A Framework for Analyzing Real-Time Tweets to Detect Terrorist Activities

Mohammad Fahim Abrar, Mohammad Shamsul Arefin, Md. Sabir Hossain

Department of Computer Science and Engineering
Chittagong University of Engineering and Technology (CUET)
Chittagong-4349, Bangladesh
Email: fahimabrar02@gmail.com, sarefin@cuet.ac.bd, sabir.cse@cuet.ac.bd

Abstract*—Terrorist organizations use different social media as a tool for spreading their views and influence general people to join their terrorist activities. Twitter is the most common and easy way to reach mass people within a small amount of time. In this paper, we have focused on the development of a system that can automatically detect terrorism-supporting tweets by real-time analyzation. In this system, we have developed a frontend for real-time viewing of the tweets that are detected using this system. We have also compared the performance of two different machine learning classifiers, Support Vector Machine (SVM) and Multinomial Logistic Regression and found the first one works better. As our system is highly dependent on data, for more accuracy we added a re-train module. By using this module wrongly classified tweets can be added to the training dataset and train the whole system again for better performance. This system will help to ban the terrorist accounts from twitter so that they can't promote their views or spread fear among general people.

Keywords—Social Media, Twitter, Terrorism, Real-Time Tweets, Machine Learning

I. Introduction

Internet technology has a lot of benefits. We can share information and ideas quickly and it is effective across the border. It is also recognized as a fundamental human right. The Internet has also proven to be highly dynamic means of communication. One of the biggest technology of the Internet is social media. Social media comes in many forms including blogs, photo sharing platforms, forums, business networks, social gaming, chat apps, social networks. Users share a lot of information daily in these social media. And in this era of data science, this data can be used to obtain different insights to improve the user experience of using the Internet.

Microblogging is a content-oriented concept in which people can interact with others both known and unknown. Twitter, which is a successful microblogging social network, has gained enormous popularity in recent year. Twitter users are restricted to writing messages of no more than 140 characters these are then turned into short messages. This short message is known as 'Tweet'. Twitter has 328 millions of daily users and 500 millions of tweets are sent per day.

Though the internet is a blessing, it has also dark sides. Terrorist organizations have highly benefited by the worldwide reach, speed, and growth of the internet. By using social media platform, mostly Twitter, terrorist organizations spread their views. They can now easily reach anywhere in the world. As the internet is now available in every country, a terrorist doesn't have to go to the battlefield to fight for their views. They can just sit in front of the computer and promote their views which can be useful for their organization to spread their network in every part of the world. Terrorist organizations which are active on social media to promote their views and spreading propaganda are often leaderless. That makes tackling terrorist propaganda a difficult task. They used social media to recruit new members from all over the world. They also spread fear among the people of different countries by using Twitter and other social media platform. Among all social media platforms, Twitter is more public. When a tweet is posted on Twitter it can reach more people than other social media platforms. So, Twitter is the first choice of terrorist organizations to spread their views and propaganda. In the last two years, Twitter has suspended around million twitter account for spreading terrorism. Their approach of suspending an account is if someone reports any account, then they suspend the account if the account really promotes terrorism. It is a very time-consuming process. The motivation behind this project is to crawl tweets from the twitter in real-time and then analyze the tweets to determine the support of terrorism. This system will significantly improve the time to review the tweets supporting terrorism.

The remainder of this paper is organized as follows. Section II provides a brief review of related work. In section III, we discuss in detail the framework of our proposed approach. Section IV presents the implementation details. Section V contains experimental results and analysis. Finally, we conclude and sketch future research directions in Sec. VI.

II. Related Work

M. Ashcroft et al. [1] made an attempt to detect jihadist messages from Twitter. They used sentiment analysis to detect if a message supports ISIS or not. They used some keyword to extract tweets from the Twitter feed. The advantage of this work is it uses three different features such as Time based features, Sentiment based features and Stylometric features to detect jihadist text. They got almost 90% accuracy using these features. One of the limitations of this study is they didn't use any real-time validation of their classification algorithm. Also, they didn't build any tools to detect jihadist text automatically. Walid M. et al. [2] studied to predict future

978-1-5386-9111-3/19/$31.00 ©2019 IEEE

support or opposition for ISIS from tweets. In this study, the authors used Twitter data to study the ISIS support of users. They used the bag of words model as feature vector which included individual terms, user mentions and hashtags. They used SVM with a linear kernel to train a classifier to predict the support or opposition of ISIS. They obtained about 87% accuracy using the SVM classifier model. One limitation of their approach is, they didn't consider real-time validation of their method.

