iJRASET
International Journal For Research in
Applied Science and Engineering Technology

Volume: 6 Issue: IV Month of publication: April 2018

DOI: http://doi.org/10.22214/ijraset.2018.4396

Comparative Analysis of Different Machine Learning Algorithms to Detect Cyber-bullying on Facebook

Dipika Jiandani[1], Riddhi Karkera[2], Megha Manglani[3], Mohit Ahuja[4], Mrs. Abha Tewari[5]
[1, 2, 3, 4, 5]Department of Computer Engineering, VESIT Mumbai, India

Abstract: Offensive language on social media has unfortunately become a common occurrence among users. The motive is to detect offensive language in a user message, post or comment and take necessary actions for the same. This is called as offensive language filtering. In this paper, we provide a comparison of different algorithms to build a solution through which Facebook users can find their cyber bullies and report them. The entire process consists of six stages: data collection, pre-processing, sessionization, ground truth, feature extraction and classification. Using machine learning algorithms for pre-processing and classification of the data and tools like Facepager and Pycharm, we have evaluated the processing, usage and accuracy of three major classification algorithms which are Naive Bayes, Support Vector Machine and Neural Networks.

Keywords: Naïve Bayes, Support Vector Machine, Neural Networks, Facebook, Facepager, tokenization, word-sense disambiguation, sentiment analysis, subjectivity, polarity, cyber-bullying.

I. INTRODUCTION

Anomaly based forensic analysis of social media refers to analysis on Facebook data using machine learning algorithms. The aim is to build a web application that will proactively detect and report cases of cyber-bullying and personal security intrusion on social media platforms (here, Facebook) using machine learning algorithms and behavioural analysis. The sub goals of our project are data collection, pre-processing, sessionization and crowd sourced labelling. The application can be used for identifying theft, theft of public data, public defamation, cyber stalking, bullying and other criminal activities on such sites. The anonymous nature of social networking applications can be leveraged by malicious users. Our focus is on conduction of forensic analysis on one of the most popular social media applications in the recent times, i.e. Facebook. The application has the ability to stop cybercrimes happening at a full-fledged rate.

II. PROBLEM DEFINITION

This project focuses on conducting forensic analyses on some of the widely used social networking applications like Facebook, Instagram to name a few. This analysis will be aimed at analysing offensive comments with the motivation of cyber-bullying posted on these applications and backtracking them to the offender. The extent, significance, and intention of the data that could be found and retrieved. If so, the suspect will be found guilty of a cybercrime since there will be a solid evidence to prove the activity was performed by him. This application includes pre-processing and analysis of data via various models of Machine Learning like Naive Bayes, Support Vector Machine and Artificial Neural Networks. The development of such an application has the ability to stop and reduce the rate of subjugating that has been happening online at a full-fledged rate. Our goal is to compare three classification models used in the development of a web application that will proactively detect and report cases of cyber-bullying and personal security intrusion on social media platforms like Facebook, Twitter, etc. using the concepts of behavioural analysis and machine learning. Further, a block action or report will be generated on the basis of the supporting evidence found through Forensic Analysis of social media sites First, a tool called “Facepager” is used to get the data from Facebook. It gives access to different posts, pictures, comments and emoticons of various public profiles on Facebook using which the training and testing data are formed. With the help of pre-processing algorithms, the raw data is converted into executable form. The pre-processed data is then classified using the machine learning algorithms- naive Bayes, support vector machines and neural networks for sentiment analysis. A comparison of these three classification algorithms is done on the basis of the processing, performance and the accuracy of each of the them. The polarity and subjectivity of each algorithm is found and plotted on a graph to compare. Also, based on the frequency of the bad words, a word cloud is generated. The bad word which has the highest frequency will have a bigger size compared to the words whose frequency is less.

III. METHODOLOGY

A. Dataset

The classification of comments plays a crucial role in the process of filtering the comments. Such a classification would help in the categorization of numerous online content into offensive and clean comments reducing the pressure on human monitoring. We have developed and made use of two datasets namely the trained dataset and testing dataset for the purpose of effectively classifying the comments. Training a dataset prior to running the codes on the testing dataset helps to find the potential negative comments for further processing of dynamically obtained comments.

