Visual analysis and empirical studies on Global Terrorism Dataset

Syeda Azim, Himanshu Aggarwal, Ishank Jain
*snjazim, hhimansh, ishank.jain (@uwaterloo.ca)

***Abstract*—One of the leading security concerns faced by the international community is the threat of global terror. To perform information extraction related to attacks by known organization we have analyzed the terrorist groups by applying natural language processing(NLP). We predict number of casualties, following major attacks in the most targeted countries, we have performed empirical studies using various Regression algorithms.**

***Index Terms*—Knowledge discovery, Machine learning, Mean square error methods, Multilayer perceptrons, Prediction methods, Regression algorithms, Supervised learning, Text mining.**

I. Introduction

For the project, we have worked on START Global Terrorism Dataset (GTD) by the university of Maryland [1]. Although, the dataset is open-source not much evaluation has been performed on it in the past. Terror attacks are an important event to analyze, it is very important to evaluate how lethal a terrorist attack can be when a particular region is targeted by a certain terror group.
We target a regression problem in our experiments for the project. The aim is to be able to predict number of casualties in a terrorist attack as precisely as possible. To address the problem we are using four different regression algorithms and then comparing their performance based on mean square error.

Section 2 includes a review of the relevant literature on some of the regression algorithms that we have employed in our experiment. Section 3 describes the Global Terrorism Database by The National Consortium for the Study of Terrorism and Responses to Terrorism (START), our main dataset for the project. Section 4 describes our methods for evaluating the severity of the attack and also presents the algorithms used to predict the number of casualties. In Section 5, we conclude our reports with reflections and suggestions for future academic work, we want to develop methods for predicting future terrorist attacks with two approaches.

A. Part 1

The first part of the project deals with Exploratory Data Analysis (EDA) to familiarize with the dataset in order to perform the pre-processing, data cleaning, and to further address the regression problem. With all of the time-series information and geographical information available, the best way to understand this data was by using visualizations. We have used HTML code to create a Tableau dashboard Fig.1 for interactive visualization of the dataset. The dashboard shows the terrorist activities across the globe for each year.

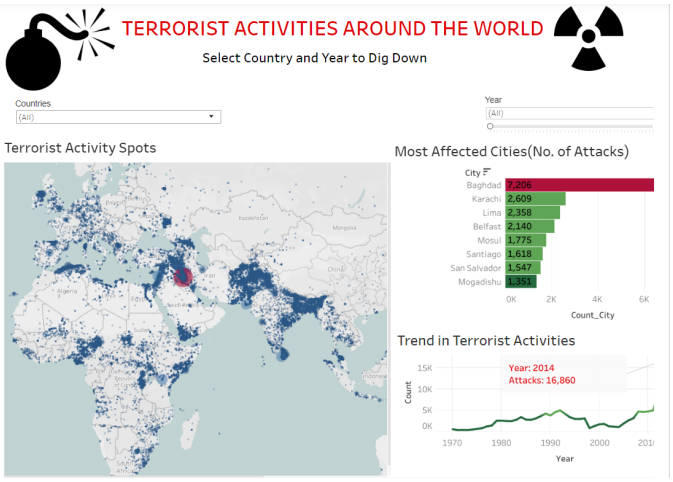

Fig. 1. Number of attacks by terrorist activities over the years

Additionally, the dashboard shows most targeted cities and the trend of terrorist activity for the selected year.

B. Part 2

Once we have understanding of how the data is distributed with the help of visuals, we begin to prepare our dataset for the regression problem. In this part we learn about the pre-processing methods and cleaning tools used to assemble the data.
For pre-processing we have replaced the missing values with most frequent values in the column. Additionally, if an attribute's variance in orders of magnitude is more than the variance of other features, then that particular attribute might dominate other features in the dataset. To avoid this we have used standardScaler library to scale the features to a range which is centered around zero. Next, we converted the categorical data to numeric data, in order to carry out machine learning algorithms.