S. Azrina et al. [3] studied to detect terrorism from text using sentiment analysis. The advantage of their study is they did a comparative analysis on several techniques to detect terrorism from the text. They did a comparative analysis of Neural Network, Support Vector Machine, Sentiment Analysis with Naive Bayes and Lexicon based approach. Then they finally adopted and improved Naive Bayes method for their research. One of the limitations of this study is it uses user behavioral analysis to improve the accuracy of Naive Bayes algorithm as it shows medium level accuracy compared to other algorithms. But it is not always possible to analyze a user's behavior if he doesn't have that much of tweet history on Terrorism.

Lisa K. [4] studied to classify tweeps and tweets as being multipliers of jihadism. They used Machine Learning to build a classifier that can analyze a tweet to find multipliers of jihadism. They used AdaBoost classifier to train a model. They analyzed both Arabic and English tweets. They obtained about 84% accuracy for Arabic tweets and 98% accuracy for English tweets. They didn't test their model in a real-time environment.

Pooja W. et al. [5] studied to classify radical tweets in the categories such as Media, War terrorism, Extremism, Operations, Jihad, Country and Al-Qaeda. They built a dictionary to classify the tweets into different categories. They built the dictionary by looking at tweets containing hashtags like Al-Qaeda, Jihad, Terrorism, and Extremism and by collecting relevant words for their purpose. They built a process based on the presence of the word in the dictionary. Their process obtained about 90% accuracy. Their study is proof that keywords can be used successfully in classifying tweets.

V. Wisdom et al. [6] studied well on Twitter data analysis. They used python for their analysis of Twitter data. They used a python library called 'Tweepy' for analyzing the tweets. The advantage of this study is they described numerous analysis on tweets such as term frequencies, bigram terms, most used hashtag, most used mentions. The limitation of this study is it only deals with the basic analysis of Twitter data. No advance technique of analysis such as sentiment analysis, natural language processing, machine learning is introduced here in this study.

Kathy L. et al. [7] classified twitter trending topic into different categories as the list of trending topics provided by Twitter is often hard to understand what these trending topics are about. They classified the Twitter trending topics into 18 different categories like politics, technology, sports etc. They used a bag of words approach using TF-IDF (Term Frequency-Inverse Document Frequency) which provides how important a word is in a document. The best accuracy is obtained from using Naive Bayes Multinomial classifier (65.36%). It performed better than Naive Bayes (45.31%) and SVM (61.76%).

M. Trupthi et al. [8] used Natural Language Processing (NLP) and Machine Learning Technique for sentiment analysis. The advantage of their procedure is they analyzed real-time tweets using Twitter Stream API. The limitation of this study is, they used the word of sentence individually rather than analyzing the sentence individually. So, the semantic meaning was neglected which is present between words. They also failed to analyze if a user's tweet contain sarcasm or they really mean it.

Lee S. et al. [9] compared four text mining methods: Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Allocation (LDA), and Correlated Topic Model (CTM) using topic model and spam filtering. They concluded that PLSA shows the highest performance and next to LDA, CTM, and LSA in order. One of the limitations of this study is they only considered statistical approach and didn't extend their study to syntactical and morphological approach.

A work on topic discovery based on text mining techniques was presented by Pons P. et al. [10]. They proposed a hierarchical clustering algorithm that combines partitioned and agglomerative approaches to produce topic hierarchies. They considered document place, time reference, and textual contents. This resulted in less time complexity while detecting a new topic. The accuracy of their proposed method is not satisfactory.

III. System Architecture and Design

The system architecture of Twitter Terrorism Detection Framework comprises five basic modules: Twitter data crawler module, the storage module, tweet classification module, the output module, and training module. The function of twitter data crawler module is to crawl real-time tweets from Twitter using Twitter Streaming API. The storage module stores the tweets temporarily. Tweet classification module predicts the category of the tweet. The output module shows the output of the system. The training module builds the classification model which is used to predict the category of each tweet. The architecture of the framework is shown briefly in Fig. 1.

The first module is the Twitter Data Crawler Module. This module is used for fetching the real-time tweets from Twitter. First, we need 4 API keys from the Twitter developer console. These 4 keys are needed to get access to Twitter API. Using these 4 keys we set up a twitter real-time tweet listener. This listener allows us to collect real-time tweets from Twitter.

