VALLEY VIEW UNIVERSITY

FACULTY OF SCIENCE

DEPARTMENT OF INFORMATION TECHNOLOGY

A PROJECT SUBMITTED IN PARTIAL FULFILMENT FOR THE REQUIREMENT

FOR A BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

PROJECT TOPIC:

DETECTION OF CYBERBULLYING ON TWITTER USING MACHINE LEARNING

BY:

BANNERMAN BENEDICT QUANSAH

218IT02000021

DATE:

11TH OCTOBER, 2021

TABLE OF CONTENT

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to the Almighty God for grace and peace of mind. Secondly, a special thanks of gratitude to my supervisor, Mr. John Kingsley Arthur, and all the lecturers of the department for guidance and advice during the semester and for the good works they are doing at the Department. Finally, I would also like to thank my family and friends who helped me financially, emotionally, socially and spiritually in finalizing this project within the time frame.

DECLARATION

This is to declare that the research work underlying thesis has been carried out by the under mentioned student, supervised by the under mentioned supervisor. Both the student and supervisor certify that the work documented in this thesis is the output of the research conducted in the fulfilment of the requirement of a Bachelor of Science degree in Information Technology

Student:	Supervisor:
Bannerman Benedict Quansah	Mr. J.K Arthur
……	……

ABSTRACT

Bullying frequently occurs on social media, which is a place where many young people interact with people. As the number of social networking sites grows, so does cyberbully. There are numerous papers on cyberbullying detection, but most of them are based on the semantic closeness of words, neglecting how words are used in each context. This research focuses on an artificial cyberbullying detection system to detect cyberbullying on Twitter based on semantics and contexts using word2vec. This paper uses word2vec to extract features. Also, various machine learning algorithms (Logistic Regression, Decision Tree, Random Forest, XGBoost, AdaBoost and Naïve Bayes) will be used as models for training and prediction.

CHAPTER ONE (Research Project Proposal)

1. Introduction

Cyberbullying is the act of threatening, harassing, or bullying someone using modern methods of communication, such as social media apps/sites, to communicate with each other and with anyone/everyone in the globe, *Rahul et al. (2020).* Cyberbullying includes not just creating a false identity and publishing/posting an embarrassing photo or video, as well as spreading unfavourable rumours about someone and threatening them, *Rahul et al. (2020).* Cyberbullying's effects on social media are frightening, and it has even resulted in the murder of some unlucky victims. The victims' conduct also alters as a result of this, affecting their emotions, self-confidence, and dread, *Rahul et al. (2020).* As a result, a thorough solution to this problem is necessary. Cyberbullying must be eradicated *Hani et al. (2019).* The problem can be addressed by employing a machine learning approach to detect and prevent it; however, this must be done from a different perspective, *Van Hee et al. (2018)*. This research focuses on an artificial cyberbullying detection system to detect cyberbullying on Twitter based on semantics and contexts using word2vec. This paper uses word2vec to extract features. Also, various machine learning algorithms (Logistic Regression, Decision Tree, Random Forest,

XGBoost, AdaBoost and Naïve Bayes) will be used as models for training and prediction.

1.1 Subject and Field of Study

The subject of study is the detection of cyberbullying on Twitter using machine learning, and the field of study is machine learning.

1.2 Problem Statement

The semantics of words as expressed in valid statements and the context in which words are used in statements influences the accurate detection of cyberbullying. However, existing works are silent on the word semantics. Furthermore, the popularly used machine learning (ML) algorithms (Logistic Regression, Decision Tree, Random Forest, XGBoost, AdaBoost and Naïve Bayes) for classification of the text as per the existing works lacks their optimal network configuration. Therefore, this paper seeks to incorporate Grid Search algorithm to fine-tune the hyperparameters of these ML algorithms.

1.3 Project Objectives

1.3.1 Global (General) Objectives

The main goal of this project is to create and implement a cyberbullying detection system to prevent cyberbullying on Twitter

1.3.2 Specific Objectives

I. To create and implement a cyberbullying detection system to prevent cyberbullying on twitter, taking into consideration both the semantic closeness and how words are used in various context using word2vec.

II. To augment the performance of the ML algorithms used by incorporating Grid Search which will automatically select the optimal network parameter configuration using Grid Search.

1.4 Background of Study

The usage of information and communication technology, particularly social media, has altered the way people connect and create relationships, with data indicating that social media applications are widely used worldwide. According to a recent survey by Pew Research Center (2018), Instagram (75%) and Snapchat (73%) were found to be the most popular among those aged 18 to 24, while Facebook

and YouTube were more prevalent among those aged 50 and up (i.e., 68 percent). Unfortunately, this opens the door to anti-social behaviors, including misogyny (Anzovino et al., 2018; Liu et al., 2018), sexual predation (Bogdanova et al., 2014), sexism (Frenda et al., 2019), and cyberbullying (Kowalski et al., 2019; Chatzakou et al., 2017a, b; Hosseinmardi et al., 2015).

