relationship graph. This report is arranged

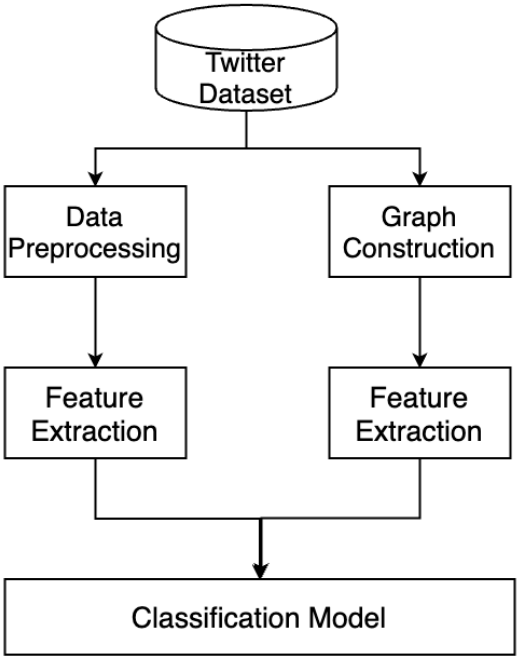

Figure 2: Workflow of Cyberbullying Classification

as follows. Chapter 2 provides the definition of the problem and the motivation behind this topic. It also describes the objective of our project. In Chapter 3, we give some definitions and classification models used in our project. Chapter 4 introduces some background and previous work that related to the Cyberbullying classification problem. In Chapter 5, we present our methods and the experiment process and the classification models. We discuss how to retrieve data from the Twitter platform and the features of dataset. Additionally, we detail the evaluation process and the results. Chapter 6 concludes and covers the hypotheses related to the cyberbullying detection problem as well as part of some future research.

CHAPTER 2

Motivation and Problem Definition

Cyberbullying is a social behavior that has many unique features. It can be anonymous and spread quickly among a group of social network users. A common way to detect cyberbullying is to analyze the textual features of the messages Since many of the cyberbullying messages contain aggressive words or mean words. However, not all these aggressive words represent cyberbullying and some cyberbullying may not contain obvious aggressive words. It will be inaccurate if we only depend on textual features when classifying complex messages. For example, "such a big pig" may have different meanings in different scenarios. It may describe a big pig on the farm and it is a normal comment, or it may describe someone who is overweight and this will be considered as cyberbullying. Hence, only depending on the textual features is not enough. In our project, We assume that cyberbullying is related to how aggressive the social network group is. The classification will base on not only textual features but also social network relationship features such as users' community and retweet history.

Twitter, Facebook, Youtube are popular social media platforms that may generate thousands of aggressive words every day. Our project focuses on Twitter dataset and expands the textual tweet dataset by adding more social network related features. The existing dataset contains two items, one is the Tweet id and the other one is the category of each Tweet, which identifies whether a tweet is bullying or not. Dataset contains cyberbullying Twitter content and normal twitter content. Our project will expand this dataset by adding retweet users' information, users' followers and tweet content. For example, for each tweet, we will collect a retweet user list for those who retweet the specific tweet and the sender's follower information. Besides, A social network graph G = (V, E) will be constructed based on the community information.

The vertex represents users and the edge between two vertices means the connection between two users. The weight of edge means how closeness between two users and it can be calculated based on the interaction times between two users. The vertex contains a value that represents how aggressive the user is. If the user posts many cyberbullying messages or retweets many cyberbullying messages, he may tend to be a bullying user. This aggressive value is calculated based on the Twitter content posted by this user. Given this social network graph and textual content of Twitter, we can classify a Twitter message into cyberbullying or non-cyberbullying by applying text analysis and graph analysis methods. The objective of this project is as follows:

1. Construct a social network graph to represent the relationship between users as part of the dataset

2. Analysis textual features of cyberbullying message and use these features to classify cyberbullying by applying different machine learning models and Analysis retweet information of training data and combine social network graph as part of the classification process

3. Analysis the social network relationship features and use this feature to classify bullying users

4. Evaluate the accuracy of the proposed solution and figure out the importance of social network features

CHAPTER 3

Definitions and Techniques

3.1 Twitter API

Twitter [4] is a social media platform where users can post their thoughts and comments. Twitter provides APIs for developers to collect tweet and comments on Twitter platform for analysis. It also has different category for different purpose and cases. Twitter APIs support multiple programming languages, which brings convenience for developers. By utilizing Twitter API, developers can retrieve information from public conversations to understand what's happening among a group of people and analyze features of various conversations. They can also choose to collect Tweet contents by different time period or real time. The user's information and activities can also be retrieved by using user's ID or tweet ID that posted by the user.

Tweets Object is not only refer to tweet content, but also contains many other information, such as the user who posted this tweet, the time it is posted and other tags it contains. User object is also important since it contains user features such as the followers of the user, retweet information, and the other users that this user is following. For each user, they will have their own profile that can represent his personal preference and social activity preference on online social network platform. This information is helpful when analyzing social network relationships among Twitter platforms and other text analysis problems. Here are some useful APIs to retrieve tweets and retweet information.

- GET statuses/show/id will return the content of tweets with specific id and its author.
- GET statuses/retweets/id will return retweet information of the specific tweet, such as user information of who retweets the specific tweet.
- GET followers/ids will return a list of follower IDs of a specific user.

- GET statuses/user_timeline will return a list of most recent tweets posted during the time periods.

Twitter platform has provided different kinds of APIs in various areas. It is a powerful tool for developers who want to do research based on Twitter platform. For our project, since the dataset resource is from Twitter platform, we can use Twitter API to retrieve data with more features, such as user's name, user's followers, and tweet content. All in all, in order to study the effect of social network activities on cyberbullying behaviour, it's an efficient way to use Twitter API to retrieve useful information related to online social network activities and create different data sets to analyze the problem.

3.2 Graph Terminology

A Graph $G = (V, E)$ consists of nodes and edges and it is used to describe relationships among a set of items. The items are called Nodes in a graph and it is represented by V. The relationship is called an edge and is represented by E, which connects two nodes in a graph. If two nodes are connected by an edge, these two nodes are called neighbors. Neighbors indicate that two nodes have a relationship under the assumption.

A graph can be categorized as directed graph and undirected graph. If edges in a graph have an orientation, the graph is a directed graph; if edges in a graph have no orientation, the graph is an undirected graph. A directed graph may represent more information about the relationships between two nodes. The direction is always represent by an arrow. For a directed graph, the in-degree of a node means the number of edges incoming to a node. Out-degree of a node means the number of edges leaving a node.

A graph can be represented in several ways. One way to represent a graph is by

using a $n \times n$ adjacency matrix A. n is the number of nodes in a graph. If there is an edge between node i and node j, then the value of $A[i][j]$ will be 1. Otherwise, the value of $A[i][j]$ will be 0. It is efficient when the size of the graph is small and the graph is dense. The other common way is by using an adjacency list. For each node v, there will be a list that stores nodes adjacent to v. It is efficient when the size of the graph is large and the graph is sparse. In python language, there is a library called Networkx, which can also represent a graph. It will be more convenience to convert the node and edge information to graph by use networkx library.

3.3 Neural Network Models

3.3.1 LSTM

Long short-term memory(LSTM) is a special Recurrent neural network(RNN) architecture that is used to process long-range dependencies [5]. As Figure 3 [5]shows, For each state in LSTM architecture, there are two lines coming in and going out. One of the lines stands for hidden state and the other line works as gradient during backpropagation. So, the difference between LSTM and a general RNN is that LSTM has feedback connections that means it has two transmission states [6].

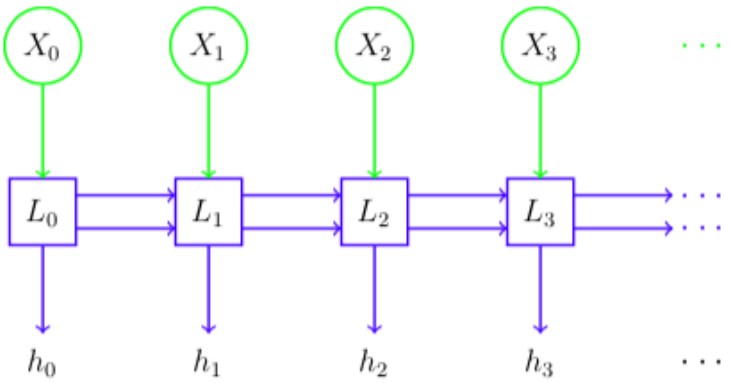

Figure 3: Example: LSTM

LSTM architecture consists of cells, information flow, and gates. As Figure 4 [5]

shows, each cell in LSTM architecture has an input gate, forget gate, output gate and intermediate gate. It also has sigmoid function and hyperbolic tangent function to compute gate vectors and work as activation functions. Cells are used to study the dependencies among elements in the input sequence. These gates are used to control the information flow going in and out of the cell. During the training phase, the weights of connections between cells will be calculated.