The training dataset is hatebase.csv, which is the static dataset obtained from hatebase.org, a Canadian website that provides a crowdsourced, multilingual corpus consisting of a repository of words and phrases indicating hate speech. This site is dynamically updated with new additions every single day. In addition to the data, the hatebase.csv file also consists of a column that contains the values indicating their degree of offensiveness. The trained dataset is coded with threshold being a fixed value indicating positive and negative comments on either side.

The test dataset is fetched using a data crawler tool called Facepager. It is a Facebook Graph API dependent tool, that can be used to extract Facebook data in the form of posts, photos, videos, group activity and the most import element in our case, comments. Once the comments are extracted from post(s) from the particular user’s profile, it is exported to csv. This now becomes the testing dataset.

The comments that are thus obtained could be in the form of structured, semi-structured or completely unstructured text. It is necessary to handle all three kinds of data. For this purpose, pre-processing is important.

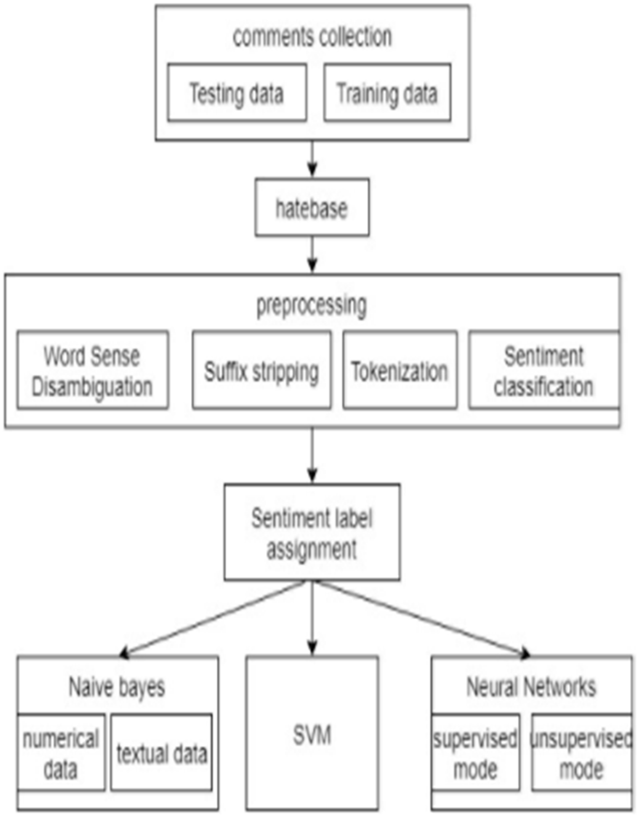


Fig. 1 Methodology

B. Pre-processing algorithms

1) *Word Sense Disambiguation:* In any language, same words can mean different things with respect to a particular context or reference. For example, the word 'bow' can mean the act of bending forward and it also refers to an archery equipment. To know the exact meaning of the sentence, we need to eliminate the ambiguity. For this purpose, we used the Naive Bayes classifier along with a unique keyword identifier.

