Mining the Global Terrorism Dataset using Machine Learning Algorithms

Alaa S. Alsaedi
Department of Computer Science and Engineering
College of Computer Science
Taibah University
Madinah, Saudi Arabia
asaedi@taibahu.edu.sa

Arwa S.Almobarak
Department of Computer Science and Engineering
College of Computer Science
Taibah University
Madinah, Saudi Arabia
arwa.almobarak777@gmail.com

Saad T. Alharbi
Department of Computer Science and Engineering
College of Computer Science
Taibah University
Madinah, Saudi Arabia
stharbi@taibahu.edu.sa

Abstract—**From the beginning of the present century, terrorist attacks have represented a massive concern for both developed and developing countries alike. Therefore, countries spare no effort to use all available means to fight and eradicate them. Due to this, we aim through this paper, to take the advantages of the available data mining and machine learning techniques, by applying them to the Global Terrorism Database (GTD) dataset in order to acquire valuable information about the predicted attacks and attackers. We believe that these models will be of great benefit when utilized by governments and intelligence agencies; since they help them to make pre-emptive strikes against terrorist groups quickly and over a short period of time. In our paper, we focus on two tasks: predicting the success of an attack and predicting the identity of the terrorist organization that is behind the attack. To implement this, we adopted three machine learning algorithms: K-nearest neighbor (KNN), Naive Bayes (NB) and Random Forest (RF), which we used to train our models. More specifically, each algorithm was used to construct two models, one for each task, where the data were sampled for the first model using the holdout method, while for the second model cross-validation was employed. In the end, we compared the performance of the models in terms of accuracy, precision, recall and F-measure metrics. We noticed that the RF models outperformed the other models, while the NB models were the least efficient among the three algorithm models.**

Keywords— ***classification, K nearest neighbor, Naïve Bays, Random Forest, Cross Validation, Accuracy, global terrorism***

I. INTRODUCTION

In the last 20 years, terrorist attacks have increased throughout the world's countries, becoming a crucial influencing factor for international politics [1]. Thus, governments have been paying more attention to national security issues and focusing on how to counter the terrorism. Recently, machine learning (ML) techniques are used in order to find the valuable hidden knowledge behind attack events and by which the responsible experts expect to get a clear comprehensive picture of what the terrorists are thinking, in order to increase defenses against their terrorism acts. Currently, terrorist attack classification is basically carried out by field experts [2]. This, however, may be time consuming and may add subjectivity into the results. To better understand the existing distribution of terrorist attacks, ML could be used as an alternative. Since ML is a powerful method for classification, it is used in many research areas, such as, sentiment analysis, social software, biomedicine and healthcare [3]. Moreover, it assists experts in decisions making. Inspired by these ML benefits, we aim to apply K-nearest neighborhood (KNN), naive Bayes (NB) and random forest (RF) to the Global Terrorism Database (GTD) to build our models, and in order to achieve the following goals. First, we aim to construct three models that are able to accurately predict the possibility of success of an attack based on specific information, using the three mentioned classification algorithms and depending on the holdout sampling method; second, to build three models that are able to accurately predict the identity of a terrorist organization that is behind an attack, provided that we know the necessary information about that attack, using the three algorithms and depending on cross validation sampling; third, to evaluate the accuracy of the three models—NB, KNN, and RF—based on the well-known metrics (accuracy, precision, recall, and F-measure); finally, to evaluate the accuracy of each algorithm, based on the two sampling methods we adopted, in terms of accuracy, precision, recall, and F-measure. This paper is structured as follows: Section 2 introduces a brief review of previous related works. Section 3 describes our methodology in detail. The experimental results are reported and discussed in Section 4. Finally, Section 5 concludes this paper and shows future works.

II. LITERATURE REVIEW

Mo et al. [4] used the GTD to construct models that predict terrorist events. They employed three different algorithms: naive Bayes, support vector machine and logistic regression. To enhance accuracy, they utilized two methods for attribute selection which are maximal relevance and minimal redundancy maximal relevancy. They found that the logistic regression classifier achieves 78.41%, emphasizing the feasibility of using machine learning techniques in the domain of terrorism. Also, they demonstrated that choosing the proper attribute selection methods would increase model accuracy.

