Scientific Research Publishing

Journal of Computer and Communications, 2020, 8, 50-62
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Improvised Technique for Analyzing Data and Detecting Terrorist Attack Using Machine Learning Approach Based on Twitter Data

Aditi Sarker[1*], Partha Chakraborty[1*], S. M. Shaheen Sha[2*], Mahmuda Khatun[1], Md. Rakib Hasan[3], Kawshik Banerjee[4]

[1]Department of Computer Science and Engineering, Comilla University, Cumilla, Bangladesh
[2]Department of Computer Science and Engineering, Jahangirnagar University, Dhaka, Bangladesh
[3]Department of Information and Communication Technology, Comilla University, Cumilla, Bangladesh
[4]Department of Computer Science and Engineering, Rajshahi University of Engineering & Technology, Rajshahi, Bangladesh
Email: *aditisarker407@gmail.com, *partha.chak@cou.ac.bd, *shaheencse20@gmail.com, mahmuda@cou.ac.bd, rakib@cou.ac.bd, k.banerjee2024@gmail.com

How to cite this paper: Sarker, A., Chakraborty, P., Sha, S.M.S., Khatun, M., Hasan, Md.R. and Banerjee, K. (2020) Improvised Technique for Analyzing Data and Detecting Terrorist Attack Using Machine Learning Approach Based on Twitter Data. *Journal of Computer and Communications*, **8**, 50-62.
https://doi.org/10.4236/jcc.2020.87005

Received: May 19, 2020
Accepted: July 27, 2020
Published: July 30, 2020

Abstract

Social media are interactive computer mediated technology that facilitates the sharing of information via virtual communities and networks. And Twitter is one of the most popular social media for social interaction and microblogging. This paper introduces an improved system model to analyze twitter data and detect terrorist attack event. In this model, a ternary search is used to find the weights of predefined keywords and the Aho-Corasick algorithm is applied to perform pattern matching and assign the weight which is the main contribution of this paper. Weights are categorized into three categories: Terror attack, Severe Terror Attack and Normal Data and the weights are used as attributes for classification. K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) are two machine learning algorithms used to predict whether a terror attack happened or not. We compare the accuracy with our actual data by using confusion matrix and measure whether our result is right or wrong and the achieved result shows that the proposed model performs better.

Keywords

Terrorist, Ternary Search, Aho-Corasick Automata, Confusion Matrix

1. Introduction

The main idea of this paper mainly comes from the Holey Artisan terrorist attack which was happened on 2016 in Bangladesh and some people were updating their status on social media for getting help. Police and general people get to know about more useful information and inside situation from their social media status.

Our main purpose is to develop a system to detect terror events committed by terrorists by analyzing terrorist attack related tweets from twitter. Twitter is not just a platform for broadcasting information but an informative interaction. In order to develop a strong security system to prevent this attack, people have now adopted sophisticated mechanisms with the help of various modern technologies.

Based on paper [1] which was mainly developed to get detecting phase during natural disaster, in this paper we try to develop an improved technique for detection of terrorist attack event and we applied Aho-Corasick algorithm to perform pattern matching so that we can assign weights to extracted tweeted words, which was not applied previously.

When the terrorist attack happens, people normally broadcast this information using twitter. Analyzing tweeted data, we can detect a certain terrorist attacks and where it took place. Hence, researchers have paid attention in the past few years on the study of supervised machine learning classifiers in order to analyze tweeted data.

The work of [2] discussed the prediction of whether a person is terrorist or not based on the social network analysis and pattern classification. To predict the terrorist attacks, SVM was shown to be more accurate classifier compared to other classifiers especially NB and KNN [3]. Better accuracy was achieved in predicating terrorist group works using the combination of various predictive models [4]. An approach was proposed to build a dictionary using tweets containing hashtags like Al-Qaeda, Jihad, Terrorism, and Extremism and by collecting the relevant words [5]. Another approach using ISIS related tweets to predict the future support was developed [6], where twitter data is used to study the antecedents of ISIS support of users. Go *et al.* [7] have done another study in tweets sentiment classification. Another machine learning based approaches [8] [9] are very familiar in this context. The main objectives of this paper are to filter of extracting for finding the words and geo-location, to introduce Pattern Matching and Weight Assigning (PMWA) machine for finding the weights of filtered words and to calculate the weighted sum of tweeted words in linear time complexity by applying Aho-Corasick automata.

The rest of this paper is described as: Section 2 describes the theoretical model, Section 3 explains the system model, Section 4 shows the experimental results and finally Section 5 concludes the overall works and the future plan.

2. Theoretical Model

2.1. Twitter 4j API

To collect data from twitter, Twitter 4j API is used. Twitter 4j is a Java library for the Twitter API [10]. Java application can be used easily with the twitter service by Twitter 4j. Most recent twitter data mainly 1 or 2 week data based on hashtags can get using streaming API through which data can be extracted.