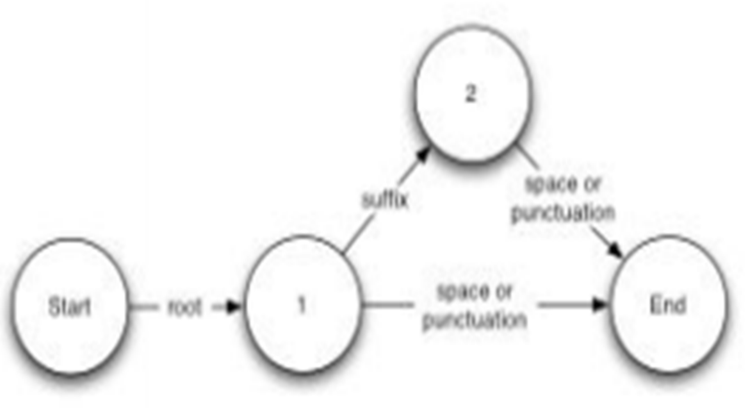


Fig. 2 Word Sense Disambiguation

2) *Tokenization:* Tokenization is a process of slicing the data into the smallest possible unit. The Facebook comments are divided into single word units and stored as a csv file. Tokenization has two steps. First, the text is tokenized into sentences. And then the sentences are tokenized into words. Tokenization is followed by stop words removal and lemmatization. For example, consider the sentence, "John Doe is an architect.". This will produce the following tokens: 'John', 'Doe', 'is', 'an', 'architect'.
3) *Stop-words removal:* A stop word is a commonly used word like in, the, an, for, of, etc. We certainly do not want these words to take up space in our database or utilize the processor. Hence, we eliminate these words along with the punctuation marks. For this purpose, we maintain a database of characters and words that we consider as stop words based on the document frequency of each word.
4) *Sentiment Analysis:* Sentiment analysis, also known as opinion mining, is used to find out the opinions and sentiments about some topic. People use microblogging websites like Facebook, Twitter, Instagram to express their opinions. We used Facebook to get our training data using the Facepager application. We extracted the comments and classified them into positive, negative and neutral comments. Using SVM, Naive Bayes and Neural networks, a comparison of the processing and the accuracy of each algorithm is evaluated.

C. Offensive comments classification

Subjectivity- Subjectivity is a measure which tells us whether the data is subjective or objective. A subjective may or may not express feelings and emotions. For example, "i like monsoons" is a subjective statement and does express some feelings. But the sentence "I want to go home" does not depict any kind of emotions and is still considered to be subjective data. Polarity- Polarity describes the type of emotions expressed in the data. It tells us if the data is positive, negative or neutral. Generally, the intensity of emotions determines the strength of a sentiment, for example, "1+ series are the best phones available in the mobile market" depicts a positive emotion, whereas "The services provided by Toyota are horrible" showcase negative emotions.

1) *Naive Bayes Algorithm:* Our main goal is the classification of comments into offensive and clean i.e. it is a binary classification and partly subjective classification with respect to their sentiment and subject matter. We used the powerful scikit-learn library in Python for this purpose. This library is better than nltk because where nltk only supports Gaussian based Naïve Bayes, scikit-learn supports its multinomial distribution. This library can be downloaded like any other library in Python by simply using the "pip install" command. In addition to scikit-learn, other libraries imported for smooth processing were nltk, csv, numpy, pandas, genism, etc. Pycharm SDK is the platform used for coding.

The Naïve Bayes method is previously known to be an effective machine learning algorithm pertaining to the classification of spam content. It classifies both numerical and textual data. One of its major features is that it believes in independence between any pair of feature points. Some of its main advantages over other classification methods like SVM, Decision trees are higher training efficiency, quicker convergence to solution, comparatively easier implementation and large vocabulary-oriented data handling. In one pass of the testing data, it first computes the conditional probability of individual features with reference to the test dataset. Following that, it applies the Bayes theorem to obtain the posterior probability.

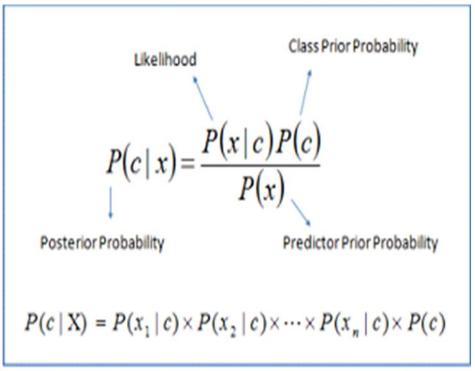

Likelihood
Class Prior Probability
$P(c|x)=\frac{P(x|c)P(c)}{P(x)}$
Posterior Probability
Predictor Prior Probability
$P(c|X)=P(x_1|c)\times P(x_2|c)\times\cdots\times P(x_n|c)\times P(c)$

Fig. 3 Naive Bayes equation

Using the Naïve Based Classification method, we could obtain about 79% accuracy. However, it largely depends on the quality and quantity of training and testing data. Once the classification is done, the results can be stored back into a csv file, to be backtracked to the original comment. This method did show some inefficiencies. The Naïve Based classifier will not be as effective for features that are highly dependent like short texts. Further, conditional independence assumption cannot be wholly relied on in real world data.