In addition, Snehanshu et al. [5] stated that one of the biggest problems facing the world today is terrorist attacks, which have dangerously evolved in the past decades. For this reason, counterterrorism must also gain huge priority in all countries. Countries now use various techniques to predict future attacks. The authors proposed a method to predict upcoming attacks,

978-1-7281-5052-9/19/$31.00 ©2019 IEEE

targets and the used weapons using two machine learning algorithms. They employed both random forest regression and random forest classification. The study found that the accuracy of attack type achieved was 79% for the classification model versus 41% for the regression model. The accuracy of weapon type prediction for the classification model was 86%, and for the regression model 41%. Finally, due to the small amount of data and target number of classes, target type accuracy was very low.

Some researchers used social networks to predict the attacker's behavior, which is essential but at the same time complex to realize. They clarify that there are distinctive characteristics of members of terrorist groups on social networking sites. In a recent study, Li et al. proposed a framework that consists of three main components to examine the attack behavior for these groups. First, they used social network analysis to extract the group behavior features. Then, the wavelet transforms features of networks that belong to the terrorists were predicted. Finally, they employed pattern recognition methods to identify the group's behavior depending on the association between the network and the behavior. They investigated their framework on the Al-Qaeda dataset, and the results indicated high accuracy of their framework in estimating the terrorist group's behavior [6].

Enders and Sandler [7] proposed to use conventional time-series analysis to introduce a model that measures the autoregressive based on a predefined threshold. In addition, they investigated the waves of terrorist activities throughout the world in the near and long-term. Najgebauer et al. [8] illustrated a proactive warning system that uses analysis of the semantic networks built on an ontology model for estimating the preparation works of terrorist actions.

Tutun et al. [9] suggested a model called "evolutionary simulating annealing lasso logistic regression (ESALLOR)" that is employed to select the significant features to be used in a similarity function in order to understand how terrorists will strike in the future. Next, they developed another similarity function to predict the relationships between the attacks. When they tested the framework on real terrorism actions from 2014–2015, their system reached high accuracy, estimated at 90%.

Gohar et al. [10] showed a novel framework used to predict the terrorist group, which was composed of four popular classifiers: Iterative Dichotomiser 3, naive Bayes, decision stump (DS), and K-nearest neighbor. To combine the results of these classifiers, the majority vote technique was used. They compared the performance of the combined classifiers with the individual classifiers and found that their framework had a better accuracy rate and minimum error rate than the individual classifiers.

III. METHODOLOGY

A. Dataset

We used the GTD, which is a huge, global dataset that includes information about the terrorist attacks around the world from 1970 to 2017. In addition, the GTD has information about 181692 terrorist attacks and contains 182k rows and 135 columns. Moreover, it includes details on the attacks such as date of attack, state, target, location, type of attack, whether the attack succeeded or not and others. The dataset is maintained by researchers at the National Consortium for the Study of Terrorism and Responses to Terrorism (START) headquartered at the University of Maryland. We downloaded the GTD from the Kaggle website [11].

B. Data Preprocessing

1. Data Preprocessing for Task 1: Prediction of attack success

We used the Jupyter Notebook to implement our project, which is considered a powerful tool to develop interactive data science applications [12]. Therefore, we started in Jupyter Notebook with thoroughly inspecting and understanding the dataset to decide on the required preprocessing operations. Then we worked on removing and reducing unnecessary data.
We started the data cleaning process by checking and removing all columns that contained missing values. Fortunately, all columns that contained missing values were not useful in the models training phase. After that, we noticed that some information was duplicated in our dataset: once in numeric form and then in text form, such as, "targtype1_txt," "country_txt," "region_txt," "attacktype1_txt," "weaptype1_txt," "dbsource," "individual," "extended," "INT_LOG," "INT_MISC," "INT_ANY" and "INT_IDEO." In such instances, we used the columns with numeric information only and removed the textual columns.
In addition, to prepare our dataset to be processed efficiently, we used the labeling method, replacing each distinct group name with a specific ID. Meanwhile, each unknown group name was also replaced with a specific ID [13]. Then, we rearranged the dataset columns again to facilitate the creation of the frame model. For example, for the frame that will predict the succession of the attacks; we moved the 'success' column to the end.