2.2. Aho-Corasick Automata

The Aho-Corasick algorithm is a kind of dictionary-matching algorithm which was developed by Alfred V. Aho and Margaret J. Corasick that locates the elements of a finite set of strings (the "dictionary") within an input text [11]. It matches all strings simultaneously. The complexity of the algorithm is linear in terms of the length of the strings plus the length of the searched text plus the number of output matches. A finite-state machine (FSM) that quadrate a tire with additional links between the various internal nodes is formed by this algorithm. For each string from a set say whether it occurs in the text can be searched using this algorithm. For example, indicate the first occurrence of a string in the text in O (|P| + |Q|), where |P| is the total length of the text, and |Q| is the total length of the pattern.

Array π[i] = max(x): a[0 … n) = s(i - x..i], *i.e.*, π[i] is the length of the longest own suffix that matches the prefix of the substring [0 … i] to know what is the length of the longest suffix of some text P which is also the prefix of the string Q need to build automata. Need to add characters to the end of the text, quickly recounting this information. One by one feed the automaton with text add character to it, corresponding to the longest own suffix of the current state we can say the suffix link is a pointer to the state. It is easy to see that suffix links on such automatons is the same as π from KMP. For given string Q we can answer the queries whether it is a substring of text P. Uniting our pattern set in trie. At each vertex of trie will be stored suffix link to the state corresponding to the largest suffix of the path to the given vertex in automata which is present in the trie that it is absolutely the same way as it is done in the prefix automaton. It remains only to learn how to obtain these links. In this way, run a breadth-first search from the root. Then we "push" the suffix links to all its descendants in trie with the same principle.

We applied Aho-Corasick algorithm to perform pattern matching so that we can assign weights to extract tweeted words. Sometimes it is difficult to understand that which type of tweet it is because of having some extra word in the keyword. We match every word of it.

We assign a weight of a tweet based on the matching keyword. For declaring the phases of a tweet weight is needed.

2.3. Ternary Search

Ternary search is a technique for finding the extreme point of unimodal functions. By unimodal function means it has one of two behaviors 1) function strictly increases first, reaches a maximum and then strictly decreases and 2) function strictly decreases first, reaches a minimum, and then strictly increases. In a ternary search we divide a search points into three parts, and then we discard one part where our result doesn't exist for sure. Suppose $f(x)$ is a unimodal function on an interval [l, r], we want to fine the maximum of the function. Consider 2 points m1 and m2 where l & lt; m1 & lt; m2 & lt; r and values for these points are

f(m1) and f(m2). Now we get one of three options, first one, f(m1) & lt; f(m2) for this we discard the left side of m1, since the desired maximum cannot be located on the interval [l, m1] and the maximum is located in the segment [m1, r]. Second one f(m1) & gt; f(m2) For this we discard the right side of m2, the maximum is located on [l, m2], so discard the interval [m2, r]. Third one (m1) = f(m2) for this case, we discard either left interval [l, m1] or right interval [m2, r]. Then we can replace the current interval [l, r] with l = m1 or r = m2 and we will repeat this until the difference of l and r is not close enough. Then the average of l and r will be our desired maximum. The most common way of choosing m1 and m2 is: m1 = l + (r – l)/3 and m2 = r – (r – l)/3.

For finding a maximum or minimum point in the U-shape graph, the ternary search is the best choice. So in this paper ternary search is used to define the time interval in which tweets were published on twitter. A ternary search [12] [13] determines either that the minimum or maximum cannot be in the first third of the domain or it cannot be in the last third of the domain, then repeats on the remaining two-thirds. For calculating the weight we need max time and min time of tweets (in minutes) which is in the file, also need a ratio which we identified using a ternary search.

Predefined Data

We use predefined word and then assign. The examples are given in **Table 1**.

We formulate an equation to assign a weight of a tweet. The equation is:

$$\text{Weight of a tweet} = (\text{maxtime} - \text{mintime}) * \text{ration} * \text{number of matches} \tag{1}$$

The ratio is found by using ternary search. Lower bound and upper bound of ternary search is 0.0 and 1.0. The actual value is calculated by taking the average accuracy of five random iterations of each ratio point.

2.4. K-Nearest Neighbor (KNN) Algorithm

We have a new point to classify, we find its K-nearest neighbor from the training data and the new point is assigned from the majority of classes. The distance is calculated by using the following measures: Euclidean, Minkowski, Manhattan.

Table 1. Predefine words.

