Acknowledgement

I would wish to show my special thanks of gratitude to my Chairperson Dr. Long- Zhuang Li as well as my Committee Member Dr. Dulal Kar, who gave me the golden opportunity to produce this excellent project on the topic, “Evidence-Based Sentiment Analysis of Real Time Data from Twitter”.

I also thank my family for their continuous support. Finally, I would like to thank all my friends who helped me a lot in finishing this project within the limited time.
ABSTRACT

Social media is the commonplace that allows individuals to create an account, share their views or opinions. Some of the social media are Facebook, Twitter, Instagram, etc. Twitter is one of the social media that allows the user to share the ideas within the character count of 140. Twitter works as an excellent platform for sentiment analysis. Anyone can follow anyone to receive the tweets to track the person in their timeline. As a matter of fact, the tweets of any account is visible to anyone who has a Twitter account. As a result, sentiment analysis has become challenging research area in the past decade.

In the proposed system, we use Twitter 4j interface to connect our application with Twitter. The sentiment of the tweets is analyzed based upon the Naive Bayes classifier. After analyzing the sentiment of the tweet, we connect and mine the Twitter data and store them in a MySQL database. Naïve Bayes classifier is used on mined data to get positive and negative tweets. These label tweets are taken to analyze and further processed to sentiment normalization. Finally, the result of Naïve Bayes and sentiment normalization are combined to get accurate results. This sentiment analysis is performed for secured and non-secured data that are collected from Twitter. The proposed system can be used for alerting the users in advance before any disaster or event is going to happen in a public place.
# TABLE OF CONTENTS

ABSTRACT ................................................................................................................................. iii

TABLE OF CONTENTS ........................................................................................................ iv

LIST OF FIGURES ................................................................................................................ vii

1. BACKGROUND AND RATIONALE .............................................................................. 1
   1.1 History of Twitter ........................................................................................................ 1
   1.2 Evolution of Data Mining .......................................................................................... 2
   1.3 Sentiment Analysis ................................................................................................... 3
   1.4 Sentiment Analysis Applications .............................................................................. 4
   1.5 Existing Systems ....................................................................................................... 5

2. NARRATIVE ..................................................................................................................... 6
   2.1 Problem Report ......................................................................................................... 6
   2.2 Project Objective ....................................................................................................... 6
   2.3 System Requirements ............................................................................................... 7

3. SYSTEM DESIGN .......................................................................................................... 8
   3.1 Eclipse IDE ............................................................................................................... 9
   3.2 XAMPP ..................................................................................................................... 9
   3.3 MYSQL .................................................................................................................... 9
   3.4 Creating Twitter Application .................................................................................... 10
   3.5 Data collection ......................................................................................................... 10
5.3.1 Test case 1: Twitter credentials................................................................. 26
5.3.2 Test case 2: Parsing the Dataset............................................................... 26
5.3.3 Test case 3: Pre-process tweets................................................................. 27
5.3.4 Test case 4: Naive Bayes ......................................................................... 28

6. CONCLUSION AND FUTURE WORK ............................................................. 29

BIBLIOGRAPHY AND REFERENCES .................................................................. 30
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Evolution of Data mining</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Classification Techniques</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>System Design Flow</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>Naïve Bayes Classification</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>Servlet Architecture</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>JSP Flow Diagram</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>Creating Twitter Application</td>
<td>16</td>
</tr>
<tr>
<td>8</td>
<td>Keys and Access Tokens</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>XAMPP Control Panel</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>Collecting the Tweets</td>
<td>19</td>
</tr>
<tr>
<td>11</td>
<td>Naïve Bayes Classification</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>Sentiment Normalization</td>
<td>22</td>
</tr>
<tr>
<td>13</td>
<td>Evidence-Based Approach</td>
<td>23</td>
</tr>
<tr>
<td>14</td>
<td>Graphical Representation</td>
<td>24</td>
</tr>
<tr>
<td>16</td>
<td>Parsing the Dataset</td>
<td>27</td>
</tr>
<tr>
<td>17</td>
<td>Pre-Process the Tweets</td>
<td>27</td>
</tr>
<tr>
<td>18</td>
<td>Naïve Bayes Test Case</td>
<td>28</td>
</tr>
</tbody>
</table>
1. BACKGROUND AND RATIONALE