For example, Facebook is one of the most popular social networking sites, allowing users to create their profiles, upload images, and videos, and send private and public messages. It has a broad reach, as thousands can see any comment or post of people, thanks to "liking" and "sharing" methods, allowing cyberbullies to effortlessly spread ugly or unwanted information about their victims (Choo, 2016). Instagram users can share photos and videos, follow others, and participate in Stories. It's just as easy to create new, anonymous personas for cyberbullying as it is on Facebook. Because of the speed and scale of the distribution mechanism, threatening comments or humiliating photographs can go viral in hours (Hosseinmardi et al., 2015).

Victimization of cyberbullying, particularly among young people, has been under increased criticism. For example, after teens were bullied on their site,

resulting in multiple deaths, Ask.fm (a platform that allows people to ask each other questions anonymously) had to implement new safety procedures. Similarly, as a technique to curb cyberbullying, Instagram has added shadow banning online abusers (i.e., preventing a bully from posting or commenting on a post) (LiveMint, 2019).

On the other hand, Twitter is one of the top five social media platforms where the majority of users are subjected to cyberbullying (turbofuture.com, 2019). It allows people to send 280-character messages and currently has over 330 million active users (Statista, 2018). Cyberbullying and Twitter studies frequently revealed widespread examples of the issue, with the potential for significant, negative consequences for its victims (Chatzakou et al., 2017a; Balakrishnan et al., 2019; Sterner, 2017). Twitter has taken several steps to combat cyberbullying, including blocking unwelcome messages from users who do not have a profile picture and enabling a time-out function that bans users who use abusive language, among other things. Despite these excellent efforts, cyberbullying is still a problem on the network (Bernazzani, 2017; Twitter, 2019).

In most recent case, Shubham et al. (2021) proposed a Bi-directional long short-term memory (BLSTM) model to detect cyberbullying in tweets. After testing the model on 35,787 labelled tweets, the GloVe840 word embedding technique and BLSTM provided the best results on the dataset with accuracy, precision, and F1 measure of 92.60%, 96.60%, and 94.20%, respectively. However, the feature extraction technique (GloVe840) used is based on the use of global word-to-word co-occurrence counts over the entire corpus, hence, there is a need for improvement.

1.5 Scope of Study

The scope of this study is focused on only research. The implementation of a system will be ignored for now.

1.6 Justification of the study

Given the rapid advancement of computer technology as well as the rapid growth of social networking, it has become necessary to look into detecting cyberbullying to prevent victims of cyberbullies from cyberbullies. The cyberbullying detection system will detect and prevent cyberbullying on Twitter which will help prevent the effects of cyberbullying.

To provide furtherance in the research area of machine learning and cyberbullying;

1.7 Methodology

Two or more cyberbullying datasets will be collected and manually put together to form a larger dataset. Feature extraction method (word2vec) will be used to extract features of the dataset and change the words into vectors. For the model training and testing, six classifiers (Logistic Regression, Decision Tree, Random Forest, XGBoost, AdaBoost and Naïve Bayes) will be used. The performance of the model will be checked using an accuracy score and other commonly used evaluation metrics.

1.8 Expected Results and possible use of study

At the end of this research, the proposed system should be able to detect cyberbullying on Twitter tweets with accuracy score higher than the previous papers.

1.9 Presentation of thesis

This study is organized as follows:

Chapter one: In this chapter, the general introduction, statement of the problem, the background to study, the research objectives and significance of the study, the scope of the study,

the definition of terms, and the organization of chapters are present.

Chapter two: This chapter is the literature review phase of this paper.

Chapter 3: This chapter gives detail explanation on how the research problems will be solved.

Chapter 4: This chapter explains the experimentation and analysis of results.

Chapter 5: Give the conclusion of the research work

CHAPTER TWO (LITERATURE REVIEW)

With the introduction of various sorts of social media platforms, social media's acceptance increased. As these online platforms have grown in popularity, cyberbullying has increased as well. This section looks at some of the most important works that have attempted to address cyberbullying using machine learning and deep learning techniques.

Haidar et al., (2017) proposed a Naïve Based and Support Vector Machine model that identifies cyberbullying in tweets focusing on English and Arabic languages. The authors tested the model on Facebook and Twitter data, and it yielded performance readings of 90.1 % and 93.4%, respectively. Also, in *Van Hee et al. (2018)*, the authors proposed a linear kernel Support Vector Machine model to identify cyberbullying in social media texts. After testing the model on social media text, it yielded an F_1 Score of 64%. However, both Naïve Bayes and Support Vector Machine are unable to handle huge volumes of data. Therefore, there is a need for improving this model to permit a higher bullying detection.

In *Hani et al. (2019)*, the authors used Neural Network and Support Vector Machine as classifiers to detect cyberbullying in tweets. The authors tested the model on 12773 conversations messages collected from Formspring; it yielded an accuracy of 92.8% and 90.3%, respectively. In extracting the features of the data for training, Term Frequency Inverse Document Frequency (TFIDF) and sentiment analysis were used. However, these feature extraction algorithms focus on only the sentiments (whether a word is negative or positive) and the importance of words to a document, neglecting the semantics and contextual meaning of the words. Hence, there is a need to improve the feature extraction algorithm to consider the contextual meaning and semantics of the words.