Predefine Words	
Words	Belongs to
Terrorist attack	Terror attack, severe terror attack
Some people injured	Terror attack
Many died	Severe terror attack
Bombing	Severe terror attack
Shooting	Terror attack
Aniversary of attack	normal

$$\text{Euclidean: } \sqrt{\sum_{i=1}^{k}\left(p_i - q_i\right)^2} \tag{2}$$

$$\text{Minkowski: } \sum_{i=1}^{k}\left|p_i - q_i\right| \tag{3}$$

$$\text{Manhattan: } \left(\sum_{i=1}^{k}\left(\left|p_i - q_i\right|\right)^x\right)^{1/y} \tag{4}$$

Hamming Distance is used when there is an issue of standardization of the numerical variables between 0 and 1. When there is a mixture of numerical and categorical variables in the dataset.

$$\text{Hamming: } \sum_{i=1}^{k}\left|x_i - y_i\right| \tag{5}$$

2.5. Support Vector Machine (SVM)

SVM is a binary classification algorithm. Say, we have a set of N example points x_i belonging to two classes indicated by 1 and −1. If these points come up with their class labels y_i. Then data set can be written as $\left\{\left(x_1,1\right),\cdots,\left(x_N,y_N\right)\right\}$. SVM can be used to classify both linearly separable and non-separable data sets. KNN algorithm is easy to understand and easy to implement. This classification algorithm helps predict the data. As this method contributes to predict data, we compare this predicted data with our actual data and can measure whether our result is right or wrong.

2.6. Confusion Matrix

Information about actual and predicted classifications done is contained by the Confusion Matrix. By a classification system and describes the performance of a classifier model. To calculate the accuracy level confusion matrix is used here.

3. Proposed Model

We extracted social data using Twitter 4j API in JAVA based on Hashtags. Pattern Matching and Weight Assigning (PMWA) Machine is introduced to find the weights of filtered words. It has two parts, one is Aho-Corasick automata and another one is ternary search to assign weights of pre-defined words. For calculating the weight we need max time and min time of tweets (in minutes) which is in the file, also need a ratio which we identified using a ternary search. We classify data into three categories. For classifying these data, we use a machine learning algorithm that is KNN and SVM to predict the different phases and also use confusion matrix for calculating the accuracy of our experiment. Then we use matplotlib library for visualizing our work and create scatter diagram, pie-chart, earth-map and confusion matrix to count the number of true positive predictions and false positive predictions. Figure 1 shows the work flow diagram of our proposed method.

After a terror attack event, many tweets can be tracked from the Twitter. We use this social media for tracking terror attack relevant data, and this data is needed for classifying into three phases which we were determined in our work. Beside public data is very easy. Without extracting this data, we could not be able to train our algorithm and therefore use twitter data.

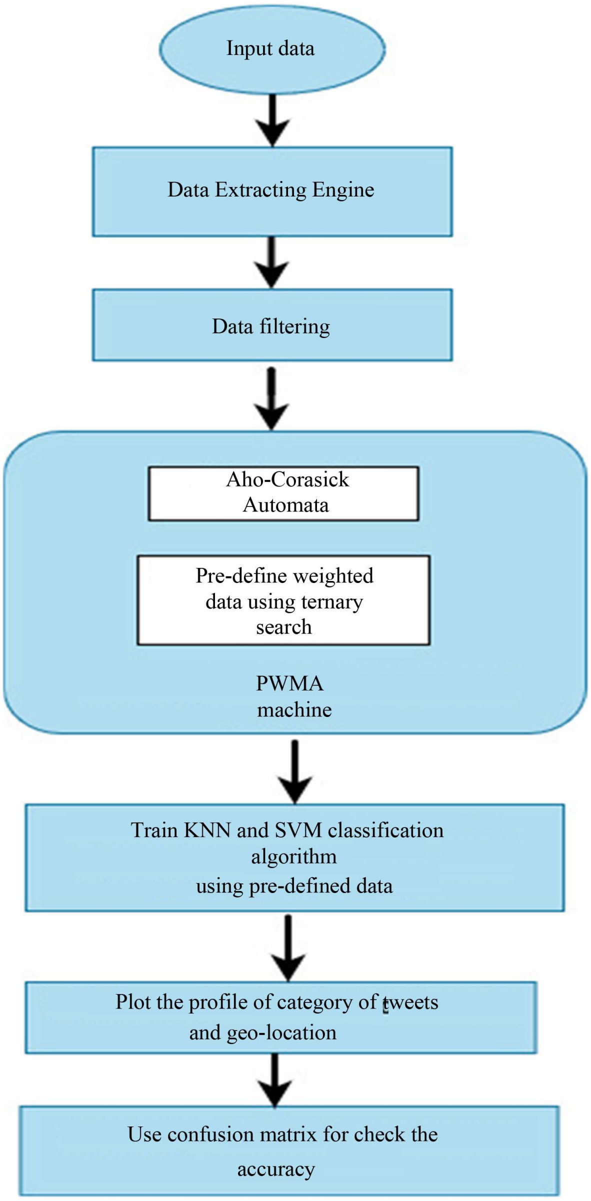

Input data
Data Extracting Engine
Data filtering
Aho-Corasick Automata
Pre-define weighted data using ternary search
PWMA machine
Train KNN and SVM classification algorithm using pre-defined data
Plot the profile of category of tweets and geo-location
Use confusion matrix for check the accuracy

Figure 1. Detection of terrorist attack event workflow.