1.1 History of Twitter

Twitter is a micro blogging platform by which users can share short messages and links to other websites, images, or videos. The word Twitter means a series of calls or sounds that are too short called tweets that make a sound produced by a bird. Twitter is a micro blogging website that is similar to Facebook, Tumbler, and Google+ which can be accessed from desktop, laptops, mobiles, etc. All the users who are registered with the Twitter can write a post with a short message limited to 140 characters or 200 bytes. In general, a message on the micro blog is written by a single user and read by one or thousands of users, also called followers.

Twitter introduced in 2006 by Jack Dorsey, Evan Williams, Biz Stone and Noah Glass, but it became famous in 2007. Initially, the number of tweets ranged from 20,000 to 60,000 a day and later the number changed to 400,000 tweets in a quarter in 2007 to 65 million tweets per day in 2010. By the end of 2010, 70,000 users are registered. Twitter has 600 million users, and the number of tweets published every day is over 300 million. [1]. People use Twitter in various purposes like for political discussions, News, Educational purpose and so forth. According to the statistics, published in March 2015, Twitter is the tenth most popular website, with 288 million active user accounts on a monthly basis. On the other side, Facebook is the first site with 1 billion monthly active users, and 23% of Internet users in USA use Twitter [13].
1.2 Evolution of Data Mining

The term data mining is introduced in 1990s. Data mining is the process of analyzing data from different perspectives and summarizing it into useful information. The extracted information used in making predictions and decision making for the benefit of organizations [2]. This information gathered can be used in applications irrespective of their field in which they found. Data mining used in many organizations such as insurance, banking, and retail, in science research areas such as astronomy and medicine and government security such as detection of criminals and terrorists.

In the 1960s, data were used to store on computers, disks and tapes where as in 1980s, the data was stored in relational databases with the help of structured query language. With the quick growth in the technology from data to databases, the methods to deliver useful information from large quantities of data are in demand [12]. There are many areas which led to the growth of data mining. Among which four are considered important, which are Artificial intelligence, Machine Learning, Statistics and data analysis as shown in Figure 1.

![Evolution of Data Mining](image)

Figure 1: Evolution of Data mining [2]
1.3 Sentiment Analysis

Sentiment analysis is the ongoing research in the text mining field. Sentiment analysis comes under the category of textual mining which means mining the text in its simplest form so that noise data can be removed. In general, there are two types of textual information is available on the web. They are facts and opinions [3]. Whatever we type in the search engines comes out as fact, which uses machine readable information such as HTML tags like title, headings and metadata. However, the same cannot apply to opinions because opinions are being taken from the web that is available in different formats like reviews, blogs, news article, discussion groups and so on. Sentiment analysis is a combination of natural language processing, computational linguistics, informational retrieval, machine learning and artificial intelligence so on. Sentiment analysis created a sensation at Text Retrieval Conference (TREC) in 2006.

Sentiment classification is defined at four levels: word level, sentence level, document level and feature level [4]. At the word level, a word can be classified into positive or negative based on the overall sentiment expressed by a writer. At the sentence level, words can be classified into positive or negative based on the overall opinion expressed by an author. In document level, a sentence can be classified into positive or negative based on overall sentiment expressed by a writer.

In feature level, an element of a sentence can be classified into positive or negative based on the overall sentiment expressed by a writer. The sentence level divides into subjectivity classification and sentiment classification. The primary aim of subjectivity classification is to find whether the sentence is subjective or objective. Most of the data found in the document and sentence level are not enough to identify each and every one in
detail about the sentiments expressed in a text as sentiments may show with different features. The first task is to identify and extract features. The second task is to determine whether the opinions on the features are positive, negative or neutral. The final task is to group the feature synonyms. The sentiment classification techniques is briefly explained in Figure 2 [13].