Rahul et al. (2020) proposed a Support Vector Machine and a Naïve Bayes to detect cyberbullying in tweets. After testing the model on Twitter tweets, the learning algorithms noticed cyberbullying with 52.70% and 71.25% accuracy, respectively. There is a need to improving these models to permit a higher bullying detection since both Naïve Bayes and Support Vector Machine cannot handle vast volumes of data. Also, in extracting features of the dataset for training the model, Term Frequency Inverse Document Frequency (TFIDF) vectorizer, which focuses on the importance a word is to a document, was used. This vectorizer neglects the semantics and contextual meaning of words. Therefore, there is a need to improve it.

Prajakta et al. (2020) proposed a Support Vector Machine and a Convolutional Neural Network model to detect cyberbullying in tweets. After testing the models on Twitter tweets, it yielded an accuracy of 84% and 90.23%, respectively. However, SVM cannot handle a vast amount of data. Hence, there is the for improving it.

In *Amgad and Suliman (2020),* the authors proposed seven models (Logistic Regression (LR), Light Gradient Boosting Machine (LGBM), Stochastic Gradient Descent (SGD), Random Forest (RF), AdaBoost (AB), Naive Bayes (NB), and Support Vector Machine (SVM)) to make a comparative analysis in detecting cyberbullying in tweets. The authors tested the model on Twitter tweets, and the experimental results showed the superiority of LR, which achieved an accuracy of around 90.57%. Also, in extracting features, Term Frequency Inverse Document Frequency (TFIDF) and Word2vec were used. But, TFIDF focuses on how important a word is to a document, and Word2Vec cannot handle unknown or out-of-vocabulary (OOV) words; hence, there is a need to improve the feature extraction algorithm.

Yuvaraj et al. (2021) proposed an Artificial Neural Network-Deep Reinforcement Learning model to detect cyberbullying in tweets. After testing the model on Twitter tweets, the model yielded an accuracy of 81.688%. However, ANN is false tolerance (that is, gives output even when part of the dataset

is corrupt), which makes its prediction inaccurate; hence, it needs to be improved.

In *Jaithunbi et al. (2021)*, the authors proposed a Support Vector Machine with linear kennel and Naïve Bayes model to detect cyberbullying in tweets. The authors tested the model on social media datasets collected from Kaggle, GitHub, etc., which yielded 71.25% and 52.70%, respectively. However, both SVM and Naïve Bayes cannot handle vast datasets; thus, the models need to be improved. The authors used TFIDF vectorizer as a feature extraction technique. TFIDF focuses on the importance of a word to a document and neglects the semantics and contextual meaning of the word. Hence, the need for improvement.

Rounak and Siddhartha (2021) proposed various classifiers like Support Vector Machine (SVM), Logistic Regression (LR), Random Forest, and Passive Aggressive (PR), which were used as an experiment to determine which classifier has the highest accuracy in detecting cyberbullying in tweets. After testing the model with Twitter tweets, the result shows that the Passive-Aggressive classifier had the highest accuracy (78.1%) when used with N-gram level TFIDF feature extraction. Whereas when using word-level feature extraction, Support Vector Machine performed the best (60.3%) compared to others.

In most recent case, Shubham et al. (2021) proposed a Bi-directional long short-term memory (BLSTM) model to detect

cyberbullying in tweets. After testing the model on 35,787 labelled tweets, the GloVe840 word embedding technique and BLSTM provided the best results on the dataset with accuracy, precision, and F1 measure of 92.60%, 96.60%, and 94.20%, respectively. However, the feature extraction technique (GloVe840) used is based on the use of global word-to-word co-occurrence counts over the entire corpus, hence, there is a need for improvement.

CHAPTER THREE (PROPOSED FRAMEWORK)

This section discusses strategies that might be useful in filling in the research gaps indicated in chapter 2. The proposed study will be broken down into sections detailing dataset description, data preprocessing, data split, and the machine learning methods used to create baseline models. The theoretical framework or architecture of this study is shown in the figure below.

3.1 Dataset Description

There is no unique dataset to be used in detecting cyberbullying. This study's dataset will be derived from two separate sources. The final dataset will be created by merging them using the pandas library's concatenation function.