The second module is the Storage Module. In this module, we have stored the tweet that is coming from the Twitter Streaming API. The Tweets are stored here temporarily. In the storage module, there is a facility to store the last 1 million tweets for further analysis. Before storing the tweets into the database we first removed the retweets from the tweet collections. Because if we keep retweets the database will be heavy as the original tweet is already stored and the retweet would only be a duplicate. Because of this, we have removed all the retweets from the database by checking the retweet symbol (RT).

In the classification module we have retrieved the real-time tweets from the database and then used our classifier model to predict the category of each tweet.

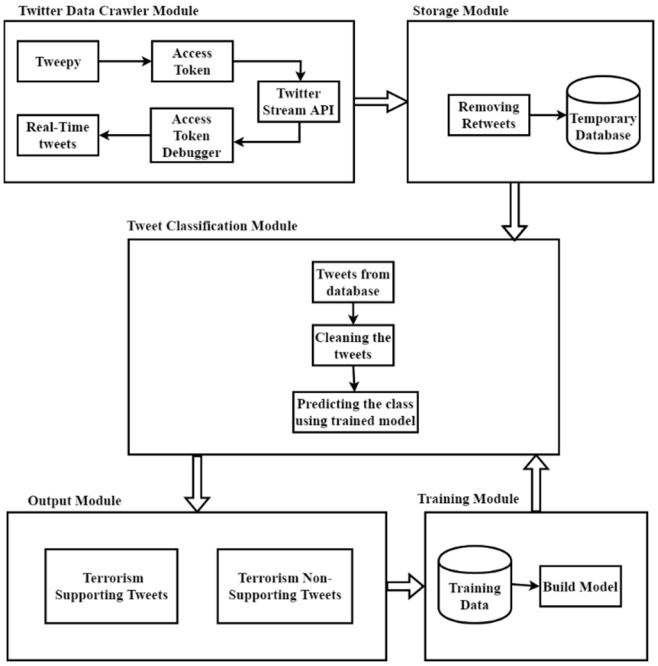

Fig. 1. The architecture of the proposed framework

According to the prediction the tweet is stored in a different table of the database. If the tweet is terrorism-supporting then the tweet is stored in a table named *matched_tweets*. Otherwise, the tweet is stored in a table called *all_tweets*.

The output module is a user interface, which is a web application. It shows the categorized output of the real-time tweets which is dynamically updated. Also, it shows the tweets that are marked as terrorism-supporting tweets.

The Training Module is used to train the classification model which is generated by using the training dataset and learning algorithms.

We divided the work of our framework into 3 different sub-tasks.

A. *Crawling Data from Twitter*

We have crawled real-time tweets from the twitter by using twitter streaming API. To do this we have provided four keys that we have collected from the Twitter developer website. The 4 keys are:

1. Access Token
2. Access Secret Key
3. Consumer Token
4. Consumer Secret Key

These 4 key are needed to get access to Twitter API. Using these 4 key we can set up a twitter real-time tweet listener. This listener will allow us to collect real-time tweets from Twitter. As we have used python to build our framework, we have used a python library that helped us getting access to the Twitter streaming API. We have used a python library *'Tweepy'*[11] to access Twitter streaming API. Algorithm 1 illustrates the crawling real-time tweets from Twitter.

While collecting real-time tweets we have made sure to handle all kinds of error that would break the connection. If the connection breaks, the crawler module will stop working. That's why we have checked for several exceptions such as database exception, limit exceeded exception etc.

Algorithm 1: Crawl Real-Time Tweets
Input: Developer Access keys
Require: Real-Time tweets streaming from twitter
1. **Begin**
2. **Call** Twitter API
3. **Call** Twitter Streaming API
4. **Set** accessTkn = ""
5. **Set** accessTknSec = ""
6. **Set** consumerKey = ""
7. **Set** consumerSec = ""
8. tweets = STREAM-LISTENER(accessKeys)
9. **Create** a table named all_tweets having the field username,
tweet, tweet_id, type
10. **if** tweets != null **then**
11. **if** tweets['retweet'] = False **then**
12. **Insert** tweets['text'] , tweets['username'] and tweets['tweet_id'] into the database
13. **End**