After all the preprocessing steps, we needed to make a clear distinction between the features and labels. To achieve this, the data were split into input and output, where all columns—except the last—will contain the features that will be the input for the classification algorithms; while the last column—which represents the labels—will be the outcome from our models. Finally, we allocated 70% of the data for training, and the remainder for testing.

2. Data Preprocessing for Task 2: Identification of the Terrorist Organization

As in Task 1, we dropped all columns that were unnecessary to the identification of organization masterminds behind the attacks, as judged by intuition. In addition, we dropped all text columns whose information was repeated in numeric columns too. Then, we deleted all entries with missing values in one or more columns. Moreover, since we focused on building a model able to predict the terrorist organization that committed

a given terrorist act, we removed all samples whose terrorist group was unknown in the dataset, and kept only the entries whose terrorist group had been active in more than 500 attacks, as shown in Fig. 1.

After that, we replaced the name of each terrorist group with a unique numeric ID so that it was easily processed by the algorithm, and to be the output that the ML algorithm predicts. To distinguish clearly between the features and labels, we split the data again into input and output. Finally, we used the K-fold method to randomly split the data into 10 equal size folds to be evaluated by the classification models (KNN, random forest, naïve Bayes).

Taliban	7478
Islamic State of Iraq and the Levant (ISIL)	5613
Shining Path (SL)	4555
Farabundo Marti National Liberation Front (FMLN)	3351
Al-Shabaab	3288
New People's Army (NPA)	2772
Irish Republican Army (IRA)	2671
Revolutionary Armed Forces of Colombia (FARC)	2487
Boko Haram	2418
Kurdistan Workers' Party (PKK)	2310
Basque Fatherland and Freedom (ETA)	2024
Communist Party of India - Maoist (CPI-Maoist)	1878
Maoists	1630
Liberation Tigers of Tamil Eelam (LTTE)	1606
National Liberation Army of Colombia (ELN)	1561
Tehrik-i-Taliban Pakistan (TTP)	1351
Palestinians	1125
Houthi extremists (Ansar Allah)	1062
Al-Qaida in the Arabian Peninsula (AQAP)	1020
Nicaraguan Democratic Force (FDN)	895
Manuel Rodriguez Patriotic Front (FPMR)	830
Sikh Extremists	716
Corsican National Liberation Front (FLNC)	639
Al-Qaida in Iraq	638
Muslim extremists	632
Donetsk People's Republic	624
African National Congress (South Africa)	607
Separatists	589
Tupac Amaru Revolutionary Movement (MRTA)	557
M-19 (Movement of April 19)	555
Abu Sayyaf Group (ASG)	527
Fulani extremists	511

Fig. 1. Names of terrorist groups that committed more than 500 attacks

Features selection for Task 1 & Task 2

In order to get the highest accuracy from our models, we needed to include only the relevant features and eliminate the irrelevant. So, the columns that we retained are: "country," "region," "vicinity," "a set of three criteria," "whether that attack was a suicide mission or not," "type of attack," "type of persons targeted in that attack," "name of terrorist organization that conducted that attack," "types of weapons used" and "type of property on which it was to be executed."

C. Training Models

After dataset preparation, we trained our models. In this project, we built our models using three classification algorithms that differ in principle of work; because we aimed to compare them in terms of their obtained results. We used, as mentioned earlier, KNN, NB and RF algorithms. Each algorithm was used to build two models for predicting the success of an attack and the identity of the terrorist organization behind the attack. Regarding to the first model, we used holdout sampling, which randomly takes 30% of the data samples for testing and trained the models on the remaining 70% to predict the success of an attack based on all input parameters. While, for the second model we used cross validation sampling.

1. *K-Nearest Neighborhood (KNN) Algorithm*

K-nearest neighbor is one of the most common algorithms used in data mining. It works by assigning a label to a test item based on the nearest k samples that lie closest to the test sample, where the K value, as we noticed from previous studies, is often a small odd value. Moreover, the KNN algorithm measures the distance between the test item and the K-nearest neighbor samples depending on the formula of Euclidian distance, represented in (1):

$$d(X, Y) = \text{sqrt}\left(\sum_{i=1}^{n}(xi - yi)\right) \qquad (1)$$

Then, the majority of those neighbors will determine to which class the test item will belong. K-nearest neighbor is known as a non-parametric lazy algorithm [14], which mean that it does not make any inherent assumptions about distribution of data into classes. This is one of its greatest strengths because in the real world, most of the data are not bound by any fixed boundaries or distributions. So, for cases where we are not sure about the nature of data distribution, KNN would most probably serve us well. K-nearest neighbor is a "lazy" algorithm [15], because it does not use the training data to do any generalization. It keeps all training data and uses it for making predictions during data testing. The most important flaw of this algorithm is that it has a high memory requirement, as all the training data need to be stored to classify the new items. For this reason, it is not suitable for large datasets, as prediction might be very slow.