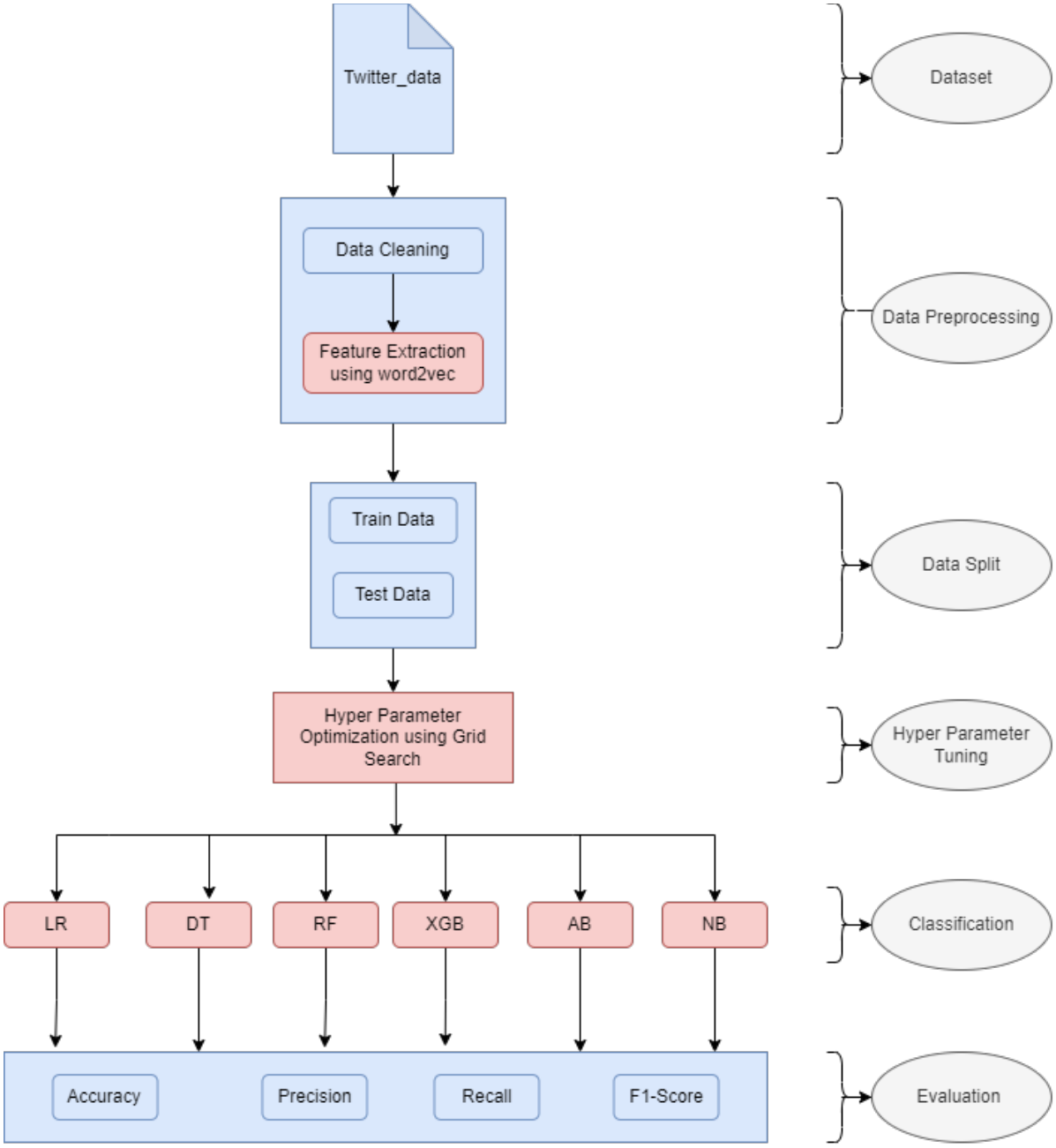

Figure 1: Proposed Framework

3.2 Data preprocessing

The data preprocessing phase is one of the most critical phases. In this phase, the data is cleaned to remove repetitive data and unwanted punctuation marks. Again, all the words in the data are changed to lower cases. Learning algorithms perform better with numbers than words. Since the

dataset used in this study is made of words, a feature extraction algorithm will be used to change the words to vectors. Word2vec, which takes the semantic closeness of words and how words are used in context into consideration, will be used as feature extraction algorithms.

3.3 Data Split

In the data split phase, the dataset will be split into training and testing datasets. Models make more accurate predictions based on the size of the training dataset. Due to this, 80% of the dataset will be used for training, and the remaining 20% will be used for testing. __

3.4 Classification

Six classifiers (Logistic Regression, Decision Tree, Random Forest, XGBoost, AdaBoost and Naïve Bayes) are used to create models for training and predicting of cyberbullying actions on Twitter. These models are used to perform a comparative analysis to determine which of the classifiers performs best in detecting cyberbullying on Twitter. Below is the detail explanation of each classifier.

a) ***Logistic Regression (LR)***: The supervised learning classification algorithm logistic regression is used to predict the likelihood of a target variable. Because the nature of the goal or dependent variable is binary, there are only two classifications. In simple terms, the

dependent variable is binary in nature, with data represented as 1 (representing success/yes) or 0 (representing failure/no). A logistic regression model predicts P(Y=1) as a function of X mathematically. It's one of the most basic machine learning algorithms, and it may be used to solve a variety of classification problems like spam detection, diabetes prediction, cancer diagnosis, and so on. The linear function is essentially utilized as an input to another function in logistic regression, such as g in the following relationship.