Figure 2: Classification Techniques [13]

1.4 Sentiment Analysis Applications

The applications of sentiment analysis are growing day to day in different fields like business analytics, marketing, customer reviews, survey responses, competitors and so on. Sentiment analysis is used more widely because of its efficiency. It uses for customer satisfaction metrics, and also we can know how much happy is the customer on that particular event or product. Sentiment analysis is also used in software companies to analyze the voice of the client, marketing, and brand reputation management and so on. Sentiment analysis is used everywhere now such as in blogs, online reviews, movie reviews.
1.5 Existing Systems

In the existing system, the tweets are collected and analyzed only for movie reviews, customer reviews on the product, trip advisor, and tourism and so on. Most of the existing systems for sentiment analysis are performed in a non-secure way for forecasting and monitoring public opinion.

Consider an example of customer reviews on hotels and tourism that is published in 2012, in which the authors used lexicon-based approach. The lexicon-based approach involves calculating the orientation of a document from a set of words or phrases. With the use of precision and recall values they achieved 80% accuracy on reviews [6], but they had tested only for hotels present in particular location for which we cannot say their approach is good [5].

Another existing method is the predicting the future of social media using twitter as a source for box-office revenues. Twitter functions as a service to publicize products and provides information to end-users. They gathered up some movie names, dates on which they are proceeding to let go, crews in the motion pictures and some other details and they performed sentiment analysis and predicted whether the film is starting to take in huge revenues or not. Nonetheless, the papers discussed on movie ratings, box-office revenues and so on [8]. However, from 2010 to 2013 there were 54 papers published on sentiment analysis on different areas [15].
2. NARRATIVE

2.1 Problem Report

Most of the existing methods are concentrating on movie reviews, customer reviews [5], election system, tourism domain, trip advisor [6] and so on. It is acceptable to use twitter as a medium for analysis. When we look at Twitter, people not only post positive and negative reviews, but also they give mixed reviews. The tweets from Twitter are very noisy, so we need to classify them accordingly to check how people are tweeting on a particular domain. To overcome all these type of problems, we are implementing an evidence-based approach, by which we can alleviate the distortion and disruption of public events using Twitter as social media and applying sentiment analysis.

2.2 Project Objective

The project aim is to perform sentiment analysis on security related tweets and normalize those tweets. Foremost, the Twitter API is processed to collect security-related keywords from Twitter and save in text file or CSV file. Second, pre-process the text document and perform Naïve Bayes classification to retrieve positive and negative tweets from the document. Third, perform sentiment normalization on those tweets to know how much positivity/negativity the tweet has and perform evidence-based approach for demonstrating the strength of the sentiment.
2.3 System Requirements

Hardware Requirements:
Processor : Pentium IV
Clock Speed : 2.5 GHZ
RAM capacity : 1 GB
Hard Disk Drive : 250 GB

Software Requirements:
Language : Core Java
Operating System : Windows 7/8
IDE : Eclipse Luna
Database : MySQL
Server : XAMPP Control Panel
3. SYSTEM DESIGN

Initially, a data is fetched from Twitter, and all the security and non-security related information is accumulated from Twitter by pre-processing of data. Pre-processing removes numbers, special characters, symbols and unwanted words from the collected tweets. Afterward, we train the dataset and perform Naive Bayes classifier to analyze the tweets for positivity or negativity, and we perform sentiment normalization to analyze if there are any mixed sentiment tweets. Finally, we perform some empirical evaluation to obtain the positive or negative tweets, and resultant tweets are saved in the database. The entire flow is shown in Figure 3.

For the database connection and server startup, we use XAMPP control panel. We check the internet connection and the database connection to link up to the application. Once we login to the Twitter page with our credentials, we should move to the Twitter API
to gather up the tokens and store them in the database for future access. The tokens can be
collected from https://apps.twitter.com/, and once we get into that, we should copy the
consumer key, consumer secret, access token, and access token secret. Twitter 4j is an
Application Program Interface (API) that is part of the Java library. The primary function
of Twitter 4j is to connect a Twitter account using these four tokens provided by Twitter.