C. Part 3

In the last part we carried out experiments using four regression algorithm: a) Multi-Level Perceptron b) Lasso Regression c) Ridge Regression and d) Random Forest. We have also provided an equation to calculate the severity which is the total number of deaths and total number of people wounded in terrorist event. Based on severity of the events we have calculated the lethality of the terrorist group. Further, we have shared the analysis of the regression results.

II. Literature Review

In this section, we discuss related works and each part with some details.

Natural Language Processing(NLP) is a tract of Artificial Intelligence and Linguistics, devoted to make computers understand the statements or words written in human languages. [2] Symbols are combined and used for conveying information or broadcasting the information. Natural Language Processing can be classified into two parts i.e. Natural Language Understanding (NLU) and Natural Language Generation (NLG) which evolves the task to understand and generate the text.
Our project focuses on Natural Language Understanding, more specifically, Information Extraction which is concerned with identifying phrases of interest of textual data. For many applications, extracting entities such as names, places, events, dates, times and prices is a powerful way to summarize the information relevant to a user's needs. These extracted text segments are used to allow search over specific fields and to provide effective presentation of search results and to match references to papers. Discovery of knowledge is becoming important areas of research over the recent years.
Knowledge discovery [3] research use a variety of techniques in order to extract useful information from source documents like removing stopwords, tokenization, and stemming. This extracted information can be applied on a variety of purpose, for example to prepare a summary, to build databases, identify keywords, classifying text items according to some pre-defined categories etc.

Regression trees (a.k.a. decision trees) learn hierarchically by repeatedly splitting data sets into separate branches that maximize each split's information gain. Information gain is the concept which refers to how much of a previously unseen instance would need to be known to properly classify a single instance. This branching structure enables regression trees to learn non-linear relationships naturally. Random Forest is an ensemble of regression tree that combines the mean predictions from many different tree.[4] Random Forest is great for learning complex, highly nonlinear relationships. Usually it can achieve quite high performance as it can estimate missing data and keep the accuracy when a size able percentage of data is missing. It may be prone to major over-fitting due to the nature of training decision trees. A completed tree model of decision can be too complex and contain unnecessary structure. Although this can sometimes be alleviated with proper pruning of the tree and larger groups of random forests. Consequently, large random forest ensembles makes the process slower and require more memory.[4]

Regularization is a technique used to penalize large coefficients to avoid over-fitting, and the strength of penalty should be adjusted.[5]

Lasso Regression is used to select features by adding an additional term to the Linear Regression loss function. Apart from avoiding over-fitting this also reduces the coefficient of less important features to zero by performing L1 regularization that adds penalty equivalent to absolute of the magnitude of coefficients. [6]

$$L = \sum(\hat{Y}_i - Y_i)^2 + \lambda \sum |\beta| \tag{1}$$

In (1), L is Loss function, λ is Penalty term, β is Coefficient. Lasso sets the irrelevant coefficient value to zero, therefore can perform feature selection by eliminating the imbalanced features.

Ridge regression also follows regularization technique, it performs L1 regularization that add penalty equivalent to square of the magnitude of coefficient. Ridge enforces the coefficient to be lower, but the coefficient is never minimized to zero. [6]

$$L = \sum(\hat{Y}_i - Y_i)^2 + \lambda \sum \beta^2 \tag{2}$$

In (2), L is Loss function, λ is Penalty term, β is Coefficient.

In any regularization technique when the penalty term is exactly zero, the regularization technique performs same as the linear regression. Again, when the penalty term is infinite, the coefficient will be zero. Therefore, the magnitude of penalty term will determine the weighting given to the various parts of the target.[7]

The MLP came into use in the mid 1980s [8] with the development of the back propagation learning algorithm by several independent researchers in the field and it has been used extensively in many kind of problems.

MLPs are useful in research for their ability to solve stochastic problems, which often allows approximate solutions for extremely complex problems. MLPs were a popular machine learning solution in the 1980s, finding applications in diverse fields such as speech recognition, image recognition, and machine translation software, but thereafter faced strong competition from support vector machines. Interest in back propagation networks returned due to the successes of deep learning.

Multilayer perceptron are often applied to supervised learning problems [9], they train on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs.