$$h_\theta(x) = g(\theta^T x) \; where \; 0 \le h_\theta \le 1 \qquad (1)$$

The logistic or sigmoid function, g, can be written as follows:

$$g(z) = \frac{1}{1+e^{-z}} \; where \; z = \theta^T x \qquad (2)$$

b) *Decision Tree (DT)*: Decision tree analysis is a predictive modeling approach that can be used in a variety of situations. An algorithmic strategy that can split the dataset in numerous ways based on different conditions can be used to create decision trees. The most powerful algorithms in the domain of supervised algorithms are decision trees.

c) ***Random Forest (RF)*:** Random Forest is a supervised learning technique that can be used to classify and predict data. However, it is mostly employed to solve categorization issues. A forest, as we all know, is made up of trees, and more trees equals a healthier forest. Similarly, the random forest method constructs decision trees from data samples, extracts predictions from each, and then votes on the best option. It's an ensemble method that's superior than a single decision tree because it averages the results to reduce over-fitting. The diagram below shows how RF works.

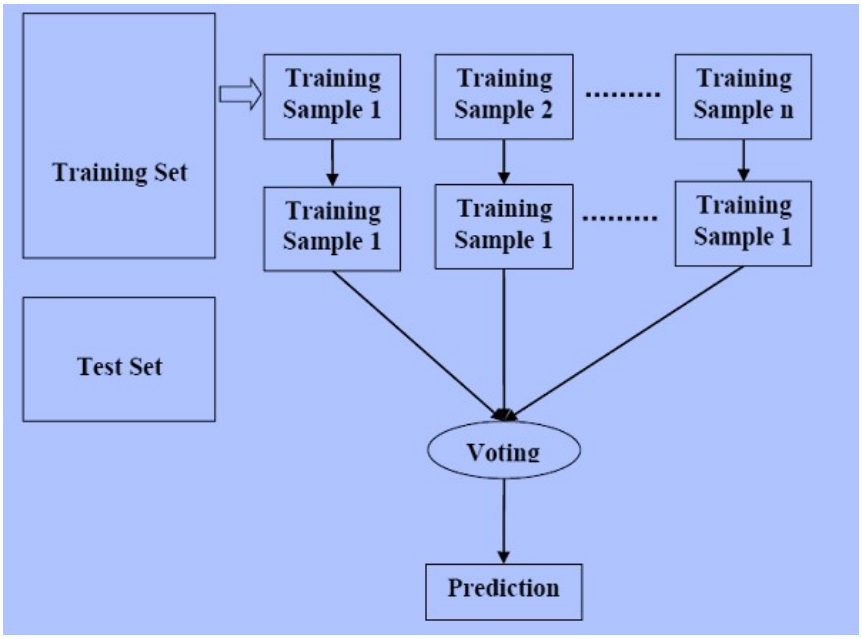

Figure 2: Structure of Random Forest

d) ***XGBoost (XGB)*:** Under the Gradient Boosting framework, XGBoost is an open-source software package that implements optimal distributed gradient boosting machine learning methods. Extreme Gradient Boosting (XGBoost) is

a distributed gradient-boosted decision tree (GBDT) machine learning toolkit that is scalable. It is the top machine learning library for regression, classification, and ranking tasks, and it includes parallel tree boosting. To understand XGBoost, you must first understand the machine learning ideas and methods on which it is based: supervised machine learning, decision trees, ensemble learning, and gradient boosting.

e) ***AdaBoost (AB)*:** AdaBoost, also known as Adaptive Boosting, is a Machine Learning approach that is employed as part of an Ensemble Method. Decision trees with one level, or Decision trees with only one split, are the most popular algorithm used with AdaBoost. Decision Stumps is another name for these trees.

   This approach creates a model by assigning equal weights to all of the data points. It then gives points that are incorrectly categorized a higher weight. In the next model, all points with greater weights are given more importance. It will continue to train models until a smaller error is received.

f) ***Naïve Bayes (NB)*:** It's a classification method based on Bayes' Theorem and the assumption of predictor independence. A Naive Bayes classifier, in simple terms, posits that the existence of one feature in a class is unrelated to the presence of any other feature.

The Naive Bayes model is simple to construct and is especially good for huge data sets. Naive Bayes is renowned to outperform even the most advanced classification systems due to its simplicity.

The Bayes theorem allows you to calculate posterior probability P(c|x) from P(c), P(x), and P(x|c) using P(c), P(x), and P(x|c). Consider the following equation:

$$P(x|c)=\frac{P(x|c)P(c)}{P(x)} \quad (3)$$

- $P(c|x)$ is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

3.5 Hyper Parameter Tuning

The task of selecting a set of ideal hyperparameters for a learning algorithm is known as hyperparameter optimization or tuning. A hyperparameter is a value for a parameter that is used to influence the learning process. To optimize the classifiers used in this paper, the Grid Search algorithm, an approach to hyperparameter tuning, is used to automatically select the best parameters for the learning algorithms.

- Grid Search Algorithm: Grid search is the most basic hyperparameter tuning approach. In a nutshell, the hyperparameters' domain is divided into a discrete grid. Then, using cross-validation, we try every possible combination of values in this grid, calculating various performance measures. The ideal combination of values for the hyperparameters is the point on the grid that maximizes the average value in cross-validation. Grid search is a comprehensive technique that considers all possible combinations in order to locate the best point in the domain.