2. *Bayesian Classification*

This type of classification is considered a statistical classifier, which employs the naive Bayes algorithm. Bayesian classification aims to predict the probability for a particular tuple to be a member in a particular class. The algorithm is based on the well-known theorem, Bayes' theorem, represented in (2). The advantages of this algorithm are embodied in its ease of implementation, its high accuracy especially when dealing with large databases and in that it is the proper algorithm to use with high dimension inputs [16]. In addition, it helps to recognize independent attributes and demonstrate how attribute values are independent of the values of other attributes in the dataset, and that only the outcome is dependent on all the attributes.

$$P(c|x) = \frac{P(c|x)P(c)}{P(x)} \qquad (2)$$

In (2), P(c|x) represents the posterior probability, which means the probability that a class holds a given data sample. The absolute probability for a class is P(c), the likelihood, defined as the probability of data sample given class is P(x|c), and the absolute probability of the data sample is P(x) [17].

To minimize computation cost, we used the assumption that there are no dependent relationships between the dataset attributes. Depending on this, it is easier to calculate the probabilities from the training dataset using (3).

$$\hat{Y} = \arg\max_{Y} P(Y)\prod_{i=1}^{n} P(x_i \mid Y) \qquad (3)$$

3. Random Forest

Random Forest (RF) is a classification and regression method based on the aggregation of many decision trees, where each classifier is generated using a random vector sampled independently from the input vector, and each tree casts each cast vote for the most popular class to classify an input vector. Moreover, The RF method is an ensemble approach that can also be thought of as a form of nearest-neighbor predictor. In addition, its runtimes are quite fast, and it is able to deal with unbalanced and missing data [18], [19]. Equation (4) represents the RF formula,

$$RFfi_i = \frac{\sum_{j \in all\ trees} normfi_{ij}}{T} \qquad (4)$$

where RFfi sub(i) represents the importance of feature i calculated from all trees in the random forest model; normfi sub (ij) represents the normalized feature importance for i in tree j; and T is the total number of trees.

IV. Results and Discussion

Predicting terrorist attacks before they happen and who stands behind them is one of the most important needs that countries are currently seeking in order to increase their security. Today, with machine learning techniques, this is made possible by building models capable of predicting, with high accuracy, the percentage of the success of terrorist attacks. Ethically, we believe that these models will be of great benefit when utilized by governments and intelligence agencies, since they help them to make pre-emptive strikes against terrorist groups quickly and over a short period of time. All these contribute significantly to the preservation of lives and property and provision of the highest levels of security to citizens.

Now, and after building our classifier models, we aim to compare their performance using four well-known metrics: accuracy, precision, recall and F-measure to check which classifier is more suitable for the GTD dataset.

Before analyzing the results, we would like to give a concise overview for each metric. The percentage of the test samples that our classifiers categorized correctly, is the "accuracy," and the "precision" represents the classifier's ability to avoid classifying the negative samples as positive [20]. "Recall" represents the classifier's capability to get all the positive samples. F-score [21] is equal to the mean of both the precision and recall, where the score is considered best if it reaches 1. To compute the metrics values, we need as an initial step to compute the confusion matrix values first, then we move to calculate the metric scores—the confusion matrix with the metric values for each algorithm is presented in the APPENDIX.

Since we have used three classification methods (KNN, NB and RF), we will conduct comparisons in three levels. First, we will compare them when the data are sampled into 70:30, using the holdout method; then we will contrast them when the data are sampled using cross validation. At the end, we will examine at one algorithm level and show how employing different sampling methods can affect the classifier performance.

In our models, both precision and recall are critical, because we want our models to achieve the highest level of recall to be able to predict all possible terrorist events. We also want our models to have the highest level of precision in order to save the government money and prevent any possible waste of time and efforts that may happen as a result of the mobilization of security authorities. For this we used the F-1 score metric to obtain the harmonic mean of precision and recall.