B. *Pre-Processing Crawled Data*

After data is crawled from the twitter the tweets are in raw form using algorithm 2. We can't use these tweets to classify or train. So, we have cleaned the tweet before using them in classification or training. To handle the special component of a tweet, we have done the following pre-processing tasks:

1. URL is removed
2. Any user mention is removed
3. Hash (#) from the hashtag is removed
4. Contracted words are converted to their long form.
5. Tokenized the tweet

Algorithm 2: Cleaning raw tweets
Input: raw tweets
Require: clean the raw tweets
1. **Begin**
2. remove url from raw tweets
3. remove hash (#) symbol of hashtags from raw tweets
4. remove user mentions form raw tweets
5. remove retweet symbol RT from raw tweets
6. convert the raw tweets into lowercase form
7. search for contracted from in tweets
8. **if** contracted form found **then**
9. replace it with long form
10. search for stop words in tweets
11. **if** stop words found **then**
12. remove the stop words
13. tokenize the tweets
14. apply stemming on the tweets
15. **End**

C. *Building Model and Generating Output*

To predict the class of the tweet we needed a mathematical model which can classify the tweets based on their features. We have used two classification algorithm. These are SVM (Support Vector Machine) and Logistic Regression. Using our training dataset we built a model that can classify the tweets accurately. By using the model that we have

built in the previous steps, we can classify a tweet. The classification result is 0 or 1 or 2. According to this result, we can show the type of tweets. Algorithm 3 is used to classify real-time tweets.

Algorithm 3: Classification of real-time tweets
Inputs: model file
Require: Classification of the tweets
1. **Begin**
2. *classifier* = load(*model*)
3. **for** each tweets in the all_tweets table in the database **do**
4. clean the tweets
5. *type* = *classifier*.predict(clean_tweets)
6. **if** *type* = 0 **then**
7. *result* = "Terrorism Supporting"
8. **else if***type* = 1 **then**
9. *result* = "Terrorism Non-Supporting"
10. **else if** *type* = 2 **then**
11. *result* = "Random"
12. **show the *result***
11. **End**

IV. Implementation Details

In this section, we have provided the implementation of our proposed system in details.

In order to start crawling from Twitter, we had to go to the twitter developer portal to get access tokens and consumer secrets. An Application Programming Interface (API) is a standardized system of programming instructions that allows web platforms to access and share information from one another. Like many other web tools, almost every social network has released its API for researchers and other web developers to use. We have used Twitter Streaming API to crawl data from Twitter. API (Application Platform Interface) will allow us to authenticate access token provided by the authority of the social network. If the API keys are valid, the crawler program can access the Twitter API and crawl tweets from the twitter. If the API keys are not valid then the API will send an API access error.

To get the API access keys first we have opened an account on the Twitter developer website. In the website, we have created an app named "Twitter Crawler Framework". This provided us with the access tokens.

Fig. 2 shows a snapshot of the app that we have created for use in the Twitter Terrorism Detection Framework.

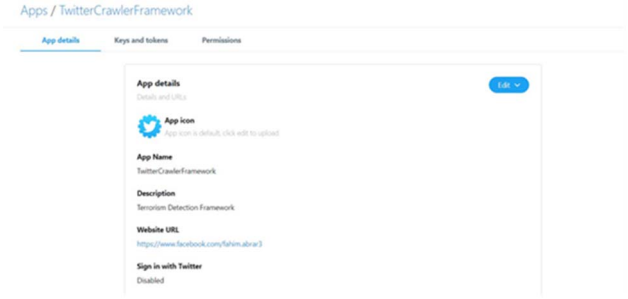

Fig. 2. Twitter Crawler Framework app in twitter developer website

Fig. 3 shows the screen where live tweets are shown. Each tweet which is random or terrorism non-supporting are shown here with respective level. Random tweets are shown using the green color box and terrorism non-supporting tweets are shown using the yellow color box. Random tweets are shown using green blocks and non-terrorism-supporting tweets are shown using a yellow block. In the top portion, there is a count which shows the number of tweets that are detected as terrorism-supporting tweets till now. On the left side of each tweet, the username of the user will be shown. It is clickable and if it is clicked it will load the profile of that specific user in a new tab on the browser.