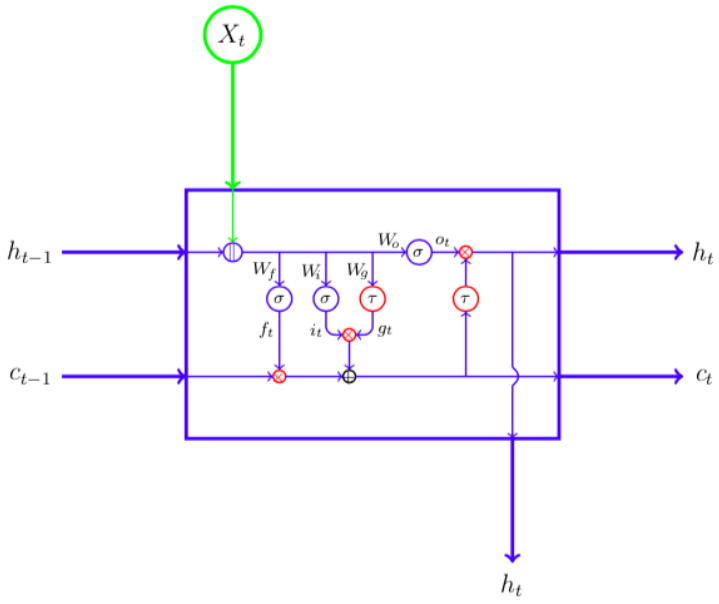

Figure 4: Example: One timestamp of an LSTM

LSTM is a very successful learning model since it can deal with both the individual data points and entire data series. LSTM has been used in different areas to solve various research problems, such as classification problems and prediction problems. It also has been applied in many areas, For example, Sentiment Analysis, Speech recognition, and HardWriting recognition [6]. Some popular products like Google Translate, Apple's Siri, and Amazon Alexa also use LSTM as part of their analysis model [5].

3.3.2 CNN

Convolutional neural network(CNN) is a neural network model that uses convolution functions on its layers. Unlike artificial neural networks(ANNs) that use fully connected layers, CNN also has layers that are not fully connected. Compared with ANNs, which all contains fully connected layers, One of the advantages of CNN is that the convolutional layers can be trained more efficiently and can help avoid overfitting problems. The architecture of CNN contains an input layer, an output layer, and multiple hidden layers. And the types of layers are convolutional layers, non-linearity layers, pooling layers, and fully connected layers [7]. Normally, Fully connected layer is the output layer. Input data of CNN is modified by filters and multiple filters will learn the features of input data. The training phase of CNN can be viewed as the phase to decide filters based on input data [8].

The model is designed to process local structures. For the problems that the local structures are important, CNN will be a good model and perform better. For example, the local features are important for image processing, Natural language processing as well as speech recognition. CNN is widely used in these areas and achieves high accuracy. The other applications contain video analysis, sentimental analysis as well as time series forecasting [8].

3.4 Supervised Machine Learning Models

3.4.1 SVM

Support vector machine(SVM) is a supervised machine learning model that is based on a labeled training dataset. Unlike HMM and PCA that generate scores to do classification, SVM can generate classifications based on data label directly. So, a set of classifier scores can be applied on SVM to achieve a better classification [9]. SVM is used for binary classification and labels can be represented as 0 and 1, or -1 and $+1$. The data points in SVM can be seen as vectors, the goal of SVM is

to find a hyperplane to divide the dataset into two categories and the hyperplane should maximize the distance between the nearest data points on each side of the hyperplane. So, Training phase in SVM is to find the maximum margin hyperplane and the hyperplane is one dimensional less than the space. As the Figure 5 shows, for two dimensional space, the hyperplane is a line that maximizes the margin.

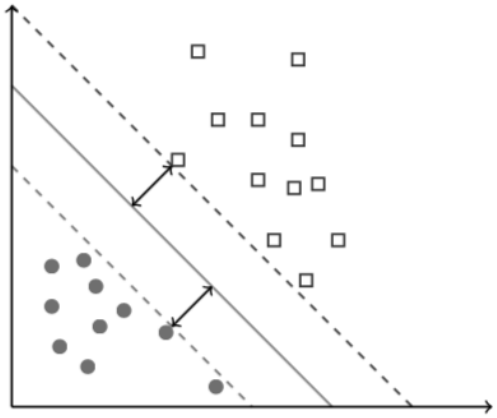

Figure 5: Example: Maximum-margin hyperplane .

The classifier can be both linear and nonlinear, it depends on the choice of kernel function. The kennel function can be linear, polynomial, Gaussian Radial Basis function and Hyperbolic tangent. In SVM, soft margin and feature space are used to process data that is not linearly separable [9]. There are several advantages of SVM. As [9] mentions, The global maximum will be obtained when training SVM. And the kernel trick is very strengthful. SVM can be applied in various areas of classification problems and it is very popular.

3.4.2 Random Forest

Random Forest consists of multiple decision trees and is used in regression and classification problems. In the decision tree, the leaves contain the final decision. In decision trees, the number of nodes in each level depends on the number of

selected features of the training data set. The idea of decision trees is that the tree construction is based on information gain. However, decision trees are NP complete and have overfitting problems. Random forest can help reduce the overfitting problem. Bagging(bootstrap aggregating) is an important idea in Random forest. Random forest bagging observations and features. In this way, the random forest will choose various training data and features to construct multiple decision trees. Each tree is constructed based on a random selected data sample. To get a better node split, it randomly selects features. Then it will achieve the final classification based on the output results from all decision trees. As [9] mentions, The advantage of random forest is that it has less overfitting problem, however, it is more complex to build.

3.4.3 AdaBoosting

AdaBoosting is a machine learning model that can be used in classification problem. It combines other weak classifiers or machine learning algorithms together to achieve a better performance than individual classifier. The idea of this model is that it considers the biggest weakness of the whole algorithm in each iteration and chooses a classifier that can improve the weakness in each iteration [10]. It uses greedy idea in each step to improve the performance for whole process and it is an adaptive model. AdaBoosting model has better performance among the boosting family and is widely used in classification problem. For AdaBoosting model, in order to decide the most appropriate classification algorithm in each iteration, it uses exponential loss function to calculate the loss score. For AdaBoosting model, we want a smaller loss and higher score to have a better performance [10]. All in all, the beauty of AdaBoosting is that it uses many weak classification algorithms to make a better classification and if the number of weak classification algorithms is larger, it would get a better result.

3.5 Feature Extraction Techniques

3.5.1 Word2vec

In Natural Language Processing area, word is the smallest granular. In order to better analyze the word and its relationship, it is common to convert the text to a better analysis format, such as a list of number. Word embedding helps us map the word to mathematical space. Word2vec is one of the ways to show word embedding by using neural network and it is used to analyze the sentiment behind the sentence as well as semantic inference. After the three layers neural network training model, Word2vec maps each word to into a K-dimensional real number vector. Then, we can select three dimensions of the vector and put them into the coordinate system and analyze the position of points of each word. If two words have similar semantic meaning, they will be close to each other. So, the vectors can reflect the relationships among the words as well as the features of the words. There are mainly two models to implement Word2vec. As the Figure 6 shows, The first model is Skip-gram that

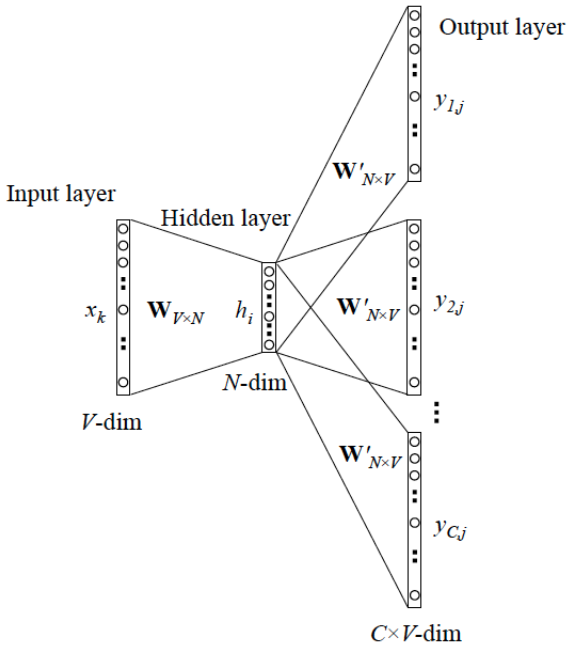

Figure 6: Illustration of Skip-gram model

predicts the context based on the current word. The input is a word and the output is the predicted context that contains more than one word. Figure 7 shows the second

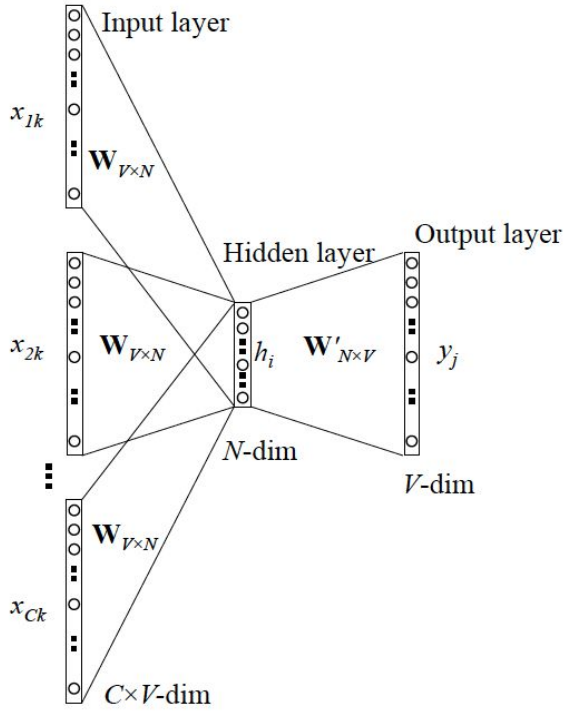

Figure 7: Illustration of CBOW model

model named CBOW that predicts the word based on context. The input is context which contains some words and the output is the predicted word. In our experiment, we implemented Word2vec by using gensim.models.word2vec from Gensim library with our own tweet text corpus. There are many parameters may affect the performance of word2vec model such as 'window', 'min_count' and 'size'.