4. Experimental Result

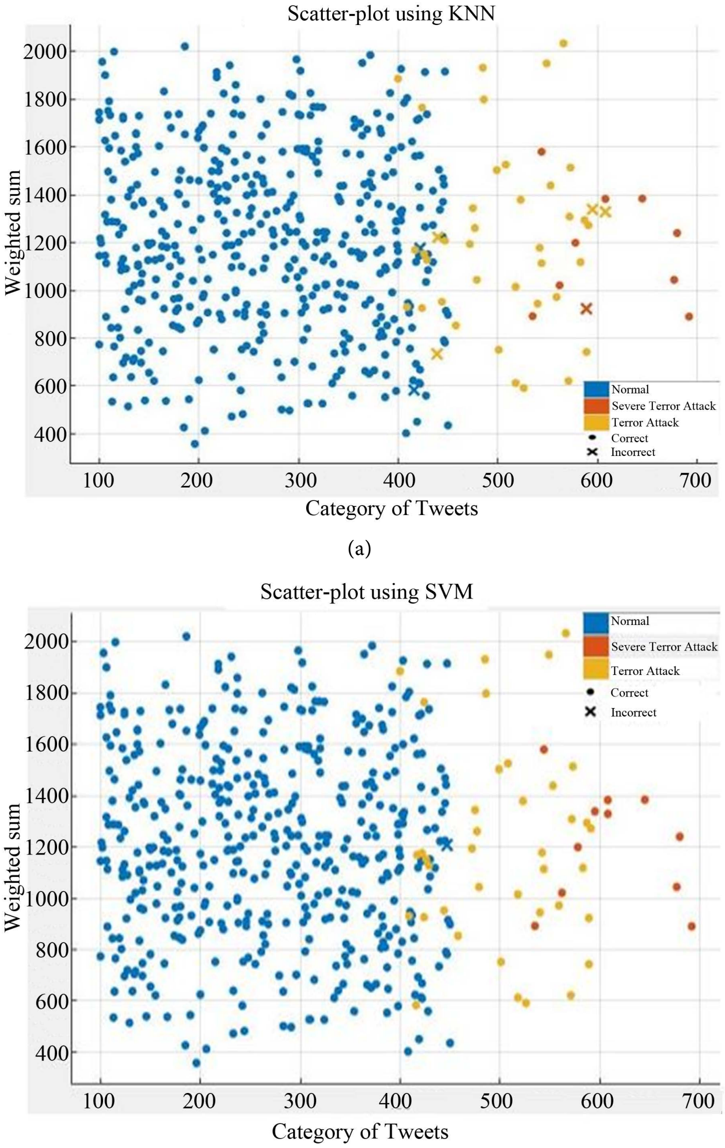
Here we use two datasets. The first Data set contains more than 1000 tweets and another one contains 250 tweets. For each dataset we took a different amount of tweets of different location and time. We used scatter-plot to visualize the classification using KNN and SVM. We visually represent the percentage of the actual value and the predicted value of different classes using a pie chart. We also used Scatter diagram, confusion matrix and earth map.

For Dataset I

Figure 2 shows the profile of category of tweets by plotting tweeted data against the weighted sum taking a normal data weighted sum, terror attack data weighted sum, severe terror attack data weighted sum as parameters. Both KNN and SVM provides almost same classification accuracy with increase in number of tweet data. But the result of SVM outperforms that of KNN with the decrease in the number of tweet data.

Figure 2(a) represents the scatter plot of the category of tweets against weighted sum using KNN classifiers and Figure 2(b) represents the scatter plot using SVM classifier. Here circle represents correct prediction and cross mark represents the incorrect prediction. Different colors are used to represent the categories. Blue is used for normal category, yellow for terror attack and red for severe attack.