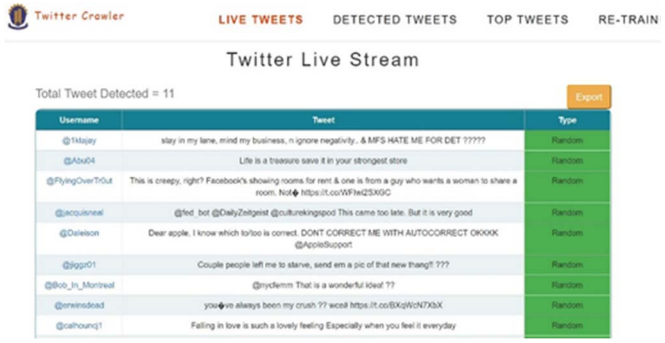

Fig. 3. Live Tweets

Fig. 4 shows the detected tweets screen. In this screen, the tweets that are detected as terrorism-supporting tweets are shown. Besides every tweet, there are two buttons. One is called RAND and another is called TRNS. RAND button should be clicked if the tweet is random tweets but falsely detected as terrorism-supporting tweets. TRNS button should be clicked if the tweet is not terrorism-supporting but falsely detected as terrorism-supporting tweets. Those tweets will be stored in the database with the new label and later can be used to retrain our model.

Fig. 5 shows the top 10 tweets that support terrorism. We have ranked the tweets according to their predicted probability values.

Fig. 4. Detected Tweets screen

Fig. 5. Top Tweets screen

Fig. 6 shows the tweets that are stored for retraining purposes. This list shows the tweet for retraining and the type of the tweets as random or terrorism non- supporting.

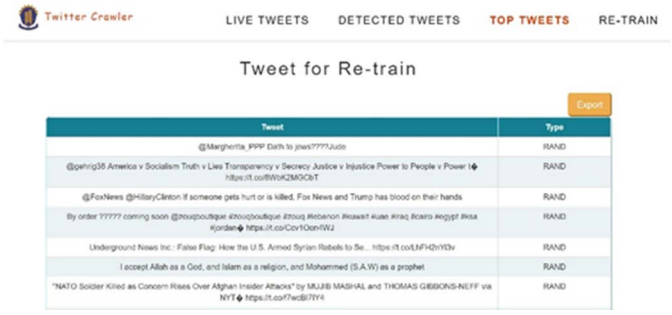

Fig. 6. Re-train screen

V. Experimental Results and Analysis

A. Experimental Setup

The developed Twitter Terrorism Detection System has been implemented on a machine having the windows 10, 3.40 Core i5-8250U processor with 8GB RAM. The detection system has been developed with Python and the user interface is developed with Bootstrap CSS, HTML, JavaScript, jQuery and AJAX. We have also used MySQL database to store our data and used PHP to populate the user interface with data from the database. For coding, we have used Pycharm for python and PhpStorm for coding the user interface.

B. Performance Evaluation

a) Crawling Mechanism of Dataset

We crawled tweets from Twitter to make our dataset. We have collected tweets to make our dataset in three different ways. These are:

1. Collected random real-time tweets
2. Collected tweets by using some keywords related to terrorism
3. Collected tweets from some known terrorist twitter account

After collecting all the tweets we manually analyzed the tweets and divided them into three different classes and finally put them in a CSV file. In the CSV file, there are two columns one is tweets and another type. Tweets column contains the tweets and the type column contains the type of that tweet in numeric form. We have used the numeric form to represent the type of tweet because the classifier algorithm needs the type to be in the numeric form so that it can predict the outcome of a tweet.

TABLE I Summary of collection method

Collection Type	No of tweets
Randomly from Real-Time Tweet Stream	21248
Using Terrorism-Related Hashtags	15770
Crawling Terrorist Profiles	18105
Total =	55123

In Table I, the number of tweets collected using different collection methods are shown. These tweets are further analyzed and categorized into different classes to make our dataset.

In Table II a number of tweets of different types in our training dataset are given. Using this dataset we have trained our classifier to build a model.

TABLE II Summary of the Training dataset

Type of Tweets	Numerical Value in Dataset	No of tweets
Terrorism Supporting	0	13369
Terrorism Non-Supporting	1	16506
Random	2	38617
	Total =	55123

b) Selection of N-Gram

In this study, we have considered using 3 types of N-gram. These are

1. Unigram (N=1)
2. Bigram (N=2)
3. Trigram (N=3)

For both of our classification algorithm, we have used this 3 types of N-gram with varying size of the feature vector to select the best N-gram and size of the feature vector for which we get the best result.