3.5.2 TF-IDF

TF-IDF(Term Frequency-Inverse Document Frequency) is a statistical method that can evaluate the importance of a word to the document or context in the corpus. The importance of a word increases when it appears more times in the document, however, at the same time the importance of a word may decrease if the frequency of

the word's appearance increases in the corpus. It means that the more a word appears in an article or a document and the less it appears in all other documents or articles, the more it can represent the article or document as a feature.

3.5.3 Node2vec

Network Structure is widely used to represent complex relationships, especially in graph related problems. In order to better analyze the characters of the network or graph structure, graph embedding is very popular to convert the nodes of the graph to a lower dimensional vector. The vector can reflect the structure or characteristics of the original graph. For example, if two nodes have similar structure, their transformed vectors should be similar. Node2vec can be seen as a extension of Deepwalk, which combines Breadth-first search and Depth-first search. The special of Node2vec is that it uses a biased random walk procedure to explore the surrounding nodes [11]. Figure 8 shows the way how random walk works in node2vec. As the paper [11] points out,

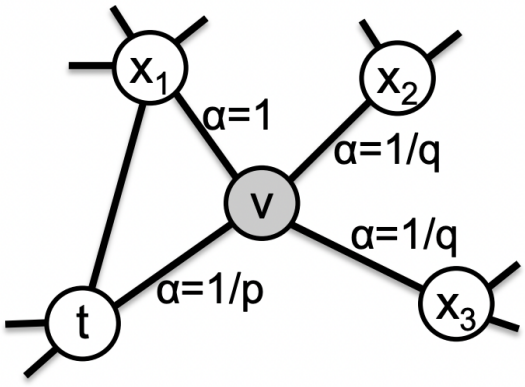

Figure 8: Random walk procedure for node2vec

when going from node t to node v, it will also consider the search bias for the next step. Since random walk combines the idea of Breadth-first search and Depth-first search, the time and space complexity is optimized and the efficiency is also improved. Node2vec is widely used in classification problems.

3.6 Experiment Evaluation Techniques

- Accuracy: is the most common metric to evaluate the performance of the model and it is easy to calculate.

$$\text{Accuracy} = \frac{\text{Number of correct prediction}}{\text{Number of total prediction}}$$

In our classification problem,

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

where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives (see [12]).

- F1 score: is used to evaluate the accuracy of binary classification problem. Recall reflects the how well the classification model can recognize positive samples. The higher recall score, the better it can recognize positive samples. Precision reflects the ability of classification model to identify negative samples. F1-score is a combination of the two scores. The higher the F1-score, the more robust the classification model.

$$F1 - Score = \frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}}$$

- ROC curve: we can calculate the value of AUC, which is the area under the ROC curveand. It is a better way to evaulate the performance and reduce imbalanced accuracy problem. ROC curve uses TPR(True Positive Rate), which is sensitivity and FPR(False Positive Rate), which equals to 1-specificity. The ideal is TPR=1, FPR=0 point, so if the the ROC curve is closer to the point (0,1) and more deviates from the 45-degree diagonal, then result is better. In other words, the larger sensitivity and specificity, the better.

CHAPTER 4

Related Work

4.1 Text Analysis Approaches

Text analysis is the main part of cyberbullying classification by analyzing the textual features of the posted messages. Machine learning models and neural network analysis are baseline approaches in text analysis. K. D. Gorro et al. [13] introduce a machine learning technique called Support Vector Machines (SVM) model to classify data by using defined cyberbullying tags and comparing data with aggressive word libraries. And K. D. Gorro et al. [14] talk about how to pre-process data and rules used in converting data into tokens for SVM analysis. The idea of the SVM is that it depends less on features and it extracts features of each label while training data. So, the SVM can not only be used in common languages, such as English, Chinese, and others but also works well in analyzing the untypical languages. V. Singh et al. [14] define different levels of cyberbullying based on how aggressive the sentences and messages are. They utilize the CNN model as the baseline for their implementation. V. Singh et al. [14] compare the performance of CNN with Decision Tree, Support Vector Machine(SVM), and Long-short Term Memory(LSTM) network. The results show that their CNN model has a better performance based on their data set.

AS. Sadiq et al. [15] introduce a deep neural model to detect aggressive words and messages. Most of the cyberbullying datasets are from Twitter and the length of each message and Tweet may be short, which makes it difficult to accurately understand the meaning of the sentence and determine whether a Tweet cyberbullying or not. They talk about how to preprocess the dataset and use TF-IDF to evaluate the importance of each word as part of a textual feature. S. Sadiq et al. [15] propose a DNN model that trains the dense layers with more important features to select textual features and uses a multilayer perceptron as a learning model to classify the training data.

Their approach is focused on how to improve the classification accuracy based on the short-length messages. They also evaluate their approaches by comparing them to the other CNN models and the results show that they achieve higher accuracy. The researches discussed above suggest that the machine learning model and neural network methods are widely used in cyberbullying classification. And the accuracy of each model related to the textual features of the dataset and overall neural network has better performance.

Cyberbullying has been a serious social behaviour problem and the detection of cyberbullying has attracted much attention. To analyze the cyberbullying behaviours, various datasets from different resources have been studied. Marc-Andre.L et al. in [16] study the relationship between the performance of different classification models in different dataset. The goal of the paper is to find a model that can fit various types of cyberbullying data set and detect cyberbullying correctly. In [16], they propose five ensemble models that combine existing classifiers in different ways and study the performance of the model. The result shows that for the same model that performs well in one dataset may not perform well in the other. The similarity of the neutral-meaning words in different dataset is low and it makes ensemble more difficult. They also give a conclusion that the better way to find a classifier that can fit for all dataset is to merge different datasets as a single training dataset during the training phase.

4.2 Social Network Analysis Approaches

The main parts of social network analysis techniques are data mining and social network analysis. Social network analysis can help decide how to select and extract features of the data, which is the first important step in cyberbullying detection. Social network features such as the closeness between users, how aggressive the social

network is may also affect the cyberbullying classification result. A. Squicciarini et al. [17] study the pairwise interactions between users who spread the bullying message and dynamics of cyberbullying. They learn how users correspond to the cyberbullying message and learn relationships between those users by constructing social network graphs. The social networks contain the influence of user's posted comments, the interaction between users, and the types of comments, such as cyberbullying or normal. They revise the structure of social network graphs and conclude that the demographics of users, their comment habits, comments content, and location in social networks are related to cyberbullying classification. M.Dadvar et al. [18] assume that users' information such as user's gender, users' age, and users' education may affect the accuracy of cyberbullying classification. Their experiments focus on the user's gender and they apply support vector machine models to train gender-specific classifiers. Based on the results, they conclude that the accuracy has been improved if considering gender information.

Similarly, Qianjia Huang et al. [19] study the connection between cyberbullying and users' online relationships based on the relationship graphs of communication. They define five hypotheses that analyze the social network influence from different aspects. The hypotheses are based on the number of nodes among graphs, the connection number of individual nodes, the tie scores among relationships, the message exchange frequency, and related social activities. Their experiment results show that these hypotheses have given us an idea of how social networks relationship and activities affect cyberbullying classification but don't provide a perfect conclusion. These observations show that it is the right direction to analyze social networks features and it needs more research and hypotheses to better understand the social network relationships. Hence, it is a useful way to combine content and social network features together when classifying cyberbullying.

M. Pushkar et al. [3] propose a solution to detect abuse behaviour on online social media platform based on user profile. Their dataset has three labels which are racism, sexism, and normal. Author Profiles are also created to represent the follower information of the author and social network features. In their experiments, they compare the performance of only using author profile, using author profile and textual features by different feature extraction techniques. From the results, they conclude that it is efficient to detect abuse based on community information about users. In Our project, we use their dataset as our baseline. However, our dataset makes some changes and only contain two class, one is bullying, which combines racism and sexism together, the other one is normal. Our dataset also has more information of each tweet, such as user IDs, tweet content and retweets user list.

4.3 User Behaviour Analysis Approaches

Users' behaviour may be affected by their social groups. For example, if the social circle is aggressive, they may also act aggressively or bully others because aggression behavior may propagate among the neighbors. The aggression propagation on social media may also be a useful factor, which can help improve the cyberbullying classification. C. Terizi et al. [20] build a model to study how aggression is spread among the users within a period of time based on the idea of opinion propagation model and compare the simulated aggressive propagation with the real social network media. They model the dynamics of aggression propagation and they conclude that the occurrence of cyberbullying behavior can be dropped if reasonably monitor the interaction of aggressive users with the normal users. The experiment proposed by C. Terizi [20] gives us an idea that the aggression of social networks may also affect the behavior of cyberbullying. To address this feature, it is necessary to study the aggression of social networks when classifying cyberbullying. The more aggressive the

social network is, the more likely cyberbullying happens.

Considering that Cyberbullying is an intened behaviour that related to user's subjective consciousness, G. Suyu et al. [21] study the effect of users' behaviour during a period time on online social media platform in order to improve the performance of Cyberbullying detection. In their approach, they build a unified time chart for each session on social media platform and create a graph to represent the interaction among users within a session. Their dataset comes from Twitter platform and it also contains the comments of each tweets. By studying the current and history comments, user's language habit, and user's interaction within the social community, they build session classification models with CNN, HANCD and SICD et al. The results indicate that Cyberbullying detection involves user interaction based on time period is meaningful. It also show the effective of their model and provide a direction for the future work.