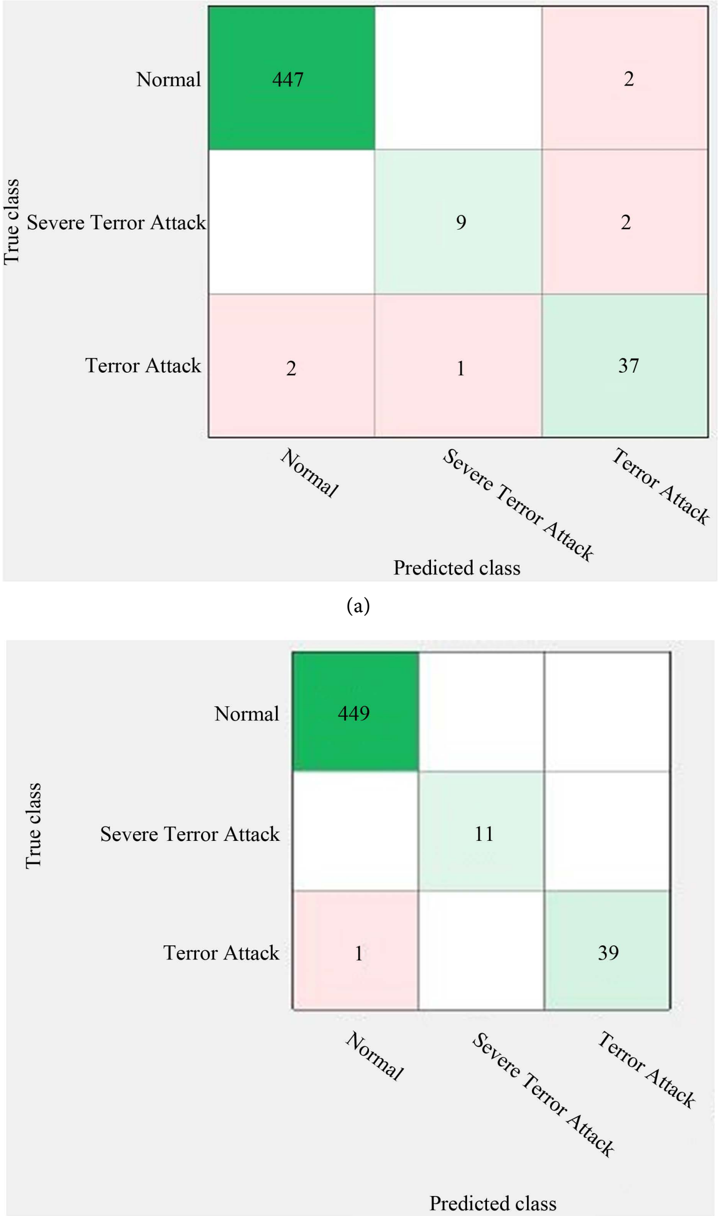
Figure 3 represents the confusion matrix that we got by using KNN and SVM. The Primary diagram represents the true value and the rest of the cell represents the false value. Here we can see that the accuracy of SVM classifier is better than KNN. Confusion Matrix mainly used for checking the accuracy.



(a)

(b)

Figure 2. (a) Scatter-plot using KNN, (b) Scatter-plot using SVM.



(a)

(b)

Figure 3. (a) Confusion matrix of KNN, (b) Confusion matrix of SVM.

By comparing **Figures 4(a)-(c)** we can see that the actual percentage of severe terror attack data is 30.0% and using KNN we got 26.76%, using SVM we got 29.79%. Also for terror attack data in the actual percentage is 26.0%, we got 23.19% using KNN, and 25.82% using SVM. For normal data the actual percentage is 44.0% and predicted percentage using KNN is 50.65% and using SVM it is 44.39%.

Figure 5(a) represents the geological location of actual data using the longitude and latitude of mentioned city or place in tweets. **Figure 5(b)** represents the geological location of predicted data using KNN and **Figure 5(c)** represents the geological location of predicted data using SVM.

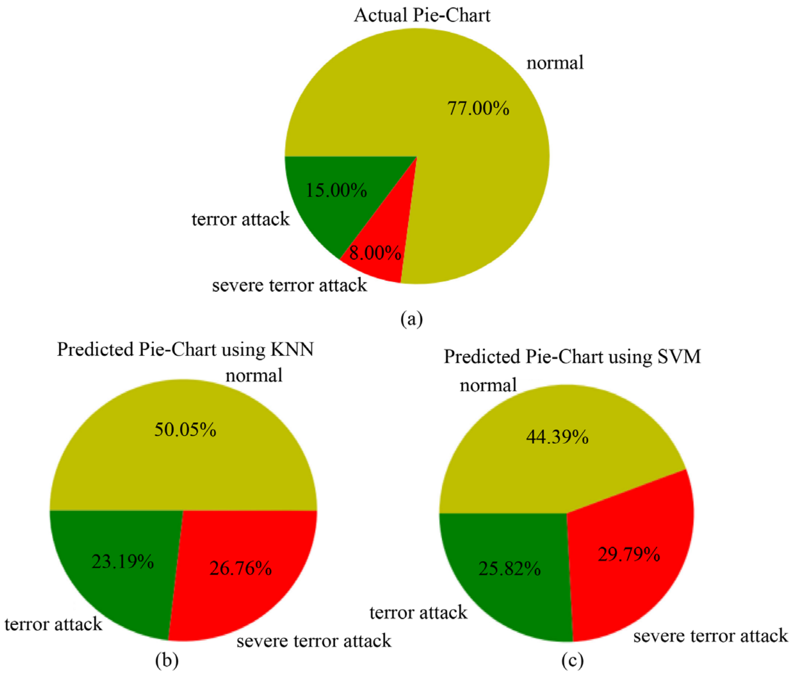


Figure 4. (a) Actual pie chart, (b) Predicted pie chart using KNN, (c) Predicted pie chart using SVM.