Generalization and fault tolerance are the main advantages of multilayer perceptron. The multilayer perceptron with back-propagation has been applied in numerous applications ranging from OCR (Optical Character Recognition) to medicine.

Large number of iterations are required for learning, so sometimes learning is expensive. Sometimes we may face scaling problem while implementing multilayer perceptron.

III. Global Terrorism Dataset

Global Terrorism Database(GTD) [1] is an open-source database including information on terrorist events around the world from 1970 through 2017.
The database is maintained by the National Consortium for the Study of Terrorism and Responses to Terrorism (START) at the University of Maryland, College Park in the United States. It contains information on approximately 182,000 terrorist events, and a total of 135 attributes for each event, including exact date, location, group, weapon, casualty, summary of the

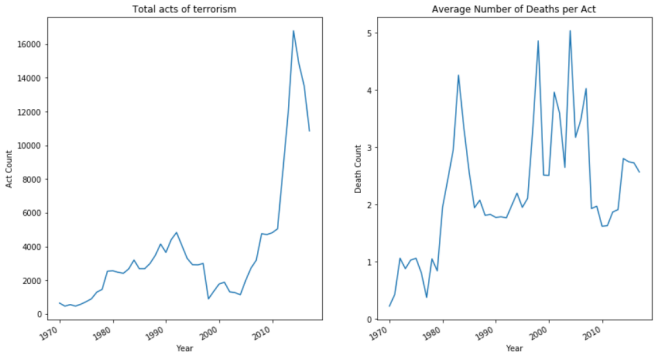

Fig. 2. Number of attacks by terrorist organizations based on regions and attack types

Fig. 3. Most frequent words used in the motive and summary following a terror event in 2015

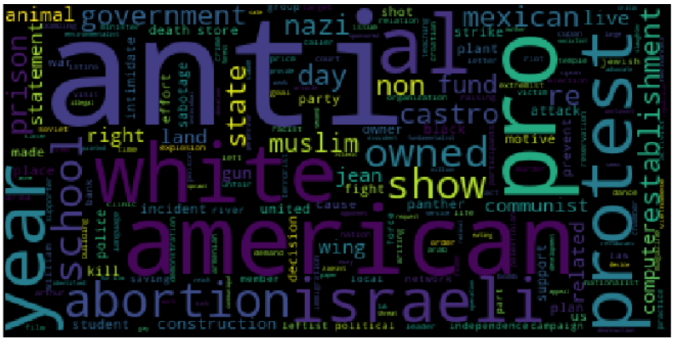

Fig. 4. Most frequent words used in the motive and summary following a terror event in United states

attack, motive and so on. It is currently the most comprehensive unclassified database on terrorist attacks in the world which includes information on more than 88,000 bombings, 19,000 assassinations, and 11,000 kidnappings since 1970. The data set is updated annually and is highly precise. It has data from over 4,000,000 news articles and 25,000 news sources for the incidents that occurred from 1998 to 2017 alone.

IV. Methods, Algorithms and Evaluation

A. *Information Extraction*

We have used two python libraries: a) NLTK [10] and b) Wordcloud to extract the most frequently used words during a terrorist event. Below we have provided the steps performed to achieve the results. We have extracted summaries from huge chunks of texts. We have used motive and summary column from the dataset as the source to extract important information. *Sentence Tokenization:* With the help of nltk tokenizer we have divided a string of written language into its component sentences which is then followed by *Word Tokenization* which further divides a string of written language into its component words. Text Lemmatization helps us to remove stem words such as dog, dogs, dogs are converted to dog. *Stopwords* are irrelevant words (for example and, the, a) that are necessary to remove to achieve quality result. We get a set of most important words after performing the above steps.

We further performed frequency distribution on the list of tokens to know which word has the highest weight amongst the list. Fig.3 and Fig.4 represent the results of information extraction. The larger font represent that the word was repeated more frequently than others. In Fig.4 'anti' has the largest font size which reflects that it is present in majority of the text published during the event. This is followed by 'american', 'white' which are present in fewer frequency and so on.