CHAPTER 5

Methodology and Experimental Evaluation

5.1 Datasets

Twitter platform has provided many useful information for text analysis such as tweets contents and user information. Also, it is one of the most popular resources used in cyberbully analysis. Normally, the researchers will collect Twitter dataset with tweet content. As shown in Table 1, the original twitter dataset [3] contains tweetId and annotation. The category data is represented by number 0, 1 and 2. 0 represents that the tweet is racism, 1 means the tweet belongs to sexism message and 2 means the tweet belongs to normal message. For our dataset, we treat racism and sexism as bullying message and combine them together. We have two labels represented by 0 and 1, we use 1 to represent bullying message and use 0 to represent non-bullying message.

tweetId	annotation
5634592**	normal(2)
5645366**	racism(0)
5723460**	sexism(1)

Table 1: Examples of the original twitter dataset

For our experiment, we need more information about the social network and the communication. In order to get more information about the user relationship, we expand the original dataset with more features. As shown in Table 2, we added tweet id, tweet content, userName, and retweet userId list into the original dataset. In order to expand the original data set [3], we use the Twitter API to add more features. Twitter API provides interface to retrieve the information such as user's name by user ID, user's friends by user's name, tweet content by tweet ID, and retweet information by tweet ID. With the help of Twitter API, we can get the necessary information for

our cyberbullying analysis.

tweetId	tweetContent	userId	userName	label	retweetList
5723357**	we have to hear the word sassy	14484**	lottie**	1	253727**

Table 2: Example of experimental twitter dataset

With the userId and UserName, we can check whether two users are friends or not. It will help us analyze the relationship between users. We also get information about retweet user list for each tweet. Since we cannot get the exact information about how close two users is, retweet user list will reflect the closeness between users. Normally, the closer the relationship is, the easier they can influence each other. Our experiment will consider the influence of the social network relationship. The original twitter dataset has about 16,000 tweet message. However, there are some invalid users and some of the account have been deleted. After filtering the useless information, the size of our dataset is about 11,000. Table 3 shows the size of data for each category in our dataset. The experiment also contains data preparation phase. During the next

category	size
cyberbullying dataset	about 2.9k
non-cyberbullying dataset	about 7.9k
training dataset	about 2.2k
testing dataset	about 8.6k

Table 3: Enhanced dataset overview

section, the data prepossessing phase will be introduced.

5.1.1 Data Prepossessing

Before the experiment, we need to process the data to make sure that it is cleaned and ready to be analyzed. The raw tweets' content contains URLs that start with http

or https. They also contain emoji such as smile face and grinning face. Also, there are many punctuation and spaces between words. Those features are not considered in the cyberbullying analysis for now. In order to make it clean, we process the data as the following steps. Before the processing, we split the dataset into two parts based on the category, which are cyberbullying dataset and normal dataset. These two datasets are processed in the same. Firstly, we removed some unimportant content in tweets content such as URLs, punctuation, emoji. Secondly, we adjusted the structure of the tweet content. we replaced multiple space with single space and convert all the content to lower case. Thirdly, the content of the tweet was processed. Stopwords are commonly used words that will not be pay attention to during the analysis period. Based on NLTK stopwords library, We created a stopwords file that contains some stopwords such as 'a', 'an', 'the', 'I', and 'you'. We removed stopwords in tweet content based on the stopwords file. Also, Lemmatization was applied to words. Lemmatization consider the morphological of words and it helps Machine Learning better applied to text. Then, the tweet content is tokenized by using After these basic steps, the tweet content is cleaned and ready to be further analyzed. Table 4 shows an example of the data preprocessing. To simplify the data preprocessing work, our

	example
Before preprocesing	Haha yes.. Kat and Andre failing!! KARMA MOTHERFCKERS
After preprocessing	haha kat andre failing karma motherfuckers

Table 4: Example of data preprocessing

project apply a useful tool called OpenRefine [1]. It is used to process the messy data and it provides powerful functions to help clean and unify the data, such as unify

[1]https://openrefine.org/

the form and filter the content. Figure 9 shows the interface of OpenRefine with the example of our Twitter dataset.

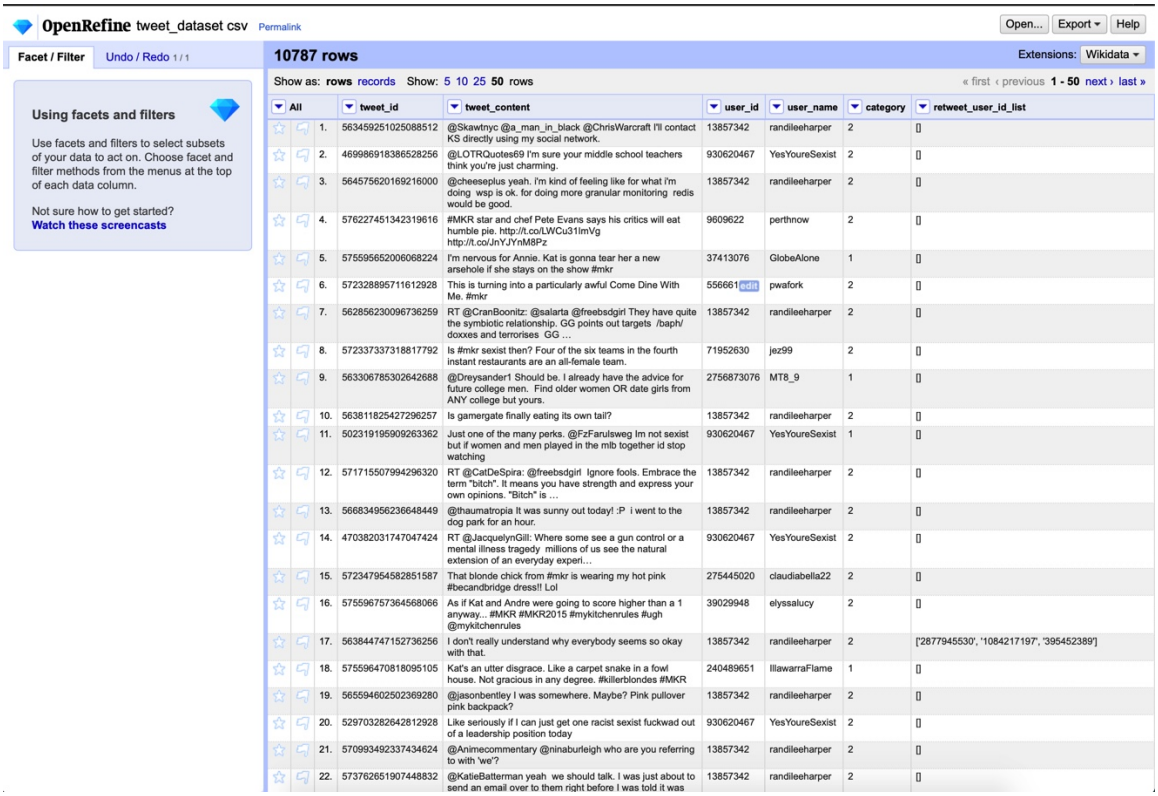

Figure 9: Example of using OpenRefine

5.2 Feature Extraction

After processing the data, we need to extract the features of the tweet content and use these features to analyze whether a tweet content is cyberbullying or not.

5.2.1 Word2vec

In our project, In order to transfer the tweet content to a feature vector, we average the feature vector of a word that appears in the weet and vocabulary list. We use the vocabulary list created by trained Word2Vec model based on all tweet dataset. Different value of parameters can affect the performance of the model. As Table 5 shows, different value of parameters have been tried for our word2vec model.

To get a better performance, we set the window size is 5, the embedding dimension is 256, and min_word_count is 5.

classifier	window=10, dimension=128	window=5, dimension=256
Random Forest	0.781	0.788
SVM	0.740	0.741
Logistic Regression	0.737	0.739
Ada Boosting	0.753	0.767

Table 5: Accuracy of classifiers with different parameters

5.2.2 TF-IDF

For our experiment, if a word appears more times in cyberbully tweets and appears less times in normal tweets, this word can represent the cyberbully as a feature. In order to calculate the TF-IDF, we use the following expression

$$W_{ij} = \frac{N_{ij}}{\sum_k N_{kj}} \times \lg \frac{D}{1 + S_{tj}} \tag{1}$$

where W_{ij} is the TF-IDF score for a word, N_{ij} is the count of a word appears in the document d and denominator is the count of all words appear in document d. S_{ij} is the amount of document that contains word t_j. Considering that the word may not appear in the corpus and denominator will be 0, we use $1 + S_{ij}$ as the denominator.