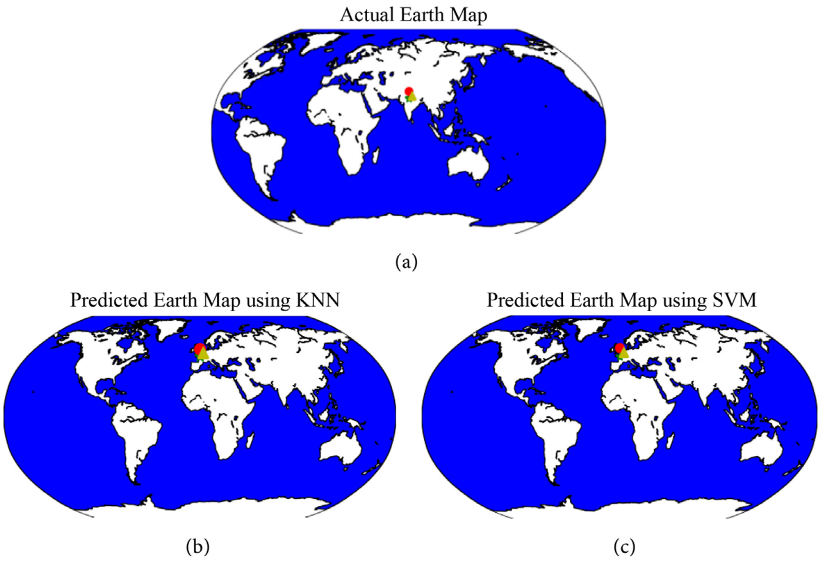


Figure 5. (a) Actual earth map, (b) Predicted of earth map using KNN, (c) Prediction of earth map using SVM.

For Dataset II

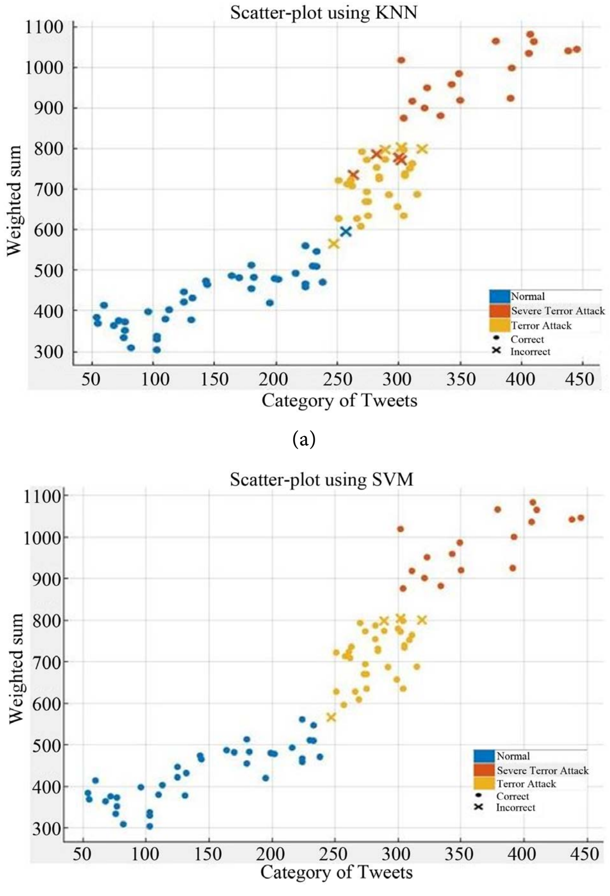
Dataset II mainly contain 250 tweets. Figure 6 shows the profile of category of tweets by plotting tweeted data against weighted sum taking normal data weighted sum, terror attack data weighted sum, severe terror attack data weighted sum as parameters. Both KNN and SVM provides almost same classification accuracy with increase in number of tweet data. But the result of SVM outperforms that of KNN with the decrease in the number of tweet data.

Figure 6 represents the scatter plot of weighted sum versus category of tweets using KNN and SVM classifiers. The circle represents the correct prediction and cross marks represents the incorrect prediction. Different of colors is used to represents the categories. Blue is used for normal category, yellow for terror attack and red for severe attack.

Figure 7 represents the confusion matrix that we got by using KNN and SVM. The Primary diagram represents the true value and the rest of the cell represents the false value. Here we can see that the accuracy of SVM classifier is better than KNN.

By comparing Figures 8(a)-(c) we can see that the actual percentage of severe terror attack data is 30.0% and using KNN we got 26.76%, using SVM we got 29.79%. Also for terror attack data in the actual percentage is 26.0%, we got 23.19% using KNN, and 25.82% using SVM. For normal data the actual percentage is 44.0% and predicted percentage using KNN is 50.65% and using SVM it is 44.39%.