A. Classifier Performance Using the Holdout Method:

We constructed three models to predict the possibility of the success of an attack based on specific information. We used NB, KNN and RF algorithms and the data sampled as 70% training data and 30% testing data. The performance of the three models is listed in Table 1. We noticed that classifiers differ slightly in their accuracy, where random forest achieved 91.62%, which was higher than KNN (90.84%) and NB (85.53%).

TABLE 1: The Performance of Classifiers for the First Model with the Holdout Sampling Method

Dataset	Classifier	Evaluation Measures			
		Accuracy	Precision	Recall	F1-score
Global Terrorism Database (GTD)	KNN	90.84 %	0.90	0.91	0.90
	NB	85.53%	0.82	0.86	0.84
	RF	91.62%	0.91	0.92	0.91

Moreover, due to the sensitivity and importance of the results that our system will achieve, we measured the precision, recall and F-score to assess the model's performance, see Fig. 2. We found that the precision, recall and F-score for the three models were almost identical, which proves that our models classify the results almost accurately. We noticed also that the RF classifier confirmed its superiority over KNN and NB.

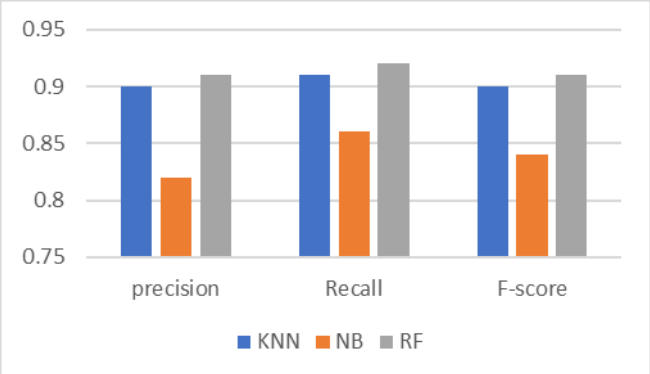

Fig. 2. The results of the classifiers comparison through Model 1 in terms of (precision, recall and F-score)

B. *Classifier Performance Using the Cross Validation Method*

Furthermore, to predict the identity of a terrorist organization that is behind an attack, another three models were implemented using the three algorithms and depending on the cross validation (or K-fold) sampling method. The data were divided into ten folds, and then evaluated by the KNN, NB and RF classifiers. We constructed three models to predict the possibility of the success of an attack based on specific information. Our achieved results from the three models are demonstrated in Table 2. We noticed that the classifiers differ slightly in their performance, where the RF outperforms the others, as shown in Fig. 3.

TABLE 2: The Performance of Classifiers for the Second Model with Cross Validation Sampling

Dataset	Classifier	Evaluation Measures			
		Accuracy	Precision	Rec-all	F1-score
Global Terrorism Database (GTD)	KNN	91.059%	0.89	0.90	0.89
	NB	74.7898%	0.77	0.74	0.73
	RF	91.8660%	0.90	0.91	0.90

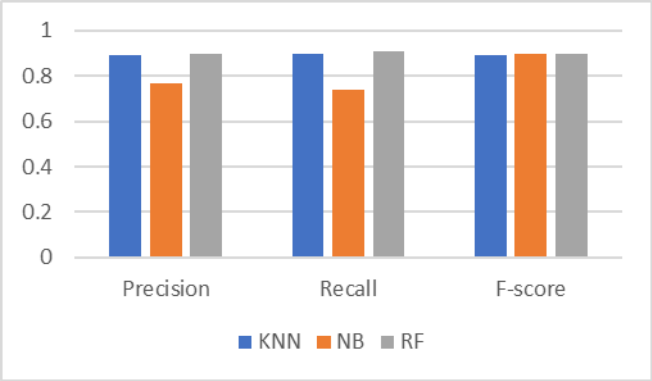

Fig. 3. The results of the classifier comparison through Model 2 in terms of (precision, recall and F-score)

C. *Comparison Between the Sampling Methods at the Algorithm Level*

Here, we compare the three algorithms we used to construct our models, in order to explore to what extent, the sampling method may affect the performance results of each algorithm. In terms of accuracy, both KNN and RF were not affected by changing the sampling method, as shown in Fig. 4. In contrast, NB models were markedly affected when cross-validation sampling was employed. Also, Table 3 shows that the KNN and RF algorithms show slight variations through the two sampling methods, while the performance achieved by the NB algorithm through Model 2 was higher than that achieved through Model 1.