We will define a method to calculate the severity [11] of an event. The base of this severity is total number of deaths and total number of people wounded.

$$Severity = \alpha * Ndeath + Nwounded \tag{3}$$

Where Ndeath is number of people died and Nwounded is number of people wounded. α is a parameter indicating how many wounded people are equivalent to dead people in terms of severity. Here we have used $\alpha = 3$ for our analysis. We have defined the lethality of a terrorist group to be the sum of the severity of its all events:

$$Lethality = \Sigma Severity \tag{4}$$

The Fig.5 shows the total number of causalities report of top 10 terrorist groups based on the total number of people killed and people wounded. Lethatlity report based on these values, calculated by the above given formula is shown in Fig.6. We have selected top 10 terrorist groups for the visualization of our results based on number of causalities(Killed + Wounded). Causalities would be our target in this dataset as we will try to predict number of causalities based on different features/variables using different regression algorithms. We will calculate mean square error for these regression algorithms and see which regression algorithm shows us the best prediction.

If we compare the results in both of these figures, we'll see that lethality increases with increase in number of causalities. More the number of people killed, more is the lethality as it is multiplied by the constant factor of alpha. Thus, even if there are less number of wounded people in an event but more number of people killed, lethality would still be high. So for terrorist groups who have done more damage are more lethal.

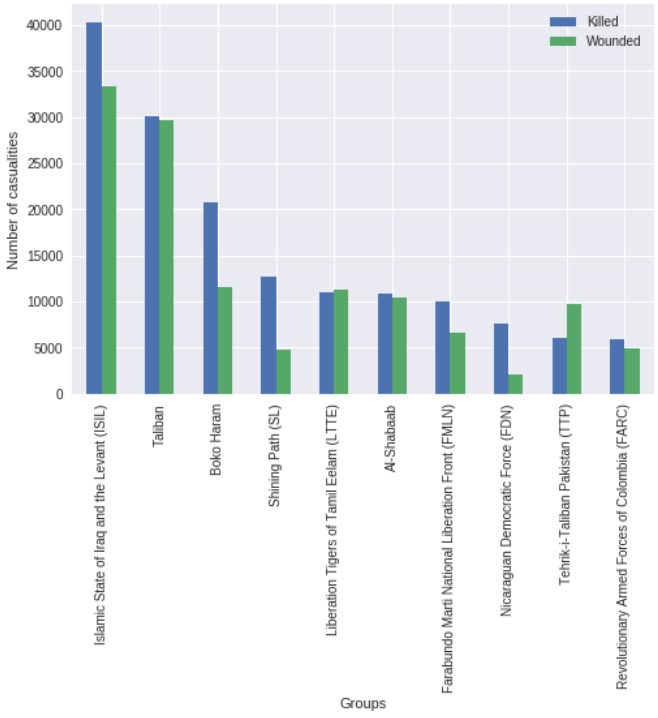

Fig. 5. Number of causalities based on terrorist groups

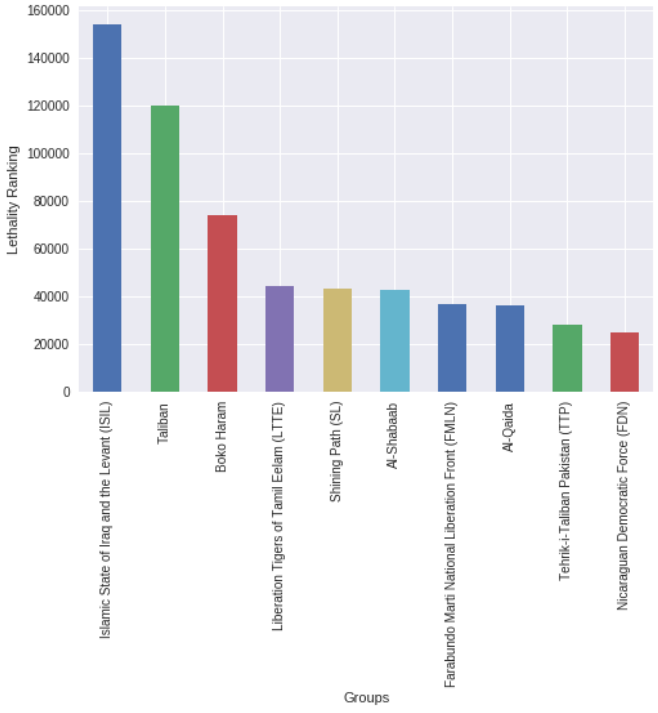

Fig. 6. Lethality report based on terrorist groups

For simplicity we have sorted the list of lithality and number of causalities.