5.2.3 Node2vec

In our project, we use node2vec model to analysis the social network relationship graph. The node in the graph is each user with a label that indicate whether a user belongs to cyberbully group or not. We use node2vec to extract the features of each user for classification model. The implementation is based on the open resource provided by Standford university [22]. Since the social network relationship graph is weighted graph, we add the weight feature in node2vec model and try different values of the parameter. 'dimensions' decides the embedding dimension of the model

and 'window size' decides the size of context to be optimized. Table 6 shows feature vector by applying node2vec on our dataset. The node is user id and the dimension of feature vector is 128.

node	feature 1	feature 2	...	feature n
35588686	0.4874009	0.19998972	...	0.10173487
561596765	-0.02297074	0.17605074	...	-0.085382156
...	...	...	...	...
180999150	0.19880648	-0.036795933	...	-0.036038816

Table 6: Overview of node features

5.3 Classify Tweet based on Text Features

The first experiment model is tweet classification based on text features. After text feature extraction, we find that TF-IDF has better performance than word2vec. The size of text feature is about 8k that contains the main content. In our experiment, we apply different classifiers on text features as follows

- Random Forest - Random Forest builds the Bagging integration based on the decision tree and randomly select attributes in the training process of the decision tree. Compared with decision Tree, it reduced overfitting problem. In our project, we use RandomForestClassifier from sklearn library. After experiments, we set maximum depth of the tree as five and use the default value of n_estimators.
- SVM - SVM is a binary classifier that maximum interval in feature space. It has different kernel functions to solve classification problem. SVM can solve linear and non-linear problems by choosing different kernel functions. In our project, we try linear kernel and radial basis function (RBF) Kernel. The performance of different kernel has be evaluated.
- Logistic Regression - Logistic Regression is a generalized linear regression analysis model that is used to solve classification and prediction problem. It compresses

the prediction range from the real number domain to the (0,1) range, thereby improving the prediction. In our project, we use 'liblinear' as the 'solver' algorithm, which is the algorithm used in the optimization problem.

- Ada Boosting - Ada Boosting is one of the most common used boosting technique. It combines weak classifiers together to bulid a more powerful classifier and it is iterative and adaptive. In our project, we use AdaBoostClassifier from sklearn library. We set the maximum number of estimators as 200 and it also uses decision tree classifier inner the ada boosting classifier.
- Naïve Bayes - Naïve Bayes is a simple classifier based on Bayes Theorem. Assume we have n features and m categories, by using Naïve bayes classifier, the category with the highest probability will be calculated based on the features. It requires that features should be independent to get a better accuracy. In sklearn library, there are three Naïve Bayes classification algorithm classes, which are GaussianNB, MultinomialNB and BernoulliNB. The advantage of Naïve Bayes algorithm is that it is an easy way to classify the category. And it can use a small number of dataset to estimate the feature probability. And it can also classify multiple classes. In our project, GaussianNB class has been used to solve tweet classification problem.
- SGD - Stochastic Gradient Descent (SGD) is an efficient classifier that is widely used in large-scale and spares problems such as text classification problem [23]. It uses gradients to improve parameters and supports multi-class classification. It has loss function and penalty parameter to adjust the model.
- CNN - CNN is a neural network model that used in image processing, face recognition, and natural language processing [24]. It uses convolution computation in each layer. In our project, we use word embedding and padding to complete the training. The dimension of embedding is 300 and all input tweet

should have same size to fit CNN model. In our project, we use the keras.layers and keras.models library to implement CNN. We use 5 different filters and GlobalMaxPooling1D layers for each layer. Also, two dropout layers and a dense layer are applied to concatenated outputs.

- LSTM - LSTM is a special RNN model that used in analyzing sequential data problem. For example, text analysis and prediction problem. In LSTM model, it decides what information need to be dropped, what new information needs to be remembered and saved in cells units and what is the output. It will remember the long term memory and forget the unimportant information. In our model, we applied foward LSTM and backward LSTM, and concatenate the two output together.

Algorithm 1 introduces the steps of classification models based on text features. We try different feature extraction method in step 3 and find that TF-IDF has better performance than word2vec. Different classifiers listed above are also applied in our experiment and the results are evaluated and compared.

Algorithm 1: Algorithm for method 1

```
1 function TextFeatureClassification(T):
  Input  : T: text corpus
  Output : A: accuracy of classifiers
2 T = preprocessData(T);
3 Training, Testing = splitDataset(T);
4 Train_features,Test_features = extractTextFeature(Training, Testing);
5 classifier = trainingWithClassifiers(classifier, Train_features);
6 A = testingWithClassifiers(classifier, Test_features);
7 return A;
```

5.3.1 Experiments and Results

After training phase, we compare the different metric to evaluate the experiment result. Table 7 shows the bullying tweet classification accuracy of different classifier

that TF-IDF to extract text feature. We use n-fold cross validation method and 20% of data is set as testing dataset and 80% of tweet data is set as training dataset. As Table 7 shows, SVM model with linear kernel has the best performance that is about 90%. For nerual network models, CNN also has a good performance and achieves 86% accuracy. The results indicate that the classification models can identify bullying and normal tweet of our dataset well. The execution time of each classifier is calculated. Based on the Table 7, the SVM model has longer running time than other machine learning methods and it also has the best accuracy. The neural network methods take much longer time to complete the training phase because we assign multiple epochs to get better accuracy.

classifier	accuracy	execution time
Random Forest	0.7405	0.3057s
SVM(linear)	**0.8943**	13.7521s
SVM(rbf)	0.7516	16.4466s
Logistic Regression	0.8712	0.1243s
Ada Boosting	0.8679	0.9967s
Naïve Bayes	0.8285	0.0063s
SGD	0.8258	0.0219s
CNN	0.8679	195.4535s
LSTM	0.7405	108.8269s

Table 7: Accuracy of method 1

Table 8 shows the f1-score of classifiers for bullying and normal category. The classification is only based on text features. The table suggests that the best score for normal category is 0.93 and best score for bullying category is 0.78. The classifiers can better classify normal tweets. Random Forest has the most imbalanced performance and the accuracy for bully data is almost zero. This may because there are more normal items in our dataset.

Figure 10, 11, 12, 13, 14, 15, shows ROC curve for classifiers. From these figures,

classifier	bullying	normal
Random Forest	0.01	0.85
SVM	**0.78**	**0.93**
Logistic Regression	0.70	0.92
Ada Boosting	0.72	0.91
Naïve Bayes	0.52	0.90
SGD	0.67	0.88
CNN	0.68	0.91

Table 8: Accuracy for each category

we can find that the performance of SVM with linear kernel is better than SVM with rbf kernel. Additionally, SVM, logic regression and Naïve bayes have similar performance, the AUC value of these model are almost 0.88. However, the performance of random Forest is poor. Different value of parameters in random forest model have been tried, but the performance didn't improve obviously, the accuracy is basically around 0.7.

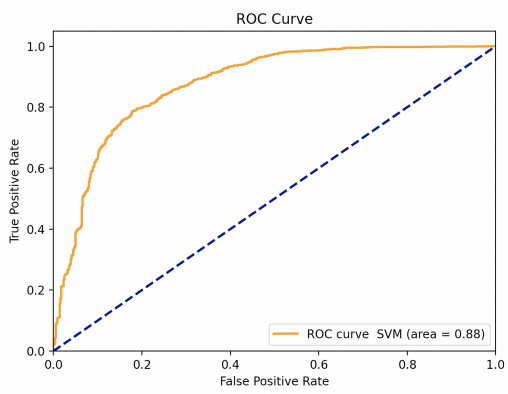

Figure 10: ROC Curve for SVM(rbf)

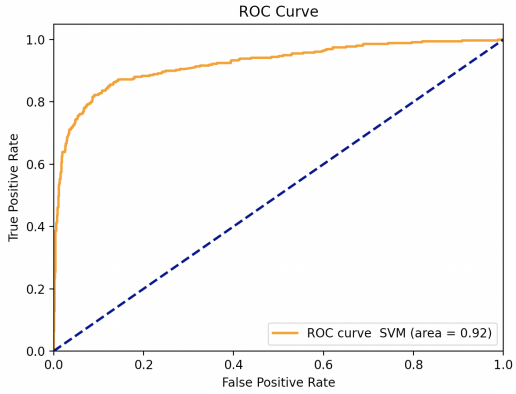

Figure 11: ROC Curve for SVM(linear)

5.4 Classify Tweet based on Social Network Relationship

In this section, we introduce how to classify tweet by combining text features and social network relationship features together.

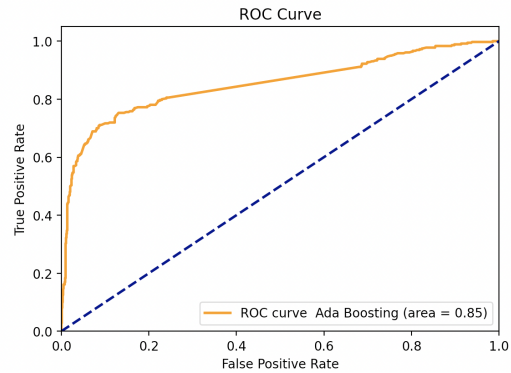

Figure 12: ROC Curve for ada boosting

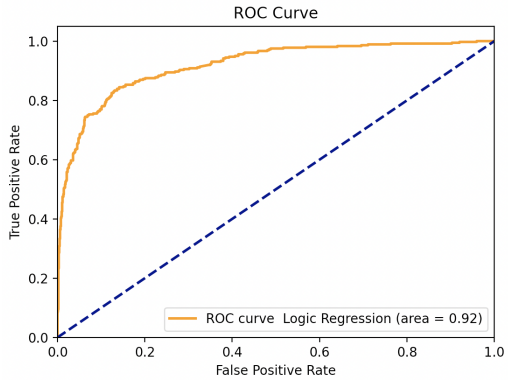

Figure 13: ROC Curve for logic regression

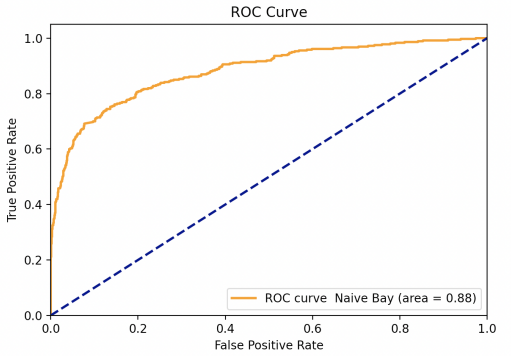

Figure 14: ROC Curve for Naïve bayes

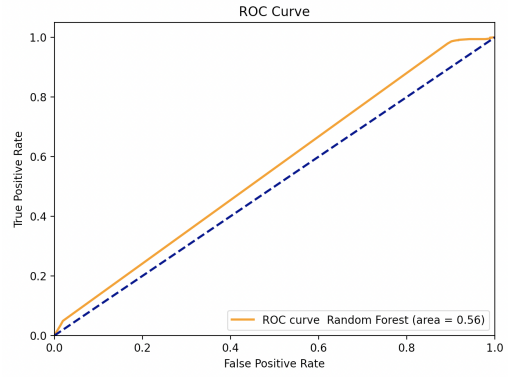

Figure 15: ROC Curve for random forset

5.4.1 Graph Construction

In order to represent the social network relationship, we need to construct a graph to analysis the community interaction. This graph contains the follower information and closeness between each user. The following algorithm 2 shows the steps to construct the graph.