3.1 Eclipse IDE

Eclipse is an Integrated Development Environment (IDE) for Java and other
programming languages. The Eclipse Platform provides the foundation for the IDE, which
composes of plugins for design. Eclipse Platforms and other plug-ins from the Eclipse are
released under Eclipse Public License (EPL). An eclipse perspective is the initial collection
and arrangement of views and an editor area. The default perspective is called Java. An
eclipse window can receive many perspectives, but merely one position remains active at
a time [16].

3.2 XAMPP

XAMPP is the most popular PHP development environment. XAMPP is an open source,
cross-platform web server developed by Apache consisting of Apache HTTP server,
MySQL database, PHP, and PERL. It allows the programmers to test their applications
without any access to the internet [17].

3.3 MYSQL

MySQL is an open-source relational SQL management database. It is the best
RDBMS, used for developing web-based software applications. It works well with even
larger data sets. By default, the username of MySQL database is root and doesn’t contain any password.

3.4 Creating Twitter Application

Millions of opinions can be tweeted daily on Twitter. These tweets can be picked up using Twitter API (Application Programming Interface) provided by Twitter. Twitter 4j is the integration of the project to collect data in the form of tweets. To create a Twitter application, a user has to login to his/her Twitter account by visiting the website https://apps.twitter.com. After creating the twitter application, secret tokens are brought forth. The secret tokens utilize to gather tweets from Twitter. Twitter gives back the consumer key and consumer secret tokens for applications which acts as credentials for the application and subsequently to update the settings.

The monitoring of tweets is possible through an Application Programming Interface (API) on Twitter that provides various methods for retrieval and access to user information. Twitter API divides into a Search API and a Stream API. The Search API provides access to a limited circle of recent tweets. The Stream API provides access to the flow of messages in real time [11].

3.5 Data collection

Tweets are accumulated from the account by using the hidden tokens from the Twitter application. Twitter authenticates the secret tokens and allows the users to access the Twitter website to collect the tweets. A list of keywords has to be used to gather the tweets that are stored in the database. Tweets basically are selected based on hashtag, keyword hashtag, and geo-location.
3.6 Data Pre-processing

The collected tweets are saved in a text file and pre-processing done so using a classifier. Each tweet is cleaned and pre-processed in the following ways [18]:

1. Emoticons, are recognized and replaced with the word “sad face” or “happy face”.
2. Emails are recognized and removed.
3. All the #, RT (Retweet), and EOT (End of Tweet) are removed.
4. All words beginning with “@”, “http://”, “http's://”, and “www” are removed.
5. The duplicate words are compressed, if they appear twice.

3.7 Bayes Rule

Influence of one event’s occurrence on the probability of another event is known as conditional probability. Bayes theorem allows for the calculation of conditional probability. Generally, in data mining, Bayes theorem is used to decide among alternate hypothesis [22]. Bayes theorem for conditional probability of A given B is

\[ P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)} \]

\( P(A) \) and \( P(B) \) are the probabilities of A and B without regard to each other.
\( P(A|B) \), a conditional probability, is the probability of A given that B is true.
\( P(B|A) \), is the probability of B given that A is true.

3.8 Naïve Bayes Classifier

The Naïve Bayes Classifier is a classification algorithm based on Bayes rule. If more than one attribute is present, the best way is to use Naïve Bayes classifier because all the attributes coming from a class are independent of each other as they are mined from Twitter. Naïve Bayes classifier structure is presented in Figure 4, where the network
consists of one parent and various child nodes. Every attribute from X1...Xn are independent of each other given a class label. The effect on one attribute does not influence over other attributes as all are independent. First, the prior probability is determined by checking frequency of each label in the training set to get likelihood for the label. The posterior probability is calculated for all classes, and the class with highest probability will be the instance label. Naive Bayes classifier is very effective since it is less computationally intensive (in both CPU and storage) and it takes a modest amount of training information [9].