In Fig. 7 we can see the graph of accuracy vs feature vector size graph for logistic regression. We can see that from the graph that for bigram with feature vector size 10000 we get the best accuracy.

In Fig. 8 we can see the graph of accuracy vs feature vector size graph for SVM. We can see that from the graph that for bigram with feature vector size 10000 we get the best accuracy.

So, in both case, bigram and feature vector size of 10000 gives the best result. So we have chosen bigram and feature vector of size 10000 in our experiment.

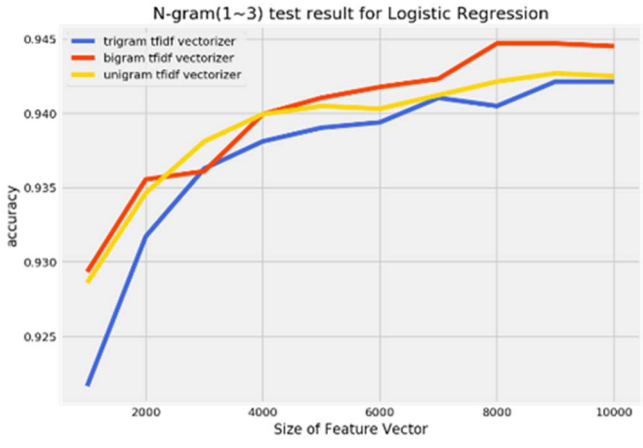

Fig. 7. Accuracy vs feature vector size for Logistic Regression

a) Performance of Test Data

We have used 20% of our total dataset for testing purpose to test the performance of our classifier algorithms. So a total of

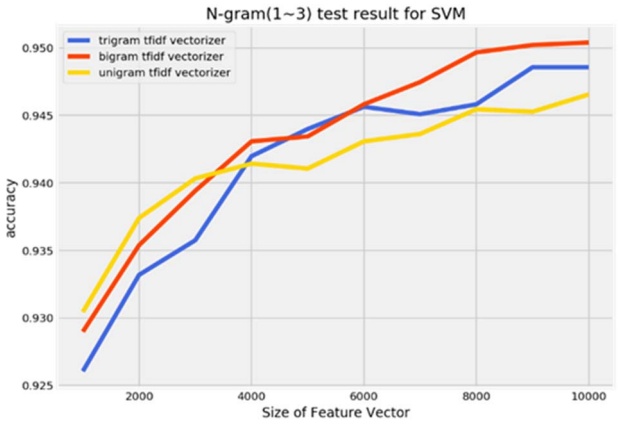

Fig.8. Accuracy vs feature vector size for SVM

16536 tweets have been used to test our classifier models. The test and train set are divided by using a random selection of tweets from the dataset. We can show the performance of our system by using the confusion matrix.

TABLE III Confusion Matrix

	TRS	TRNS	Accuracy	Precision	Recall
SVM	5152	87	95%	87%	98%
	732	10565			
Logistic Regression	4677	73	94%	83%	98%
	896	10890			

The confusion matrix for both SVM and logistic regression classifier are given in Table III from the table we can say that SVM performs slightly better than Logistic Regression. So we have chosen the SVM model to classify our real-time tweet data.

b) Performance of the Whole Framework

After checking the accuracy of our model in the test environment, we have checked our framework in a real-time tweet environment and analyzed its performance. We have executed our framework in 5 different sessions. In each session, we have crawled approximately 40000 tweets. In a total of 20000 tweets. The time frame of the collected tweets is given in Table IV.

TABLE IV Summary of real-time collection of tweets

Date	Day	Time Frame	No of Tweets Analysed	Total No of Tweets Analysed
16/10/2018	Tuesday	10:43 to 13:25	40,000	40,000
17/10/2018	Wednesday	15:30 to 18:15	40,000	80,000
18/10/2018	Thursday	09:00 to 11:50	40,000	1,20,000
19/10/2018	Friday	8:00 to 10:45	40,000	1,60,000
20/10/2018	Saturday	14:25 to 18:10	40,000	2,00,000

After analyzing the tweets we have identified how many tweets of the different category are detected in different time-frame. We have shown the result using the bar graph in Fig. 9 From the figure, we can see that on Friday (1,60,000 total tweets) the number of terrorism-supporting tweets is much higher than any other days.