We believe that the reasons that influenced the performance of the NB algorithm, making it perform less than the RF and KNN algorithms, include that NB is one of the algorithms affected by the number of attributes and data types.

TABLE 3: The performance of KNN, NB and RF among our models and two sampling methods

Algorithm	Metrics	Model 1 (Holdout Method)	Model 2 (Cross-Validation Method)
KNN	Accuracy	90.84 %	91.059 %
	Precision	0.90	0.89
	Recall	0.91	0.90
	F1-score	0.90	0.89
NB	Accuracy	85.53%	74.7898%
	Precision	0.82	0.77
	Recall	0.86	0.74
	F1-score	0.84	0.73
RF	Accuracy	91.62%	91.8660%
	Precision	0.91	0.90
	Recall	0.92	0.91
	F1-score	0.91	0.90

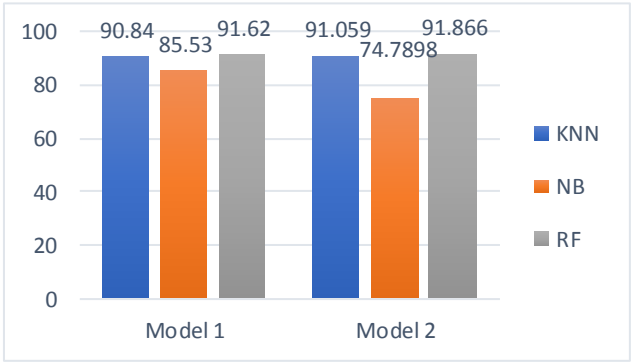

Fig. 4. The results of the classifier comparison through Model 1 and Model 2 in terms of accuracy

Naive Bayes performs well with text data while our models rely on numeric data, and the attributes are dependent on each other. All these reasons negatively affected the NB performance in terms of accuracy, recall and precision.

V. Conclusion and Future Work

We have proven that ML techniques can be used to predict the success of terrorist attacks and the identity of the terrorist organization behind the attacks with high accuracy. Our work began by preparing and cleaning the GTD dataset. Data were manipulated using two different sampling methods: holdout and cross-validation, to obtain a comprehensive performance of the classifiers. The model was trained using three ML algorithms: K-nearest neighbor, naive Bayes and random forest. We conclude that RF outperforms KNN and NB in terms of four evaluation measures (accuracy, precision, recall and F1-score). Moreover, the KNN classifier has the second highest performance, and NB achieved the lowest. Machine learning can facilitate for researchers in the field of counterterrorism a better understanding of the current situation, which demonstrates that it is truly a promising research subject and deserves more attention. As future work, we intend to explore more ML algorithms that may give us higher levels of accuracy. Also, we aim to enhance our work by employing an ensemble technique to combine the output of the three classifiers to achieve the highest possible accuracy.

REFERENCES

[1] A. Sachan and D. Roy, "TGPM: Terrorist group prediction model for counter terrorism", in International Journal Of Computer Applications (0975 – 8887) Vol. 44– No10, April 2012.

[2] L. Li and X. Zhang, "Study of data mining algorithm based on decision tree", in 201O International Conference On Computer Design And Appliations (ICCDA 2010).

[3] K. Singh and S. Bhasin, "Modification of gtd from flat file format to OLAP for data mining", in International Journal Of Innovative Technology & Creative Engineering (ISSN:2045-8711) VOL .1 NO.4 APRIL 2011.

[4] H. Mo, X. Meng, J. Li, and S. Zhao, "Terrorist event prediction based on revealing data," in *Big Data Analysis (ICBDA), 2017 IEEE 2nd International Conference on*, 2017, pp. 239-244.

[5] S. Snehanshu,H. Aladi, A.Kurian, B. Aparna, *Future Terrorist Attack Prediction using Machine Learning Techniques*, 2017, 10.13140/RG.2.2.17157.96488.