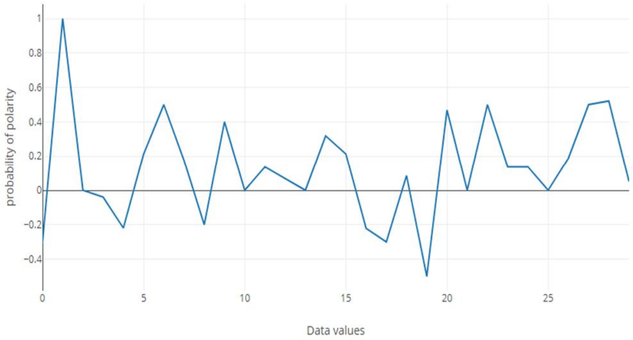


Fig. 4 Polarity using naive Bayes model

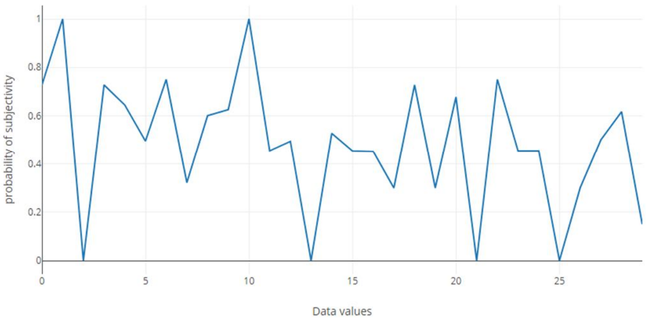
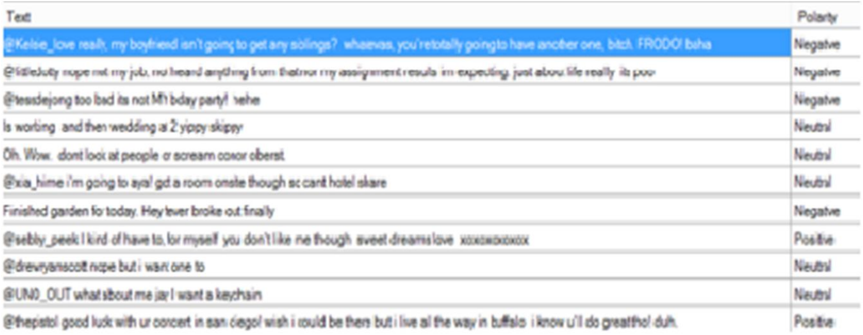


Fig. 5 Subjectivity using Naive Bayes

2) *Support Vector Machine Algorithm:* It is a machine learning algorithm is used for categorization of huge amount of text. The main motive is to find a hyperplane that separates vectors in one class from vectors in other classes. The technique which is used here is called as the kernel trick. It is used to transform our training data and then based on these transformations, an optimal boundary is found out between the possible outputs. We used support vector machine algorithm to categorize and estimate the strength of positive and negative sentiments of the comments. The experiments are performed on the dataset containing comments of a Facebook page. Naïve Bayes classifier is better than SVM in sentiment classification. Because SVM doesn't perform well when the dataset is too large or if it contains noise. The preliminary results don't seem to be promising as it gives an accuracy of 58% for this training dataset which less than the accuracy percentage of Naïve Bayes Algorithm.