The first step is to collect all nodes. In the graph, the node is each valid user from our dataset and the edge with weight indicates the follower relationship and closeness between users. The graph is generated based on the dataset that contains tweet id and category. We can get user id based on the tweet id and collect users' followers based on user id. In order to generate the graph, it needs retrieve all the valid users

Algorithm 2: Graph construction algorithm for social network relationship

```
1 function GraphConstruction(R, U, T):
  Input  : R: retweet information,
           U: user followers,
           T: text corpus
  Output : G: undirected weighted graph
2 N = extractNode(T);
3 E = buildEdgeWith(R, U);
4 E_W = assignWeightToEdge(N, E, R, U);
5 G = constrcutGraph(N,E)
6 return G;
```

and remove duplicate users since many tweets may from the same users. Also, since some users posted many aggressive words, they may be deactivate or invalid, we need to filter those users. The user information is saved in a separate csv file with the column user id, user name, user's followers and all retweet user list. To clean the user data, we use the functions from excel, we remove duplicate users by using 'remove duplicate' function in excel and unify the form of data. After removing the duplicate users, the total number of users in our dataset is 1477. Figure 16 shows the overview of nodes in social network relationship graph that generated by Gephi[2].

The second step is to generate the graph based on the user information file. The following algorithm 3 describes the details about how we construct the weighted edges step by step.

We use Networkx library in python to create a graph and use this graph to represent social network relationship. Networkx is a powerful tool provided by Python and it can be used to generate complex graph. It supports different types of files to generate the network structure. It can construct directed graph, undirected graph, multi graph as well as weighted graph and unweighted graph. This useful library helps us generate graph structures more easily. For our project, we use Networkx to

[2]https://gephi.org/

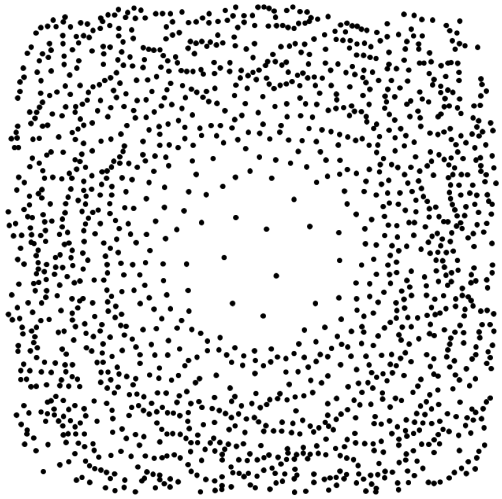

Figure 16: UserNodes of social network graph

create a undirected weighted graph. We assume that if one user follow the other user, then there is a relationship between these two users. Also, if one user retweet the other user's tweet, we assume that there is also a relationship between these two users even if they don't follow each other. The weight represent the closeness between two users and it is calculated as follows

$$weight = \begin{cases} 1, & \text{if has follower relationship or retweet relationship} \\ weight + 1, & \text{if retweet times increase} \end{cases}$$

The default value of weight is 1. The value of weight will increase as the retweet times from one user increases. After calculate the weight between nodes, we can construct the weighted edges between two nodes. As Figure 17 and Figure 18 show, user b and user c is user a's follower, so there is an edge between them and the weight is 1. Similarly, user b and user d retweeted user a's post, so there is also an edge between them and the weight is 1. After the combination of Figure 17 and Figure 18, we get Figure 19, which is a part of the final graph. In our project, we only consider the

Algorithm 3: WeightedEdges construction algorithm

```
1  function AssignWeightToEdge(N, E, R, U):
     Input  : N: nodes list,
              E: edges list,
              R: retweet information,
              U: user followers,
     Output : E_W: weighted edges list,
2    for t in N do
3      if t has followers then
4        followers = U.get(t)
5      retweet_user_list = R.get(t)
6      for v ∈ N do
7        weight = 1
8        retweet_weight = 0
9        if retweet_user_list exists and v in retweet_user_list then
10         retweet_weight = retweet times * v
11       if v in followers of t then
12         if edge(t,v) exists in E then
13           val = weight_of_edge(t,v)
14           weight_of_edge = val + max(weight,
               weight+retweet_weight)
15           E_W.add(weight_of_edge)
16         else
17           weight_of_edge = max(weight, weight+retweet_weight)
18           E_W.add(weight_of_edge)
19     end
20   end
21   return G;
```

nodes that have relationship between each other, which means if there is no follower relationship or retweet relationship between two users, we will not connect the edge between two users. In other words, we only consider the existing relationship and it will reflect in the graph. As Figure 20 shows, the center part of the graph is dense and the relationship between each nodes is complex. The node figure and graph figure is generated by Gephi[3].

[3]https://gephi.org/

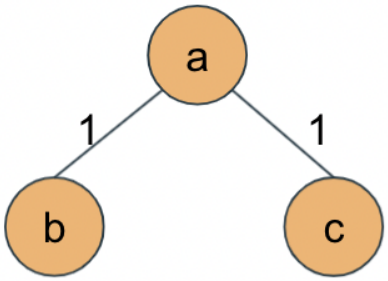

Figure 17: Follower Relationship

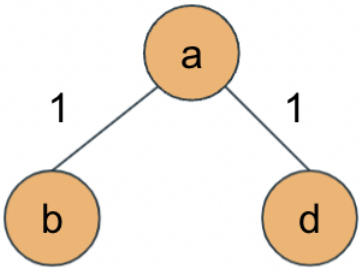

Figure 18: Retweet Relationship

5.4.2 Experiments and Results

After construct the graph, we apply node2vec to extract the features of each node. For node2vec model, we choose the 'dimension size' as 128 and 'window size' as 5. So,

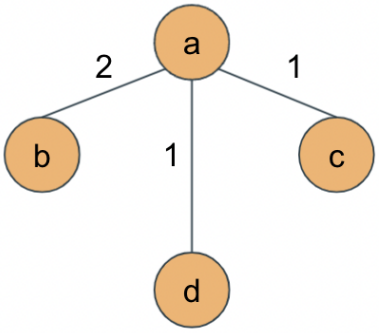

Figure 19: Overview of weighted graph

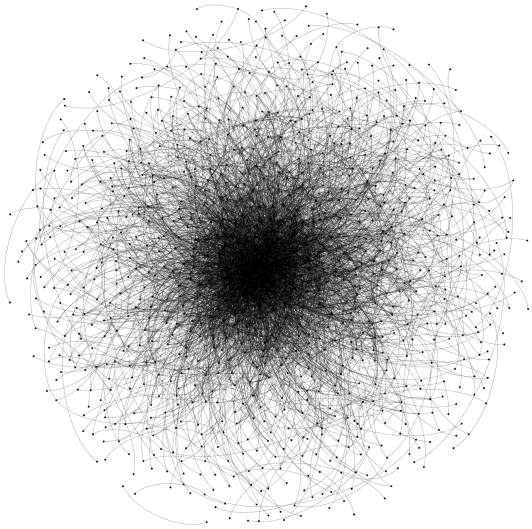

Figure 20: UserNodes of social network graph

the total number of user feature is 128. For text feature extraction, after comparing the performance and accuracy of word2vec and TF-IDF, we find that TF-IDF has better performance. So we use TF-IDF to extract text features. For each tweet, we have text feature vector by using TF-IDF and we also know the user id of each tweet. Based on the specific user id, we can get user's feature vector from node2vec. Then we can combine user's feature vector and text feature vector together to classify tweet category.