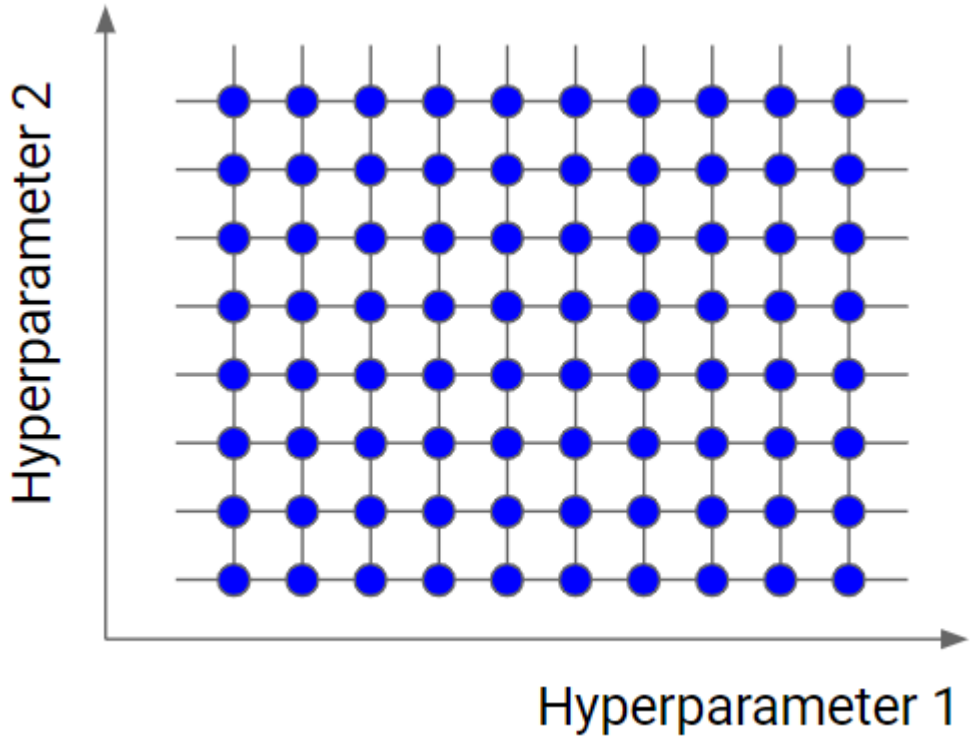

Figure 3: Example of grid search

3.6 Evaluation

In this phase, the performance of the model is checked. The accuracy score, precision other commonly used evaluation metrics will be used to determine the performance of the model.

a. Accuracy: Accuracy is a metric for evaluating a model that allows you to count how often of its predictions are correct. The following is the formula for accuracy:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (4)$$

- TP is true positive
- TN is true negative
- FP is false positive
- FN is false negative

b. Precision: The precision of a model in predicting positive labels is measured. Precision solves the question of how often a model was correct out of the number of times it predicted a positive outcome. The percentage of your findings that are relevant is known as precision. The following is the precision formula:

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

c. Recall: The percentage of actual positives that a model properly detected is calculated by recall (True Positive). You should employ recall when the cost of a

false negative is large. The recall formula is as follows:

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

d. F1 Score: The F-score or F-measure is a measure of a test's accuracy in binary classification statistical analysis. It is calculated using the test's precision and recall, with precision equaling the number of true positive results divided by the total number of positive results, including those that were incorrectly identified, and recall equaling the number of true positive results divided by the total number of samples that should have been identified as positive. The F1 Score is as follows:

$$F1\,Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (6)$$

CHAPTER FOUR (EXPERIMENTATION AND ANALYSIS OF RESULTS)

A. Experimentation

To detect and predict cyberbullying on Twitter, our dataset was imported into jupyter notebook for preprocessing and classification.

For preprocessing stopwords and word lamentizer were used to clean the data. Also, word2vec was used to extract the best features from the dataset.

For classification, six machine learning algorithms were used for training and making predictions. Also, Grid Search was used to optimize the classifiers used in our work.

B. Analysis of Results

Following the review of past studies on cyberbullying detection, several key research points emerged. The research seeks to answer the following questions:

RQ1: What is the effect of considering both semantics and context of words on the performance of the proposed method.

RQ2: What is the impact of the optimization of algorithms on the performance of the proposed method. This paper explores (RQ2) whether the learning algorithm used can be optimized.

A. Performance of the baseline models of the base paper with no feature extraction

Algorithm/Metrics	Accuracy	Precision	F1 score	Recall
Decision tree (Quinlan, 1986)	90.59	93.34	92.79	92.26
Naive Bayes	66.86	92.22	68.18	54.09
Random forest (Breiman, 2001)	91.35	94.39	93.30	91.72
XgBoost (Chen and Guestrin, 2016)	88.79	98.54	90.79	84.18
SVM (Schölkopf, 1998)	91.48	96.18	93.31	90.62
SVM (rbf)	91.67	96.14	93.48	90.96
Logistic regression	92.37	96.07	94.06	92.14

Table 1: The performance of the baseline models of the base paper with no feature extraction.