[6] Z. Li, D. Sun, B. Li, Z. Li, and A. Li, "Terrorist Group Behavior Prediction by Wavelet Transform-Based Pattern Recognition," *Discrete Dynamics in Nature and Society,* vol. 2018, 2018.

[7] W. Enders and T. Sandler, "Is transnational terrorism becoming more threatening? A time-series investigation," *Journal of Conflict Resolution,* vol. 44, pp. 307-332, 2000.

[8] A. Najgebauer, R. Antkiewicz, M. Chmielewski, and R. Kasprzak, "The prediction of terrorist threat on the basis of semantic association acquisition and complex network evolution," *Journal of Telecommunications and Information Technology,* pp. 14-20, 2008.

[9] S. Tutun, M. T. Khasawneh, and J. Zhuang, "New framework that uses patterns and relations to understand terrorist behaviors," *Expert Systems with Applications,* vol. 78, pp. 358-375, 2017.

[10] F. Gohar, W. H. Butt, and U. Qamar, "Terrorist Group Prediction Using Data Classification," in *Work. MultiRelational Data Min. MRDM2003*, 2014, pp. 199-208.

[11] Miller, E. (2018). *Global Terrorism Database*. [online] Kaggle.com. Available at: https://www.kaggle.com/START-UMD/gtd [Accessed 18 Nov. 2018].

[12] Jupyter.org. (2018). *Project Jupyter*. [online] Available at: http://jupyter.org/ [Accessed 18 Nov. 2018].

[13] Moffitt, C. (2018). *Guide to Encoding Categorical Values in Python - Practical Business Python*. [online] Pbpython.com. Available at: http://pbpython.com/categorical-encoding.html [Accessed 1 Dec. 2018].

[14] Insider Threat Detection with Face Recognition and KNN User Classification," in *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)* , 2017.

[15] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?," in *International conference on database theory*, 1999, pp. 217-235.

[16] J. K. Kruschke and T. Liddell, "Bayesian data analysis for newcomers," 2017.

[17] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*: Elsevier, 2011.

[18] Boulesteix, Anne□Laure, et al. "Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2.6 (2012): 493-507.

[19] Breiman, L. "Out-of-bag estimation, ftp. stat. berkeley. edu/pub/users/breiman." *OOBestimation. ps* 199.6 (1996).

[20] S. Robertson, "Evaluation in information retrieval," in *Lectures on information retrieval*, ed: Springer, 2000, pp. 81-92.

[21] S. P. Harter and C. A. Hert, "Evaluation of information retrieval systems: Approaches, issues, and methods," *Annual Review of Information Science and Technology (ARIST),* vol. 32, pp. 3-94, 1997.

APPENDIX

In the following line we present some of the results we reached which we were not able to show due to space limitation.

```
names of all terrorist groups that are in our data
['MANO-D' '23rd of September Communist League' 'Black Nationalists' ...
 'Fatoni Warriors' 'Minorities of Metropolitan Attacks'
 'Baloch Republican Party']
total number of groups=
3536
```

Fig. 5. Total number of terrorest groups in the dataset

Next, we will list the detailed results for Model 1, in Figs. 6–8.

	precision	recall	f1-score	support
0	0.63	0.39	0.49	5996
1	0.93	0.97	0.95	48512
avg / total	0.90	0.91	0.90	54508

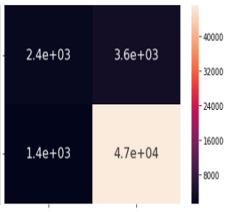

Fig. 6. KNN Results with the confusion matrix on the right

	precision	recall	f1-score	support
0	0.20	0.11	0.14	5989
1	0.90	0.95	0.92	48519
avg / total	0.82	0.86	0.84	54508

Fig. 7. NB Results with the confusion matrix on the right

	precision	recall	f1-score	support
0	0.66	0.50	0.57	6057
1	0.94	0.97	0.95	48451
avg / total	0.91	0.92	0.91	54508