Table 9 shows tweet bullying classification accuracy by using combination features. The results shows that SVM classifier with linear kernel has the best accuracy that is about 92%. Table 10 shows the comparison of the performance for some classifiers by using method 1 and method 2. Compared with method 1 that only use text features, the best accuracy of method 2 that use combination features has also been improved from 89% to 92%. For most of the other classifiers, compared with using text features only, the accuracy of using combination features are also improved. Especially for random forest classifier, the accuracy increases 6%. However, the accuracy of Naïve

bayes isn't improved. This may because the features have relationship with each other and are not completely independent. For Naïve bayes model, if the feature size is large and features are dependent, the performance may be worse. Besides, Logistic Regression also has a better performance whose accuracy is about 90%. The execution time of method 2 takes longer time compared with method1. This is because we combine the textual features and graph features together, and the size of feature vector is larger. As the size of features increase, the execution time also increase. Especially for SVM model, it takes almost one hour to complete the training. Additionally, our project uses the paper [3] as our baseline. The best F1 score from their experiment is 87.57, which is generated by using author profile and text content together to detect the abuse behavior. Compared with their best result, our best result is 91.55, which exceeding their best result by 4 points. This may because we add retweet information in our social network feature. However, their dataset has three labels and our dataset has two labels, which makes it easier for us to improve the accuracy. All in all, both results of our experiments can show the efficiency of social network features in abuse behaviour detection.

classifier	accuracy	execution time
Random Forest	0.8089	1.6555s
SVM(linear)	**0.9155**	48min18s
SVM(rbf)	0.8102	45min32s
Logistic Regression	0.9062	0.5884s
Ada Boosting	0.9023	175.1948s
Naïve Bayes	0.6502	1.1708s
SGD	0.8271	6.1348s

Table 9: Accuracy of method 2

Besides, In our project, we also construct an unweighted graph that only represents the relationship between nodes. We find that the overall performance of unweighted

classifier	method1 accuracy	method2 accuracy
SVM(linear)	0.8943	0.9155
SVM(rbf)	0.7516	0.8102
Logistic Regression	0.8712	0.9062
Ada Boosting	0.8679	0.9023
Naïve Bayes	0.8285	0.6502

Table 10: Comparison of Accuracy for method 1 and method 2

graph is better than weighed graph. As Table 11 shows, the best accuracy of unweighted graph has achieved 92% and the best accuracy of weighted graph is 91%. The accuracy of other classifiers are almost the same. Also, The execution time of unweighted graph is also improved. Based on the results, both weighed graph and unweighted graph are efficient for cyberbullying classification.

classifier	accuracy of weighted graph	accuracy of unweighted graph
Random Forest	0.8089	0.8080
SVM(linear)	0.9155	0.9247
Logistic Regression	0.9062	0.9111
Ada Boosting	0.9023	0.9043
Naïve Bayes	0.6502	0.6502

Table 11: Comparison of Accuracy for weighted and unweighted graph

Table 12 shows the f1-score of classifiers for bullying and normal category. Based on the results, the best score for normal category is 0.94 and best score for bullying category is 0.83. Overall, the classifiers work better on normal tweet than bullying tweet. Compared with scores of classification based on text features, the scores of classification based on combination features are also increased, which indicates that social network relationship features is related to cyberbullying classification and adding the related features to classifiers can improve the performance of classification.

Figure 21, 22, 23, 24, 25, 26 show ROC curve for classifiers. From these figures,

classifier	bullying	normal
Random Forest	0.45	0.88
SVM(linear)	**0.83**	**0.94**
SVM(rbf)	0.47	0.88
Logistic Regression	0.80	**0.94**
Ada Boosting	0.81	**0.94**
Naïve Bayes	0.50	0.73
SGD	0.55	0.89

Table 12: Accuracy for each category

we can find that the performance of SVM, logic regression and ada boosting are similar and they both performance better when adding related features, these models are robust, while, random forest and Naïve bayes performance poorly.

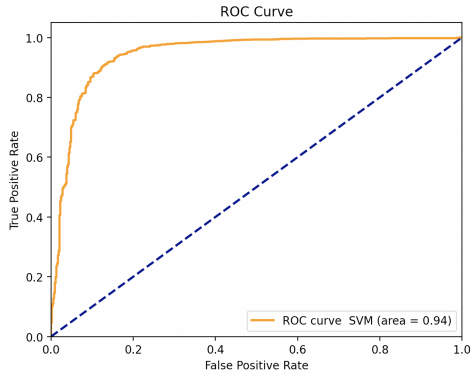

Figure 21: ROC Curve for SVM(linear)

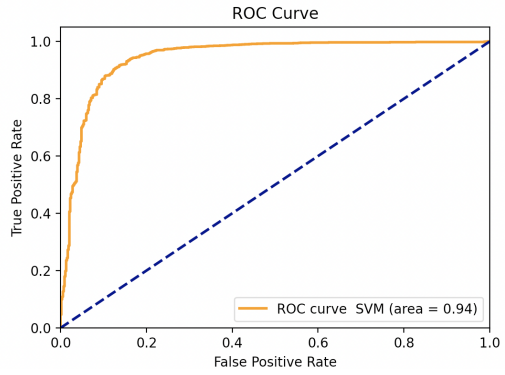

Figure 22: ROC Curve for SVM(rbf)

5.5 Classify Bullying Tweet Users based on Social Network Relationship

Based on social network relationship graph, we can also classify users based on user feature vector. In our dataset, each tweet corresponds to a user and One user may post many tweets. For each user, we create a list of tweet category where the tweets are posted by that user. As the algorithm 4 shows, If a user posted more bullying tweets than normal tweets, then this user will be marked as bullying user. Otherwise, the user will be classified as normal user. Each user has a label for himself

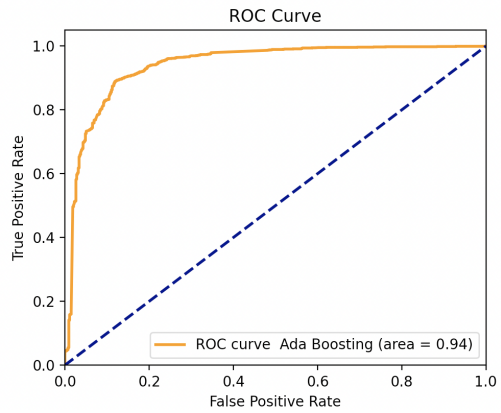

Figure 23: ROC Curve for ada boosting

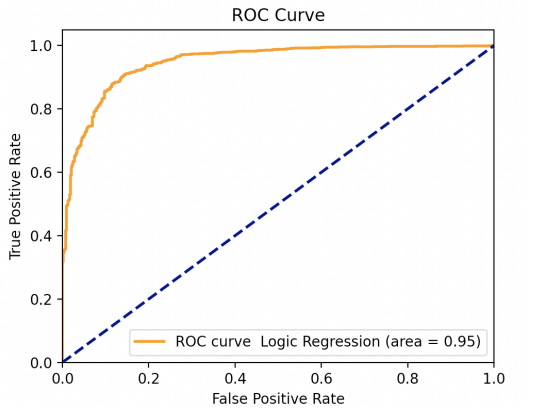

Figure 24: ROC Curve for logic regression

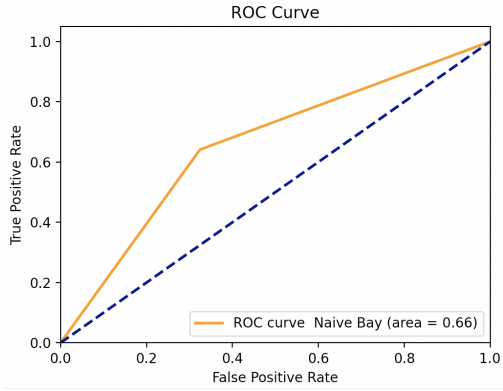

Figure 25: ROC Curve for Naïve bayes

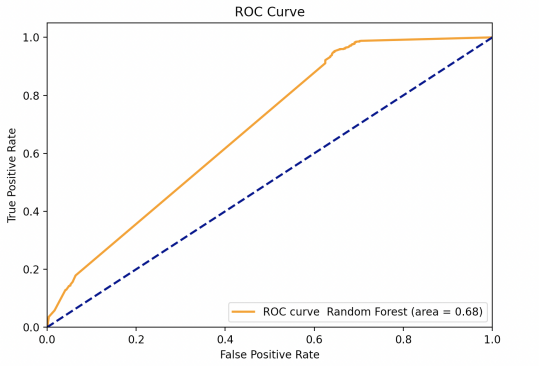

Figure 26: ROC Curve for random forest

based on the tweets he posted. In this experiment, we will classify users based on node features from node2vec. The total number of user is 1477 and the number of normal user is 1221. The ratio of bullying user and normal user is about 1:6. The performance of different classifier will be evaluated.

5.5.1 Experiments and Results

Table 13 shows the accuracy of different classifiers that used to classify users based on social network graph feature. The best accuracy is generated by SVM and Random forest model, which is 82%. The overall accuracy is not high. This may because there are more normal users than bullying users after using our identification

Algorithm 4: Identify label for each user

```
function AssignLabelToUser(T):
  Input  : T: text corpus with user and tweet label
  Output : User2Label: list of all users with label
U = extractUser(T);
L = extractAllTweetLabelForUser(U);
User2Label = assignLabelToUser(L);
return User2Label;
```

algorithm. However, based on the result, we can find that there are relationship between social network graph features and bullying user classification. The execution time of each classifiers are also shown in Table 13. Ada Boosting classifier takes the longest time and the execution time of SVM is also long.

classifier	accuracy	execution time
Random Forest	0.8209	0.0779s
SVM	**0.8209**	0.4253s
Logistic Regression	0.8074	0.0142s
Ada Boosting	0.7939	1.9204s
Naïve Bayes	0.5709	0.0025s
SGD	0.5709	0.0198s

Table 13: Accuracy for user classification

Figure 27, 28, 29, 30, 31 show ROC curve for classifiers. From these figures, we can find that the performance of these classifiers are similar and the value of AUC is amost 0.5, which means that the classifiers cannot distinguish the bullying user and normal user well. This may because the imbalance between bullying user and normal user. The total number of users is about 1400 and Based on our identification model, there are about 1200 normal users and 200 bullying users. There are more normal users than bully users. Also, the size of our user dataset is not large enough for an accurate classification. A larger dataset is also required to do the classification and

there's still a long way to go to improve the model.