![Figure 4: Naïve Bayes Classification](image)

### 3.9 Natural Language Processing

Natural language processing is a major area of research in computer science that deals with scientific study of human language, and it is a subfield of Artificial Intelligence. It is not concerned with the topic of a document, but with the opinions expressed to determine the attitude of an author to some topic to determine the polarity. Two classes form the backbone of the core NLP package: Annotator and Annotation. Annotations are the data structure that holds the result of Annotators. Stanford Core NLP inherits the AnnotationPipeline class and is customizable with NLP Annotators [10].
3.10 Java

Java is an object-oriented programming language originally developed by Sun Microsystems. Java runs on a variety of platforms such as Windows, Mac OS, and the version of UNIX and LINUX. In our project, we use Java as a programming language in which servlets and JSP are used for web development [21].

3.10.1 Servlets

Java Servlets are programs that run on web servers and acts as a middle layer between a request coming from an HTTP client and HTTP server. Using a servlet, we can collect input from users through the web page. Servlet architecture is shown in the Figure 5.

![Servlet Architecture](Google images)

3.10.2 JSP (Java Server Pages)

JSP is a technology for developing web pages that support dynamic content that helps developers to insert the Java code in HTML pages by making use of JSP tags `<% %>`. JSP pages are always compiled before they are processed by the server which is shown in Figure 6.
Figure 6: JSP Flow Diagram [Google images]
4. IMPLEMENTATION OF SYSTEM DESIGN

Environment

The project is implemented in an Eclipse IDE environment as it is more flexible for programming. We use the XAMPP control panel that consists of Apache for web services, and MYSQL for the database, to store all the retrieved Twitter data and other information.

Twitter 4j

There are libraries available in many programming languages to access to the Twitter API. Among those, Twitter 4j is an unofficial Java library for the Twitter API developed by Yusuke Yamamoto. Twitter 4j library is used to collect tweets, and a search script is written in Java to make queries [15].

The implementation of the system design is done in following steps:

1. First create an application with the twitter.
2. Collect the tweets on a particular domain by using Stream Listener class.
3. Pre-process the collect tweets.
5. Perform Sentiment Normalization.
6. Perform Evidence-based approach.
7. Result is displayed in graphical chart.
4.1 Create Twitter Application

To create an application, the developer has to login to the twitter to access the Java project. Configuration Builder class is one of the classes available in Twitter 4j jar collection, which handles collecting the data from Twitter by creating an interface between the application and Twitter which is shown in Figure 7.

![Create an application](image)

Figure 7: Creating Twitter Application

For authentication process, access tokens are used to create an application for the project. Parameters for an authentication required to configure Twitter 4j are. These tokens are shown in Figure 8.

- Consumer Key
- Consumer Secret
- Access Token
- Access Token Secret.

```java
void initConfiguration (){
    cb = new ConfigurationBuilder();
    cb.setDebugEnabled(true);
    cb.setOAuthConsumerKey("Q2zBKAOptmFcGLvK9G0Rn1imf");
    cb.setOAuthConsumerSecret("7ctcs8H3zTX0oNAAn12AI9VTpbSFcyUAjShjUKeXiI36HkQA7q");
    cb.setOAuthAccessToken("3235478995Llck1SqRLmKDAPUV6t6iRv8ibcrP0KljreWM9NE");
}
```
cb.setOAuthAccessTokenSecret("MQdMrTDyAcJhVvB06BEvBPlauX1PEle9C56yu7DCwsuFn");
    twitterStream = new TwitterStreamFactory(cb.build()).getInstance();
}

Figure 8: Keys and Access Tokens

XAMPP control panel includes Apache Server, MYSQL database, PHP, and PERL. However, we use only a server and a database for our project that need to be started before running the application. The Admin button of MySQL in the control panel can be clicked to redirect to phpMyAdmin page, is shown in Figure 9, where we can store all the database related files and keywords.
4.2 Collecting Tweets