TABLE I
Sentiment Labeling Using Svm



Text	Polarity
@Kelsie_love really, my boyfriend isn't going to get any siblings? whatevas, you're totally going to have another one, bitch. FRODO! haha	Negative
@littledutty nope not my job, not heard anything from thatnor my assignment results im expecting, just about life really its poo	Negative
@tessdejong too bad its not Mh bday party! hehe	Negative
Is working and then wedding at 2! yippy skippy	Neutral
Oh. Wow. dont look at people or scream conor oberst	Neutral
@xia_hime i'm going to ayal gd a room onsite though sc cant hotel share	Neutral
Finished garden fo today. Hey fever broke out finally	Negative
@selby_peek I kind of have to, for myself you don't like me though sweet dreams love xoxoxoxoxox	Positive
@drewryanscott nope but i want one to	Neutral
@UN0_OUT what about me jay I want a keychain	Neutral
@thepistol good luck with ur concert in san diego! wish i could be there but i live all the way in buffalo i know u'll do great tho! duh.	Positive

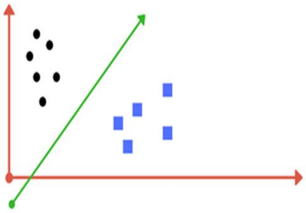
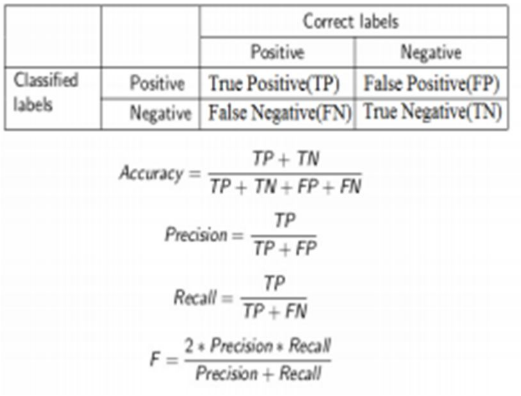


Fig.6 Sample cut to divide into two classes

The performance parameters after performing sentiment analysis on the training data uses the following formula:

TABLE II
Performance Parameters



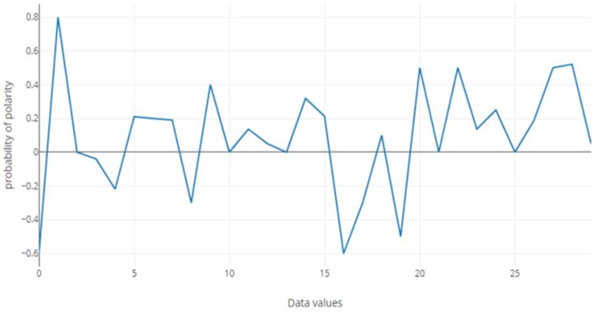
		Correct labels	
		Positive	Negative
Classified labels	Positive	True Positive(TP)	False Positive(FP)
	Negative	False Negative(FN)	True Negative(TN)

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

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

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

$$F = \frac{2 * Precision * Recall}{Precision + Recall}$$



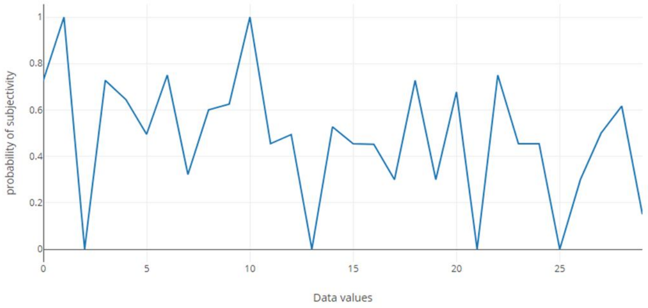


Fig. 7 Polarity using SVM

Fig. 8 Subjectivity using SVM

3) *Neural Networks:* Neural network is one of the many techniques used in machine learning and is also known as Artificial Neural Networks (ANN). The main element of ANN is the novel structure of the information processing system, which is influenced by the way information is processed by biological nervous systems and the brain. It is made of an enormous number of highly interconnected units known as 'neurons', processing in coordination to execute a particular task. Artificial neural networks learn through examples, just like people. For eg, pattern recognition and data classification.