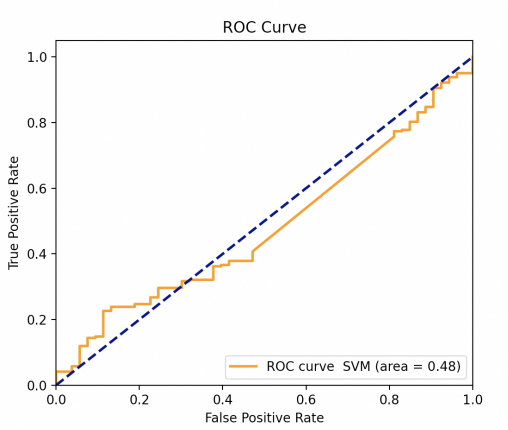

Figure 27: ROC Curve for SVM

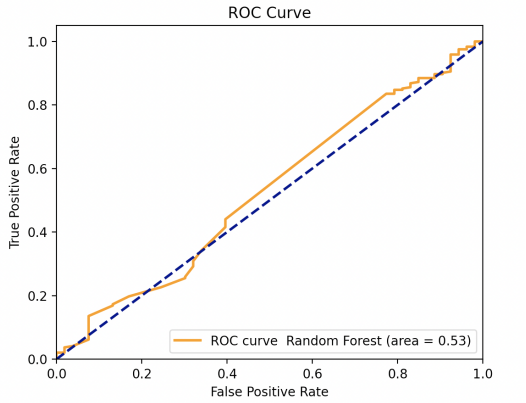

Figure 28: ROC Curve for Random Forest

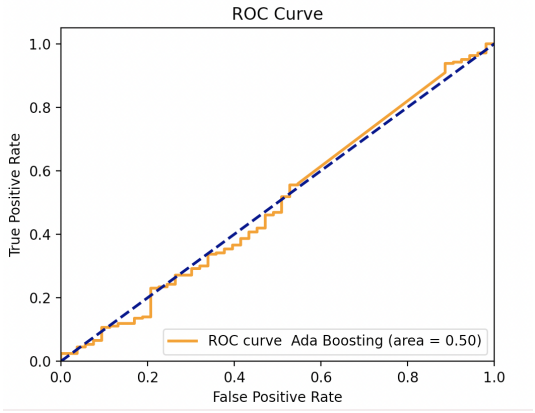

Figure 29: ROC Curve for Ada Boosting

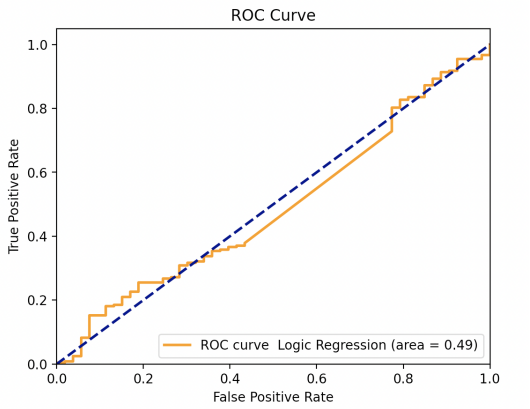

Figure 30: ROC Curve for Logic Regression

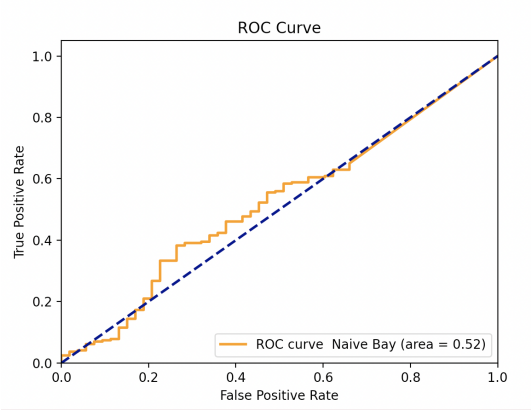

Figure 31: ROC Curve for Naïve Bayes

CHAPTER 6

Conclusion and Future Work

6.1 Conclusion

As the number of social media users increases, more and more problems threaten the safety of online social network environment. Cyberbullying is one of the typical problem that spreads rapidly among a group and can be affected by the social relationship. In this project, we added social network relationship features that represented by community graph and collected about 11,000 tweets as the dataset for our classification models. We used retweet information to represent the closeness between followers. We applied different supervised machine learning models and neural network models on the project and compared ROC curve of each model to evaluate the performance. The first phase only contained text features. The results showed that Support Vector Machine achieved 89% accuracy, which has the best performance among all other models. The second phase added community graph features to the model. The results shows that Logic Regression model as well as Ada Boosting model achieved 91% accuracy, which are the highest accuracy. All in all, the results indicated that our model can classify Cyberbullying and normal comment accurately and also social network relationship feature can affect classification based on our Twitter dataset. After adding social network relationship feature, the accuracy and performance are improved for some classification models. The results also reflected that cyberbullying has relationship with social network.

6.2 Future Work

Many advanced machine learning models have been applied on classification problems and had good performance. In order to better complete the Cyberbullying classification, there are still many features can be added and analyzed since Cyberbullying Classification is not only a machine learning topic, but also a sociological

issue. For the future work, one direction is that we can add user's profile such as user's search history and education level as extra features. Also, in the future, we can analyze the effect of community features in cyberbullying classification. Furthermore, to better analyze the efficiency of our classification model, we can increase the diversity of dataset by collecting larger size of dataset from other online social platforms such as Facebook and YouTube comments. Social network relationship feature also gives us a direction that we can use this information to classify users. In the future, we can also add other features such as users' profile to improve the performance of user classification.

LIST OF REFERENCES

[1] D. Hango, ''Cyberbullying and cyberstalking among internet users aged 15 to 29 in Canada,'' *Insights on Canadian Society*, 12 2016.

[2] K. Gorro, M. J. Sabellano, K. Gorro, C. Maderazo, and K. Capao, ''Classification of cyberbullying in facebook using selenium and svm,'' 04 2018, pp. 183--186.

[3] P. Mishra, M. Del Tredici, H. Yannakoudakis, and E. Shutova, ''Author profiling for abuse detection,'' in *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1088--1098. [Online]. Available: https://www.aclweb.org/anthology/C18-1093

[4] https://developer.twitter.com/en/docs.

[5] M. Stamp, *A Survey of Machine Learning Algorithms and Their Application in Information Security: An Artificial Intelligence Approach*, 09 2018, pp. 33--55.

[6] S. Hochreiter and J. Schmidhuber, ''Long short-term memory,'' *Neural computation*, vol. 9, pp. 1735--80, 12 1997.

[7] S. Albawi, T. A. Mohammed, and S. Al-Zawi, ''Understanding of a convolutional neural network,'' in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1--6.

[8] K. O'Shea and R. Nash, ''An introduction to convolutional neural networks,'' *ArXiv e-prints*, 11 2015.

[9] M. Stamp, *Introduction to Machine Learning with Applications in Information Security*, 09 2017.

[10] M. Stamp, ''Boost your knowledge of adaboost,'' 10 2018.

[11] A. Grover and J. Leskovec, ''node2vec: Scalable feature learning for networks,'' vol. 2016, 07 2016, pp. 855--864.

[12] https://developers.google.com/machine-learning/crash-course/classification/accuracy.

[13] Q. Huang, V. Singh, and P. Atrey, ''On cyberbullying incidents and underlying online social relationships,'' *Journal of Computational Social Science*, vol. 1, 09 2018.

[14] V. Singh, A. Varshney, S. S. Akhtar, D. Vijay, and M. Shrivastava, ‘‘Aggression detection on social media text using deep neural networks,’’ 10 2018.

[15] S. Sadiq, A. Mehmood, D. S. Ullah, M. Ahmad, G. S. Choi, and B.-W. On, ‘‘Aggression detection through deep neural model on twitter,’’ *Future Generation Computer Systems*, vol. 114, 07 2020.

[16] K. Richard and L. Marc-André, ‘‘Generalisation of cyberbullying detection,’’ 09 2020.

[17] A. Squicciarini, S. Rajtmajer, Y. Liu, and C. Griffin, ‘‘Identification and characterization of cyberbullying dynamics in an online social network,’’ 08 2015, pp. 280--285.

[18] M. Dadvar, F. de Jong, R. Ordelman, and D. Trieschnigg, ‘‘Improved cyberbullying detection using gender information,’’ 01 2012.

[19] Q. Huang, V. Singh, and P. Atrey, ‘‘On cyberbullying incidents and underlying online social relationships,’’ *Journal of Computational Social Science*, vol. 1, 09 2018.

[20] C. Terizi, D. Chatzakou, E. Pitoura, P. Tsaparas, and N. Kourtellis, ‘‘Angry birds flock together: Aggression propagation on social media,’’ 02 2020.

[21] S. Ge, L. Cheng, and H. Liu, ‘‘Improving cyberbully detection with user interaction,’’ 11 2020.

[22] https://github.com/aditya-grover/node2vec.

[23] https://scikit-learn.org/stable/modules/sgd.html.

[24] W. Yin, K. Kann, M. Yu, and H. Schütze, ‘‘Comparative study of cnn and rnn for natural language processing,’’ 02 2017.