B. Performance of LR with different types of embedding/feature extraction techniques

Embedding type	Accuracy	F1 score	Precision	Recall
One-hot encoding	89.25	90.92	93.08	91.90
GloVe 6	89.97	92.47	91.49	93.48
GloVe 42	91.95	93.66	96.41	91.06
GloVe 840	92.60	94.20	96.60	91.92

Table 2: Performance of base paper model using different embedding techniques

A. Effect of considering both semantics and context of words on the performance of the proposed method (RQ1)

The performance of the models when only word2vec is applied on the dataset used in this paper is shown in the table below.

%	Logistic Regression	Decision Tree	Random Forest	XGBoost	AdaBoost	Naïve Bayes
Accuracy	93.45	89.32	92.19	93.61	92.65	88.41
Precision	92.87	89.69	91.82	93.12	91.89	90.29
Recall	93.45	89.32	92.19	93.61	92.65	88.41
F1-Score	92.77	89.50	90.36	92.83	92.02	89.18

Table 3: Performance of models when word2vec was used for feature extraction.

It is shown clearly on *Table 3* that, XGBoost performed better than the other learning algorithms, yielding an accuracy of 93.12%. The naive Bayes method, according to the results, performed the worst with an accuracy of 88.41% of all the classifiers.

To show that our feature extraction technique performs better than that of the base paper's, we made a comparison on the performances of the various feature engineering techniques used in the base paper and our work. The chart below shows the effect of various feature engineering techniques on Logistic Regression.

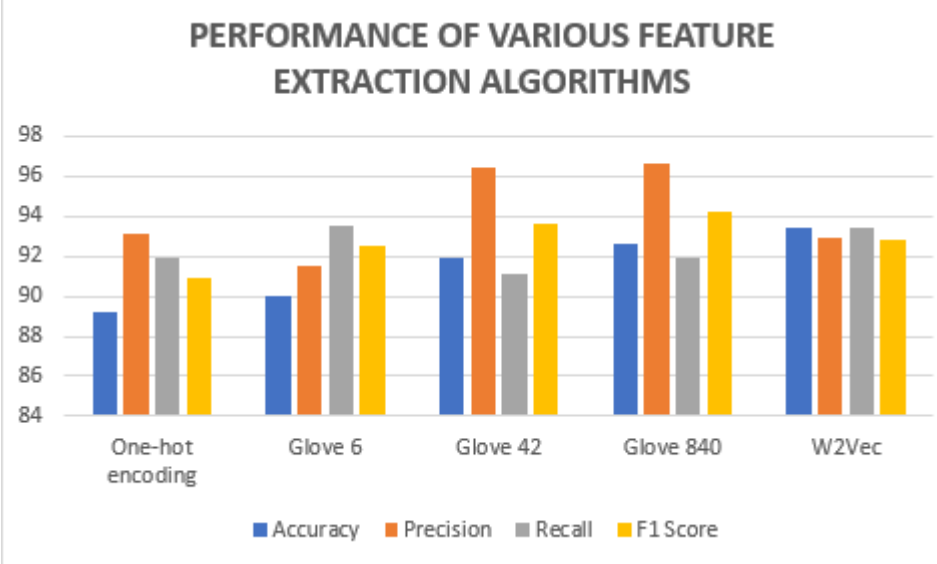

Figure 4: Performance of feature extraction algorithms on LR

From figure 4, we can see that w2vec feature engineering had the highest performance on LR with an accuracy of 92.77%.

B. Effects of feature extraction and hyper-parameter tuning on performance (RQ2)

%	Logistic Regression	Decision Tree	Random Forest	XGBoost	AdaBoost	Naïve Bayes
Accuracy	93.55	92.60	93.74	94.32	93.61	91.41
Precision	92.99	91.78	93.45	93.91	93.09	90.74
Recall	93.55	92.60	93.74	94.32	93.61	91.41
F1-Score	92.92	91.84	92.83	93.89	93.18	91.00

Table 4: Performance of models when word2vec and grid search were used

Table 4 shows the effect of performing feature extraction on the dataset and hyper-parameter tuning on the models. From the table, XGBoost performed better than the other learning algorithms, yielding an accuracy of 94.32%. The naive Bayes method, according to the results, performed the worst with an accuracy of 91.41% of all the classifiers. It's possible that this is due to a high number of false negatives, which reduced the algorithm's accuracy and recall. In terms of precision, XgBoost was the best, with very few false positives.