There are various deep neural network models available for data classification. We made use of the unsupervised model for our dataset. The main benefit of using the unsupervised mode is word2vec, that enables us to create a low dimension distributed representation of data. Skip-gram and continuous bag of words are the two popular models used in this mode. We decided to go to with skip-gram as it has a very simple and straightforward design.

The primary logic behind skip gram is that it considers each word in a huge corpus and simultaneously also takes one word which surrounds it within a specific defined 'window'. It then trains the neural network in such manner that it will predict the probability of each word to occur in the window around the focus word. To make things convenient, we create a vocabulary of various words available in our data set. We then encode this data as a vector that has the same dimensions as our vocabulary. For eg, if we have a vocabulary made out of the words “the”, “big”, “black”, “bear”, “eats”, “the” “white”, “goat”, the word “bear” is represented by this vector: [0, 0, 1, 0, 0, 0 ,0 ,0].

Based on predicting the nearby training words, a model depiction of current word is generated in the training process. After training, the word embedding is created which is the vector of weights from the hidden layer.

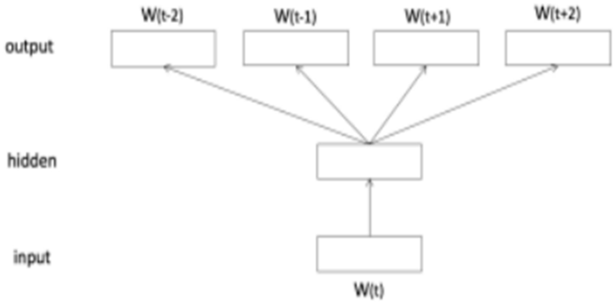


Fig. 9 Primary logic behind Neural Networks

W represents the set of vectors containing the words and the output layer predicts the next word in the scenario. Since the input is now ready, we use it to enter into the 2-layer neural network model. The probability for each text in the vocabulary will be displayed in the second layer.

The unsupervised mode gives us approximately an accuracy of 58% when we use the skip gram model. The Skip gram model reduces 39% of the errors that occur during the implementation Continuous Bag of Words Model.

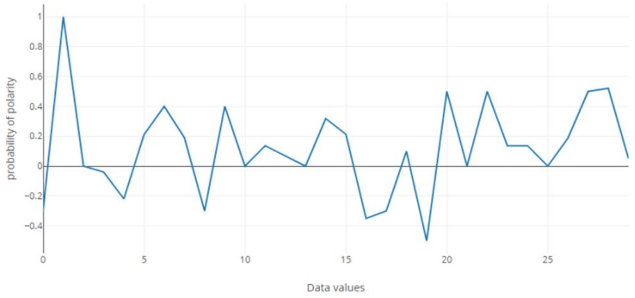


Fig. 10 Polarity using Neural Networks

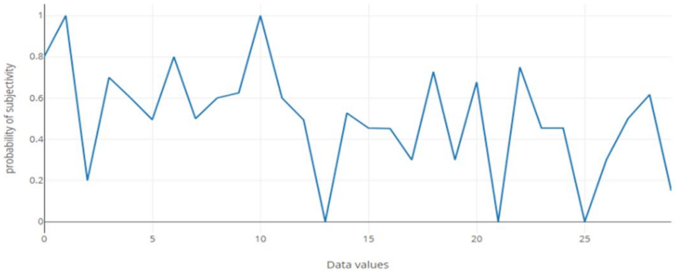


Fig. 11 Subjectivity using Neural Networks

IV. RESULTS

A. Word Cloud

The fundamental idea of creating a word cloud is to represent a set words on a canvas, where its size depends on its importance. In our case, greater the size indicates higher degree of profanity usage. Word clouds, although a little old-fashioned, gave an interesting visualization of our model. In Python, word cloud can be generated with aid from some libraries like nltk, CountVectorizer or with a regular expression. We have used the Wordcloud library. First, we read the data in the csv format. Our goal is to randomly sample a part of the dataset and display it on a canvas based on its frequency, while making sure that the words don't overlap each other at any point. For plotting the Wordcloud, we used the matplotlib library.