To gather the tweets, first, we need to create a dataset. Initially, we need to connect through Twitter 4j API library with our consumer key, consumer secret, access token, access secret token details which can be collected from https://apps.twitter.com/. Once all the details are collected, we use security related keywords of a particular domain to retrieve tweets from the Twitter API library. To retrieve those keywords in the tweets, we use Filter Query class to track the keywords. In order to get the tweets, we use the en.split() method so that tweets in all other languages will be discarded. For the training, we collect 300 tweets from Twitter through Stream Listener Class which is shown in Figure 10. The collected tweets contain tweet id, username, content, and end of the tweet (EOT) specifications that can be pre-processed.

```java
FilterQuery fq = new FilterQuery();

fq.track(keyWords);
    fq.language("en".split(" "));
    twitterStream.addListener(listener);
    twitterStream.filter(fq);
```
4.3 Perform Naïve Bayes Classification

The Naïve Bayes Classifier is a classification algorithm based on Bayes rule. If more than one attribute is present, the best way is to use Naïve Bayes classifier because all the attributes coming from a class are independent of each other as they are mined from Twitter. After pre-processing the collected tweets, we classify the tweets into three sets which are classified into positive, negative, and neutral. For our project, we collect some tweets manually label each tweet, and assign a value for each tweet. Each tweet is checked for a keyword in the database and accordingly it is stored in the respective set which is in shown in Figure 11.

```java
public void calculateNaiveBayes(Map<String, Integer> tweetList){
    try{
        List<Integer> tweetValues = new ArrayList<>();
        tweetValues.add(0);
        tweetValues.add(2);
        tweetValues.add(4);
    }
```
for (Map.Entry<String, Integer> tweet : tweetList.entrySet()){
    String tweetStr = tweet.getKey();
    tweetStr = checkNegationSet(tweetStr);
    tweetStr = checkIntensifier(tweetStr);
    if (tweet.getValue() == 4)
        positiveTweetList.add(tweetStr);
    else if (tweet.getValue() == 0)
        negativeTweetList.add(tweetStr);
    else
        neutralTweetList.add(tweetStr);
}
}

Figure 11: Naive Bayes Classification
4.4 Sentiment Normalization

In our project, we implement sentiment normalization based on a single word within a text message by using a sentiment combining function. It provides an absolute sentiment of the message as a normalized value from the range of -100 to 100. The primary goal of the sentiment normalization is to find a function that will appropriately model a relationship between two different sentences. It should not only be able to establish if a sentence is positive or negative, but also, for any two positive/negative sentences it should be able to determine which one is more positive/negative than the other. A possible approach is to normalize the sentence by calculating the average of the sentiment values assigned to words from the sentence, for instance, to differentiate between the sentences that contain five words with a sentiment value of 100 and a sentence with only one word with a sentiment value of 100. However, we cannot identify that the previous sentence is more positive than later. The normalization formula should combine both the averaged sentiment of a sentence and the number of words applied. The following formula is used for calculating the total positive and the total negative sentiments.

\[
F_P = \min \left\{ \frac{A_P}{2 - \log(p \times W_P)}, 100 \right\}
\]

\[
F_N = \max \left\{ \frac{A_N}{2 - \log(p \times W_N)}, -100 \right\}
\]

Where \( A_P \), \( A_N \) stand for positive and negative sentiments. \( W_P \), \( W_N \) are positive and negative words respectively. Instead of increasing or decreasing the value of a word’s sentiment we include the number of intensifiers in the normalization formula [23].
int evaluateFp(int aP, int wP, int iP) {
    double fP = 0, p = 4.5;
    fP = 2 - (Math.log((p*(wP + iP))));
    fP = aP/fP;
    fP = findMin(fP, 100);
    return (int)Math.abs(fP);
}

int evaluateFn(int aN, int wN, int iN) {
    double fN = 0, p=4.5;
    fN = 2 - (Math.log(p*(wN + iN)));
    fN = aN / fN;
    fN = findMax(fN, -100);
    return (int)fN;
}