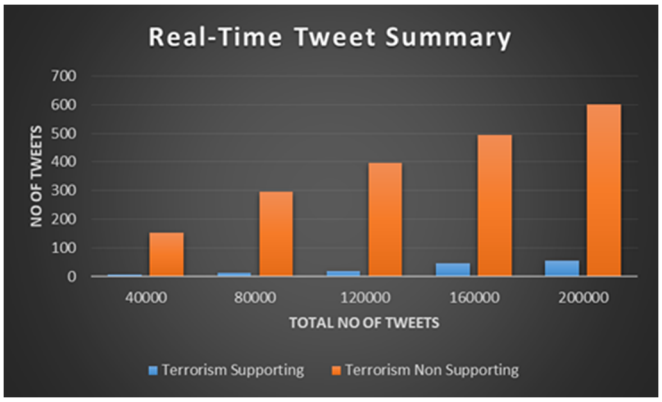

Fig. 9. Real-time tweet summary

The accuracy of our framework in a real-time environment is given in Table V:

TABLE V Comparison of actual and framework data

	Tweet Type	Framework	Actual
Support Vector Machine	**Terrorism Supporting**	55	40
	Terrorism Non-Supporting	601	433

From Table V, we can calculate that the accuracy of our framework to detect terrorism in real-time is 72% using confusion matrix.

VI. Conclusion

In this Paper, We have discussed Twitter Terrorism Detection framework to detect tweets that support terrorism from real-time tweets stream. Our framework collects real-time tweets by using twitter streaming API and analyses every tweet. It can categorize the tweet into three different classes and based on the category of the tweet, it is stored and shown in the different screen of our web application. We have also created a re-train module which will be used to retrain our model so that it can perform more accurately. Our framework has a user-friendly interface. The overall accuracy of our proposed system is 95% and 94% for SMV and Logistic Regression respectively.

In our study, the framework is limited to collect and analyze tweets that are written in English. So, further research can be done to extend this study to support other languages. The accuracy of the proposed system can be enhanced by analyzing shared images or videos on Twitter by users. Our framework will falsely detect sarcastic text as a terrorism-supporting tweet. So, further research can include detecting sarcastic tweets which actually doesn't support terrorism.

References

[1] M. Ashcroft, A. Fisher, L. Kaati, E. Omer, and N. Prucha, "Detecting jihadist messages on twitter," in *Proceedings of the Intelligence and Security Informatics Conference (EISIC)*, 2015 European, Sept 2015, pp. 161–164.

[2] M. Walid, D. Kareem and W. Ingmar, " #FailedRevolutions: Using Twitter to Study the Antecedents of ISIS Support," in *First Monday*, 2015.

[3] S. A. Azizan and I.A. Aziz, "Terrorism Detection Based on Sentiment Analysis Using Machine Learning," in *Journal of Engineering and Applied Sciences*, Vol-12, No-3, pp. 691-698, 2017.

[4] K. Lisa, "Detecting multipliers of jihadism on twitter." *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. IEEE, 2015.

[5] P. Wadhwa and M. P. S. Bhatia, "Case Studies in Secure Computing Achievements and Trends," in *Chapter Classification of Radical Messages on Twitter Using Security Associations*, page 273. 2014.

[6] V. Wisdom and R. Gupta, "An Introduction to Twitter data Analysis in Python".

[7] K. Lee, D. Palsetia, R. Narayanan, M. M. A. Patwary, A. Agrawal, and A.Choudhary. Twitter trending topic classification. In *Data Mining Workshops (ICDMW)*, 2011 IEEE 11th International Conference on, pp. 251–258, 2011.

[8] M. Trupthi, S. Pabboju and G. Narasimha, "Sentiment Analysis on Twitter Using Streaming API", in *IEEE International Advance Computing Conference*, 2017.

[9] S. Lee, J. Baker, J. Song and J.C. Wetherbe, "An empirical comparison of four text mining methods". In *IEEE 43rd Hawaii International Conference on System Sciences (HICSS)*, pp. 1-10, 2010.

[10] P.P. Aurora, R.B. Llavori, and J.R. Shulcloper. "Topic discovery based on text mining techniques." in *Information processing & management*, vol-43, no-3, pp.752-768, 2007.

[11] Tweepy, Streaming With Tweepy — tweepy 3.5.0 documentation. [online] Tweepy.readthedocs.io. Available at: http://tweepy.readthedocs.io/en/v3.5.0/streaming_how_to.html, 2017