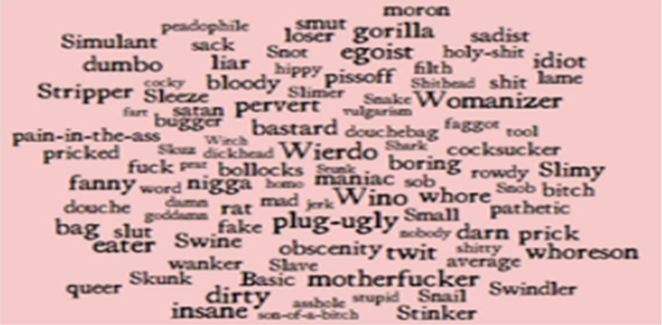


Fig. 12 Word cloud

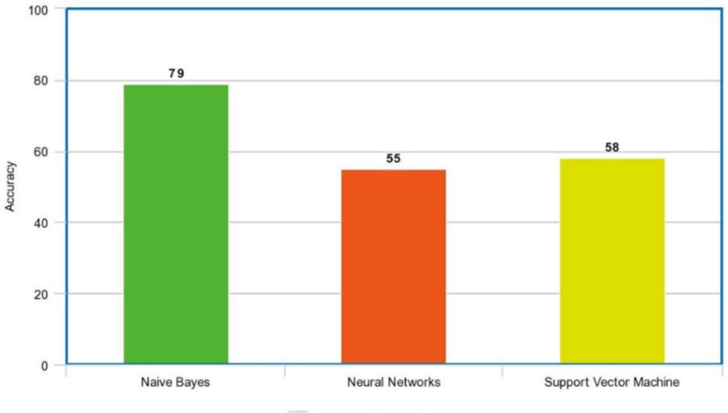


Fig. 13 Types of models

V. CONCLUSION

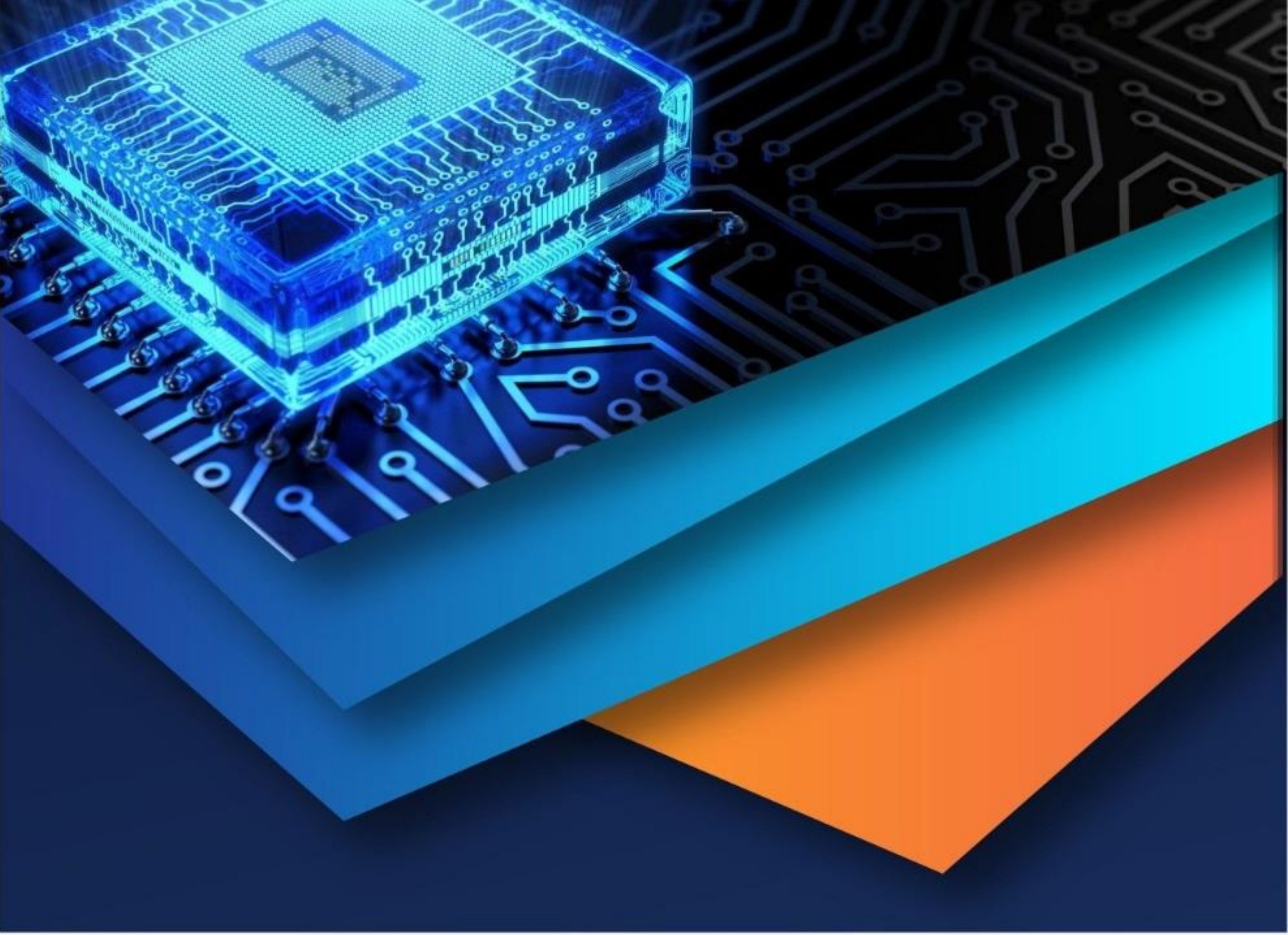
The emotional consequences that the preys of cyberbullying suffer can be disastrous and mortifying. It is even worse than face-to-face bullying, as the victim has no idea of who the bully is. On examining the computer forensic process of obtaining digital evidence from social media, and the legal aspects of such cases of cyberbullying, three models were used on the training dataset i.e. Naive Bayes, Support Vector Machine and Neural Networks. The accuracy obtained by Naïve Bayes Classification method was 79%, whereas SVM offered an accuracy of 55% and 58% of accuracy was achieved by the Artificial Neural Network Model. Thus, it is clear that the Naïve Bayes approach is the most efficient one and is therefore the best classifier for sentiment analysis, out of the three models chosen. The evidence provided by the application can be used by the users i.e., bloggers or any individual or organisation to report the crime to the cybercrime department for further legal actions.