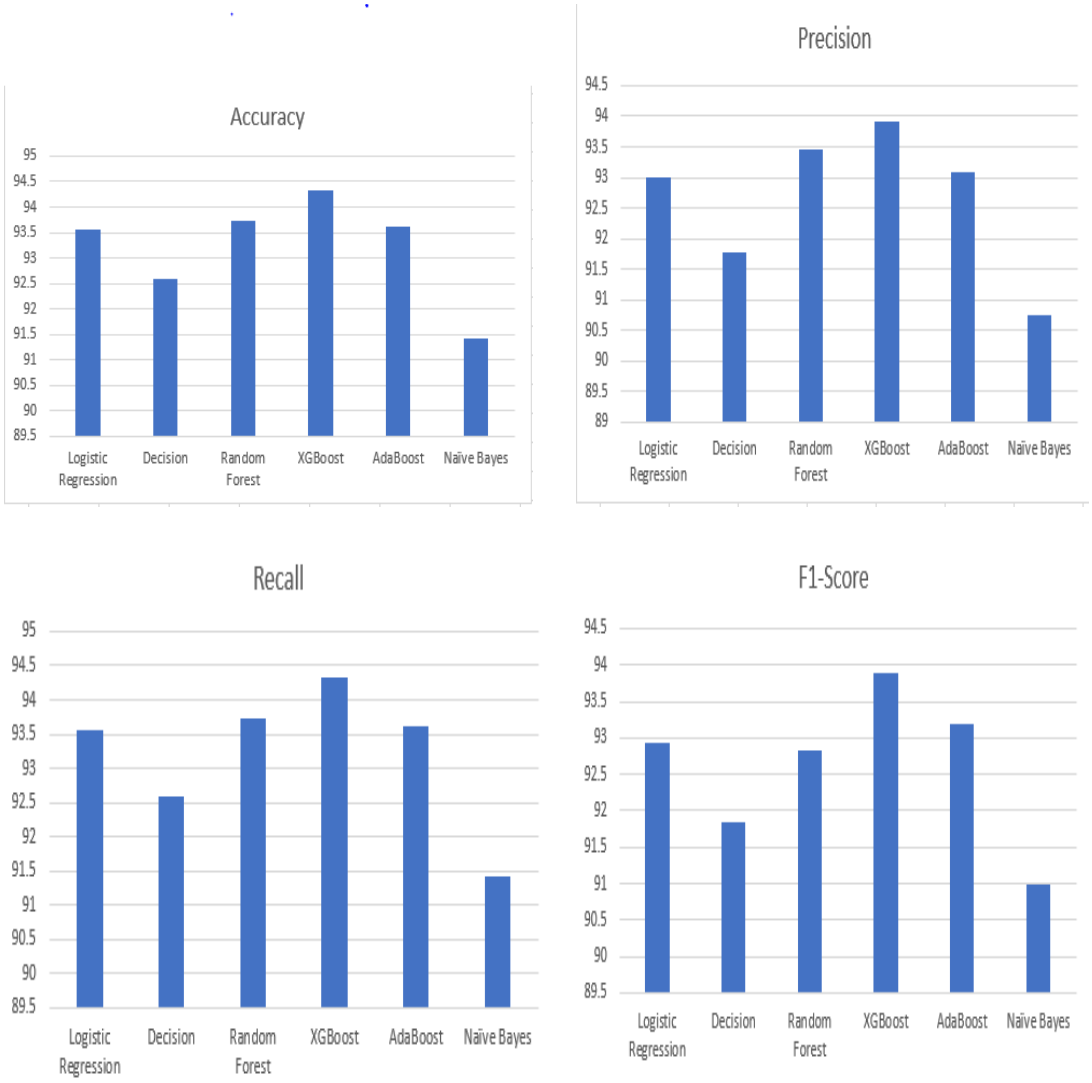

Figure 5: Performance of proposed scheme

CHAPTER FIVE (CONCLUSION AND RECOMMENDATIONS)

A. CONCLUSION

Bullying is common on social media, which is where many young people engage with one another. In this paper, six classification algorithms (Logistic Regression, Decision Tree, Random Forest, XGBoost, AdaBoost and Naïve Bayes) are used as models in detecting cyberbullying on Twitter.

A comparative analysis is done to find out which model best detect cyberbullying, and the effects of feature extraction and hyper-parameter optimization on the performance of these models. The results show that when feature extraction is performed on dataset and the hyper-parameters of classifiers are optimized, the performance of the classifiers increase. The classifiers used in this paper was evaluated on a dataset obtained from Kaggle and Random Forest, XGBoost and AdaBoost had the best accuracy of 94%.

B. RECOMMENDATIONS

i. Creation of more contextual datasets

ii. Exploitation of Graph embedding for feature extraction to suggest improvement on extracting both context and semantics. Several research works have seen improvement on base models when various forms of graph embedding techniques were used for feature extraction. For example, Lu et al. (2021) used low-rank adaptive graph embedding (LRAGE) to explore the underlying correlation structure of data and learn more informative projection.

iii. Consider evolutionary algorithms such as Genetic algorithms, PSO, grey wolf algorithms in improving the performance of learning algorithms. Several research works have seen improvement on the base models when evolutionary algorithms were

integration. For example, in the work of Zhu et al. (2021), PSO was used to improve the performance of LeNet-5 to realize intelligent fault diagnosis of hydraulic piston pump.