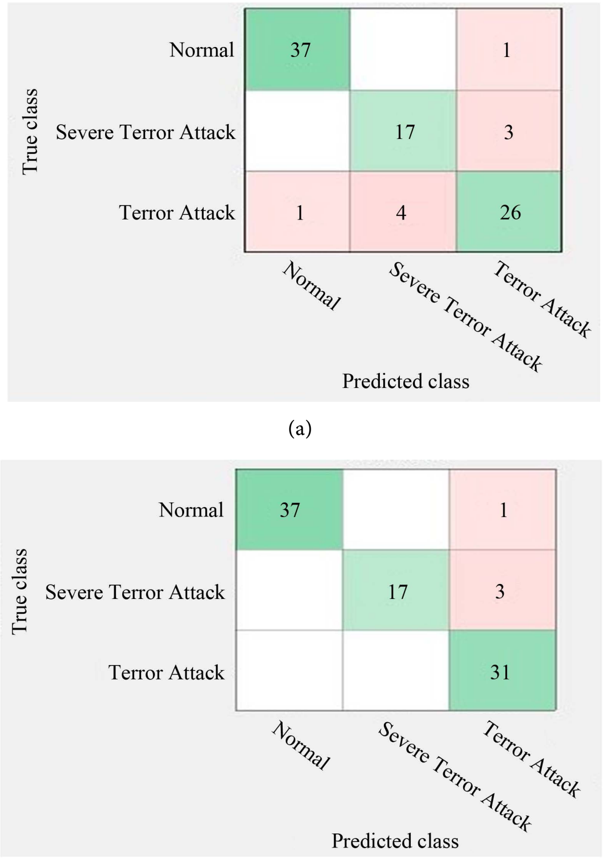
Figure 9(a) represents the geological location of actual data using the longitude and latitude of mentioned city or place in tweets. For extracting the location of the city we used a python library which is called geopy. Figure 9(b) represents the geological location of predicted data using KNN and Figure 9(c) represents the geological location of predicted data using SVM.



(a)

(b)

Figure 6. (a) Scattered plot using KNN, (b) Scattered plot using SVM.



True class \ Predicted class	Normal	Severe Terror Attack	Terror Attack
Normal	37		1
Severe Terror Attack		17	3
Terror Attack	1	4	26

(a)

True class \ Predicted class	Normal	Severe Terror Attack	Terror Attack
Normal	37		1
Severe Terror Attack		17	3
Terror Attack			31

(b)

Figure 7. (a) Confusion matrix of KNN, (b) Confusion matrix of SVM.

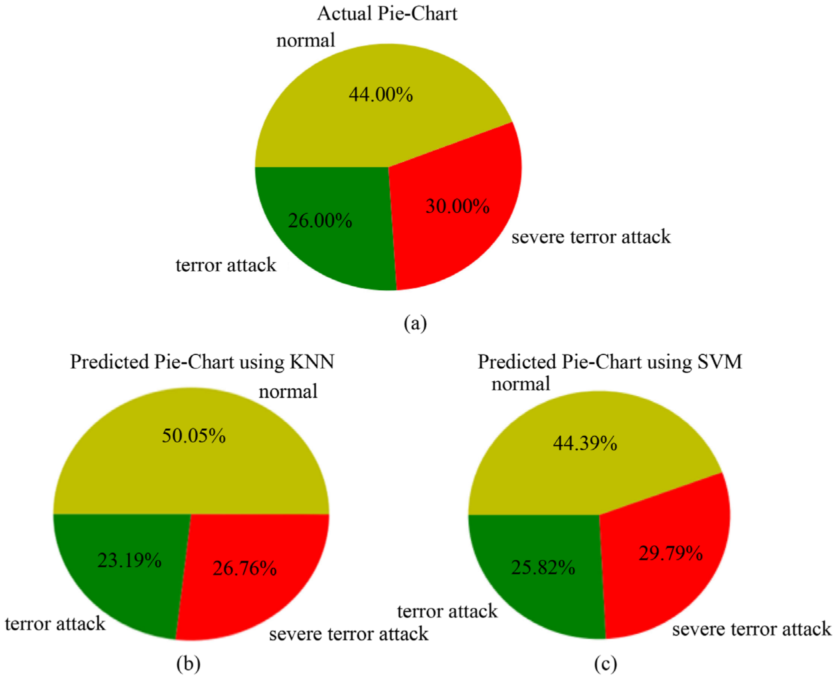
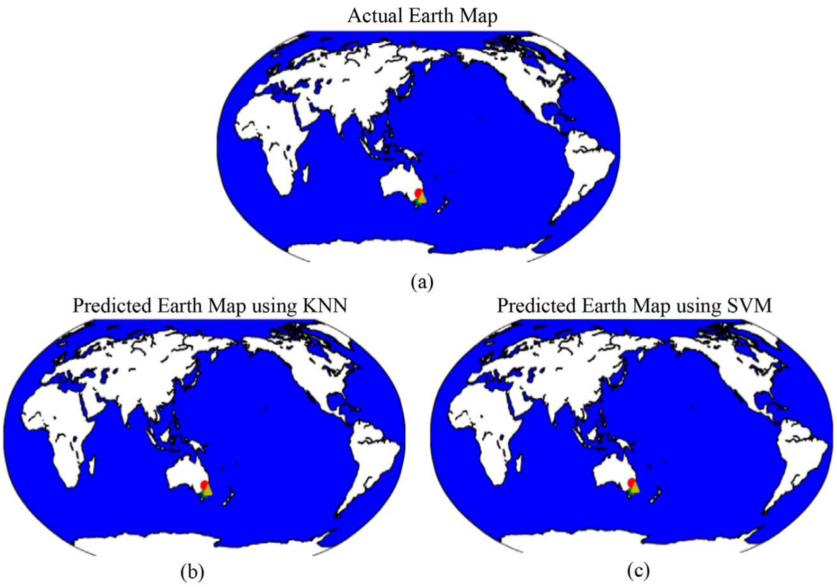


Figure 8. (a) Actual pie chart, (b) Predicted pie chart using KNN, (c) Predicted pie chart using SVM.