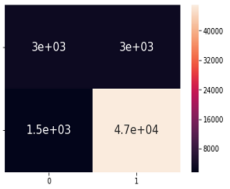

Fig. 8. RF Results with the confusion matrix on the right

	precision	recall	f1-score	support
0	0.86	0.92	0.89	277
1	0.27	0.49	0.35	112
2	0.99	0.95	0.97	267
3	0.96	0.77	0.85	202
4	0.44	0.15	0.23	156
5	0.93	0.98	0.95	63
6	0.59	0.13	0.21	248
7	1.00	0.88	0.94	160
8	0.12	0.16	0.14	55
9	0.26	0.60	0.36	60
10	0.89	0.97	0.93	455
11	1.00	0.98	0.99	335
12	0.29	0.24	0.26	63
13	0.90	0.98	0.94	163
14	0.56	0.78	0.65	58
15	0.00	0.00	0.00	71
16	1.00	0.93	0.97	89
17	0.21	0.94	0.35	83
18	0.00	0.00	0.00	55
19	0.50	0.32	0.39	231
20	0.32	0.21	0.26	52
21	1.00	0.99	1.00	747
22	0.85	0.88	0.87	51
23	0.76	0.92	0.83	63
24	0.78	0.77	0.78	102
25	0.74	0.98	0.84	187
26	0.29	0.71	0.41	106
27	0.84	0.98	0.90	135
28	0.71	0.24	0.36	328
29	0.58	0.80	0.67	241
30	0.91	0.77	0.83	561
31	1.00	1.00	1.00	62
avg / total	0.77	0.74	0.73	5838

Fig. 9. NB results on final fold

	precision	recall	f1-score	support
0	0.85	0.98	0.91	277
1	0.97	0.94	0.95	112
2	0.99	0.99	0.99	267
3	0.98	0.89	0.94	202
4	0.25	0.17	0.20	156
5	0.97	1.00	0.98	63
6	0.56	0.79	0.66	248
7	0.88	0.97	0.92	160
8	0.56	0.09	0.16	55
9	0.98	0.78	0.87	60
10	0.89	0.88	0.89	455
11	1.00	0.99	1.00	335
12	0.82	0.59	0.69	63
13	0.89	0.99	0.94	163
14	0.90	0.62	0.73	58
15	0.80	0.66	0.72	71
16	0.98	0.99	0.98	89
17	1.00	0.96	0.98	83
18	0.13	0.11	0.12	55
19	0.96	0.94	0.95	231
20	0.57	0.31	0.40	52
21	0.99	1.00	1.00	747
22	0.87	0.90	0.88	51
23	0.82	0.92	0.87	63
24	0.86	0.95	0.90	102
25	0.88	0.94	0.91	187
26	0.94	0.87	0.90	106
27	0.95	0.96	0.96	135
28	0.96	1.00	0.98	328
29	0.97	1.00	0.98	241
30	0.96	0.95	0.95	561
31	1.00	1.00	1.00	62
avg / total	0.89	0.90	0.89	5838

Fig. 10. KNN results on final fold

For Model 2, in Figs. 9–11, we list the results on the final fold only, because it produced one for the ten folds for the three classifiers. In addition, the confusion matrices are of the size 32x32, which is too large to be displayed legibly, so we have not included that here.

	precision	recall	f1-score	support
0	0.86	0.98	0.92	277
1	0.93	0.79	0.86	112
2	0.98	0.99	0.98	267
3	0.99	0.98	0.98	202
4	0.38	0.31	0.34	156
5	0.97	1.00	0.98	63
6	0.56	0.71	0.63	248
7	0.97	0.97	0.97	160
8	0.14	0.05	0.08	55
9	1.00	0.77	0.87	60
10	0.90	0.98	0.94	455
11	1.00	0.99	1.00	335
12	0.79	0.67	0.72	63
13	0.91	0.99	0.95	163
14	0.86	0.62	0.72	58
15	0.72	0.62	0.67	71
16	0.98	0.99	0.98	89
17	0.97	1.00	0.98	83
18	0.55	0.11	0.18	55
19	0.97	0.93	0.95	231
20	0.75	0.29	0.42	52
21	0.99	1.00	0.99	747
22	0.94	0.90	0.92	51
23	0.84	0.94	0.89	63
24	0.86	0.98	0.92	102
25	0.87	0.90	0.88	187
26	0.96	0.88	0.92	106
27	0.98	0.93	0.96	135
28	0.96	0.99	0.97	328
29	0.95	0.99	0.97	241
30	0.95	0.98	0.96	561
31	1.00	1.00	1.00	62
avg / total	0.90	0.91	0.90	5838

Fig. 11. RF results on final fold