VI. ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society’s Institute of Technology for supporting our project and extending help whenever required. We extend our immense gratitude to our mentor Assistant Professor, Mrs. Abha Tewari for her kind help and valuable guidance. We wish to express our profound thanks to all those who assisted us in the information gathering of this paper.

REFERENCES

[1] Forensic investigation of social networking applications. -Dr Mark Taylor, Dr John Haggerty, David Gresty, Peter Almond, Dr Tom Berry
[2] Using Naïve Bayes Algorithm in detection of Hate Tweets. -Kelvin Kiema Kiilu, George Okeyo, Richard Rimiru, Kennedy Ogada
[3] Mandola Monitoring and Detecting Online Hate Speech -D. Stefanidis, D. Paschalides, G. Pallis
[4] Sentiment Analysis on Twitter Data Using Support Vector Machine. -Bholane Savita D., Prof.Deepali Gore
[5] Analysis of Various Sentiment Classification Techniques -Vimalkumar B. Vaghela, Bhumika M. Jadav
[6] Sentiment Classification using Machine Learning Techniques. -Suchita V Wawre, Sachin N Deshmukh
[7] Using Machine Learning Techniques for Sentiment Analysis -Oscar Romero Llombort
[8] Social Media Sentiment Analysis using Machine Learning Classifiers -Bharat Naiknaware, Bindesh Kushwaha, Seema Kawathekar



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)