Figure 12: Sentiment Normalization
4.5 Perform Evidence-based Approach

We combine a sentiment combination and normalization process to obtain two values as output. The range from 0 to 100 represents a positive tweet and -100 to 0 represents a negative tweet. Initially, this algorithm compares with the absolute values of two sentiments and classify the tweets as positive or negative depending on which of the value is greater. To consider an example, the words “like”, “lazy”, “love” etc. either be positive or negative so they can be concluded as neutral words. Positive and negative evidences are combined separately and the maximum value determines the classification result.

```java
public class EvidenceBasedApproach {
    void saveTweet(String tweet, String type) {
        try {
            String sql = "insert into evidence_based (tweet, category) values('" + tweet + "," + type + ")";
            DBConnection.executeQuery(sql);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Figure 13: Evidence-Based Approach
4.6 Graphical chart

After doing all the process, the tweets are displayed in the doughnut chart representing how many positive tweets and negative tweets are present.

Figure 14: Graphical Representation
5. TESTING AND EVALUATION

Essentially, there are two types of testing. They are manual testing and automation testing. In manual testing, the tester takes the role of the end-user and tests the software to identify the bug or any unexpected behavior. Unit testing, integration testing, system testing, and user acceptance testing comes under the category of manual testing. In automation testing, the tester writes the script and uses some tools to test the product. By using automation testing, we can increase the test coverage, improves accuracy, and saves time and money when compared to manual testing. Some software testing tools include Selenium, Silk Test, Test Complete, Win Runner, and IBM Rational Functional Tester.

5.1 Unit Testing

Unit testing is also referred to as Module testing. This task is generally done by the programmer, not by the tester as it requires in depth knowledge of the internal program design and code. Each module of the application is tested on an individual basis to examine for the required yield.

5.2 Integration Testing

Individual software modules are combined and tested as a group. Integration testing can be performed in two ways: Bottom-up and Top-down.

5.2.1 Bottom-up integration Testing

In this approach, testing is conducted from sub module to the main module. The advantage of bottom-up approach is that test conditions are easier to create, observations of test answers are more leisurely.
5.2.2 Top-down integration Testing

In this approach, testing is conducted from main module to sub module. The advantage is once the I/O functions are added, the representation of test cases is easier.

5.3 Test Cases

5.3.1 Test case 1: Twitter credentials

We have to check the correct credentials before login to twitter. If we go with the wrong username and password it won’t allow to share or write the tweets with other users, but it sets aside to read the tweets which is shown in Figure 15.

![Figure 15: Twitter Credential Test Case](image)

5.3.2 Test case 2: Parsing the Dataset

To parse the dataset, first we need to collect the tweets related to security and store them in one text file or CSV file. Initially, the dataset is chosen by the user and tweets in the file is used by the system. The system will not continue if the CSV file or a text file is not in the correct format which is shown in Figure 16.
5.3.3 Test case 3: Pre-process tweets

After collecting the tweets from the dataset file, tweets need to be pre-processed. In the pre-process step, all the stop words, URL’s, Email Id, and so on are removed. The screenshot for the output of pre-processed tweets is given below in Figure 17.
5.3.4 Test case 4: Naive Bayes

The collected tweets from the dataset file are processed tweet by a tweet in Naive Bayes classifier step. In this step, the tweet is split into positive tweets, negative tweets, and the neutral tweets. If the classifier proceeds without any error, the list of classified tweets is displayed to the user in Figure 18.

![Figure 18: Naïve Bayes Test Case](image)
6. CONCLUSION AND FUTURE WORK

The primary aim of the project is to determine the intensity of the sentiment rather than finding positive or negative. This is achieved by performing evidence-based approach algorithm along with sentiment normalization. We use Twitter 4j interface to connect our application with Twitter. The sentiment of the tweets is analyzed based upon the Naive Bayes classifier algorithm. After analyzing the sentiment of the tweet, we normalize the sentiment and perform an empirical evaluation to check whether the sentence is positive or negative and then display the list of resultant tweets in the database.

We limit this project for small training set of 300 tweets, and in future we can increase the tweets as many as we want and perform automatic detection of positive and negative sentiments instead of manual label of tweets.

As of now, the present implementation is limited to Twitter applications, but in future one can implement for other social networks like Facebook and other sites to retrieve data, and classify them, to alert the user in advance of any anti-social events.
BIBLIOGRAPHY AND REFERENCES