Actual Earth Map

(a)

Predicted Earth Map using KNN

Predicted Earth Map using SVM

(b)

(c)

Figure 9. (a) Actual earth map, (b) Predicted of earth map using KNN, (c) Prediction of earth map using SVM.

Table 2. Comparison the classification accuracy of KNN and SVM.

	Comparison the classification accuracy	
	Accuracy percentage using KNN	Accuracy percentage using SVM
Data Set I	97.1%	97.7
Data Set II	89.4	92.4

From above comparison Table 2, we find that SVM gives better classification accuracy than KNN.

5. Conclusions and Future Work

An improved system model is proposed in this research for the detection of terrorist attack event those terrorist attacks that already took place by analyzing social data collected from twitter. We extract words using string parsing and do pattern matching by applying Aho-Corasick algorithm that runs linear time complexity in order to find out the weights of tweeted words. To categorize those data, the weighted sum is passed to both KNN and SVM classifier and our achieved results show that the proposed model performs better. The main limitation of this research work is that, datasets are not affluent enough to show the accurate result.

In future, we intend to further develop our system in order to identify the terror attacks before happening and prevent them by changing the keyword selection and pattern of related post from terrorist people and messaging data analysis; and for classification we are going to use another machine learning classification or clustering algorithm. This method can also be applied to detect COVID-19 affected area.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Huq, Md.R., Abdullah-Al-Mosharraf and Rahma, K. (2017) Data Analysis and Phase Detection During Natural Disaster Based on Social Data. *International Journal on Computational Science & Applications* (*IJCSA*), **7**, No.1/2/3. https://doi.org/10.5121/ijcsa.2017.7301

[2] Coffman, T.R. and Marcus, S.E. (2004) Pattern Classification in Social Network Analysis: A Case Study. 2004 *IEEE Aerospace Conference Proceedings*, Big Sky, MT, 6-13 March 2004, 3162-3175.

[3] Tolan, G.M. and Soliman, O.S. (2015) An Experimental Study of Classification Algorithms for Terrorism Prediction. *International Journal of Knowledge Engineering*, **1**, 107-112. https://doi.org/10.7763/IJKE.2015.V1.18

[4] Faryral, G., Wasi, B.H. and Usman, Q. (2004) Terrorist Group Prediction Using Data Classification. *Proceedings of the International Conferences of Artificial Intelligence and Pattern Recognition*, Kuala Lumpur, Malaysia, 17-19 November 2014, 199-208.

[5] Khorshid, M.M., Abou-El-Enien, T.H. and Soliman, G.M. (2015) Hybrid Classification Algorithms for Terrorism Prediction in Middle East and North Africa. *International Journal of Emerging Trends & Technology in Computer Science*, **4**, 23-29.

[6] Magdy, W., Darwish, K., and Weber, I. (2005) #FailedRevolutions: Using Twitter to Study the Antecedents of ISIS Support. arXiv Preprint, arXiv: 1503.02401.

[7] Go, A., Bhayani, R. and Huang, L. (2009) Twitter Sentiment Classification Using Distant Supervision. CS224N Project Report, Stanford, 1-12.

[8] Chakraborty, P., Zahidur, Md., & Rahman, S. (2019) Movie Success Prediction Using Historical and Current Data Mining. *International Journal of Computer Applications*, **178**, 1-5. https://doi.org/10.5120/ijca2019919415

[9] Zulfiker, M.S., Kabir, N., Biswas, A.A., Chakraborty, P. and Rahman, M.M. (2020) Predicting Students' Performance of the Private Universities of Bangladesh Using Machine Learning Approaches. *International Journal of Advanced Computer Science and Applications*, **11**, 672-679. https://doi.org/10.14569/IJACSA.2020.0110383

[10] McNamee, L.G., Peterson, B.L. and Peña, J. (2010) A Call to Educate, Participate, Invoke and Indict: Understanding the Communication of Online Hate Groups. *Communication Monographs*, **77**, 257-280.

[11] Ourlis Lazhar, Bellala Djamel (2019) SIMD Implementation of the Aho-Corasick Algorithm Using Intel AVX2. *Scalable Computing*: *Practice and Experience* (*SCPE*), **20**, 563-576. https://doi.org/10.12694/scpe.v20i3.1572

[12] Fisher, A. (2015) How Jihadist Networks, Maintain a Persistent Online Presence. *Perspectives on Terrorism*, **9**, 3-20.

[13] Chris Hale, W. (2012) Extremism on the World Wide Web: A Research Review. *Criminal Justice Studies*, **25**, 343-356. https://doi.org/10.1080/1478601X.2012.704723