From next part onward, we'll talk about various regression algorithms that we have used in our project. For our project we have used top 15 countries with most terrorist attacks as shown in Fig.7. This would contain approximately ˜117K data samples.

The performance of various Machine Learning algorithms depends heavily on data size and structure. Thus, the right algorithm choice often remains unclear unless we directly test our algorithms through plain old trial and error.

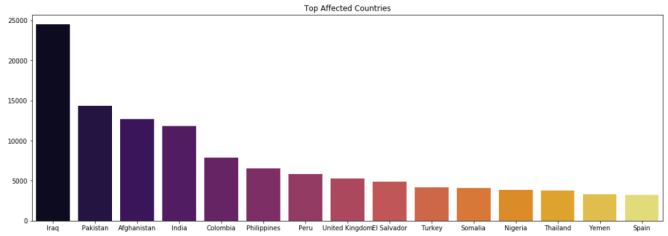

Fig. 7. Top 15 countries with most terrorist attacks

B. *MultiLayer Percetron*

Now moving on to the MultiLayer perceptron, we have used MLP Regressor to calculate the best parameters that would given us most optimized results. Running MLP Regressor on the dataset gave us the best parameters as:
activation layer : 'tanh'
solver : 'sgd'

```
{'activation': 'tanh', 'alpha': 1e-05, 'hidden_layer_sizes': 49, 'solver': 'sgd'}
```

Fig. 8. Best parameters for MLP Regressor

We have ran these best parameters on different number of hidden layers and calculated the mean square error for every layer. We have obtained the following (Table 1) results for mean square error.

TABLE I
MEAN SQUARE ERROR FOR DIFFERENT NUMBER OF LAYERS

Number of layers	Mean Square Error
10	96.3080
20	76.8299
30	46.7994
40	48.3671
50	45.1303
60	40.4991

As we can see from the results, mean square error kept decreasing from hidden layer size of 10 to size 30. Then there is a slight increase in mean square error when hidden layer size is 40. This slight increase in mean square error may be observed because of the over-fitting. Then after that, mean square error kept decreasing and at the size of 60, it ended up at about 40.5. The percentage decrease in mean square error from hidden layer size of 10 to 60 is about 58%. We can also visualize the results of this mean square error using bar plot as shown in Fig.9.

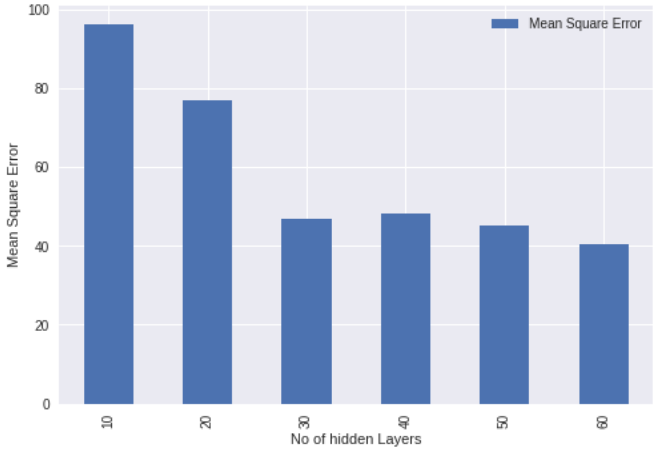

Fig. 9. Change in Mean Square Error vs Number of hidden layers

C. *Ridge and Lasso Regression*

To tune the λ value for both the regularization techniques, the model is trained with 70% of the data set for different λ values. The remaining 30% of the data set is used for testing. The algorithm is evaluated by calculating the Mean Square Error of the predicted values.

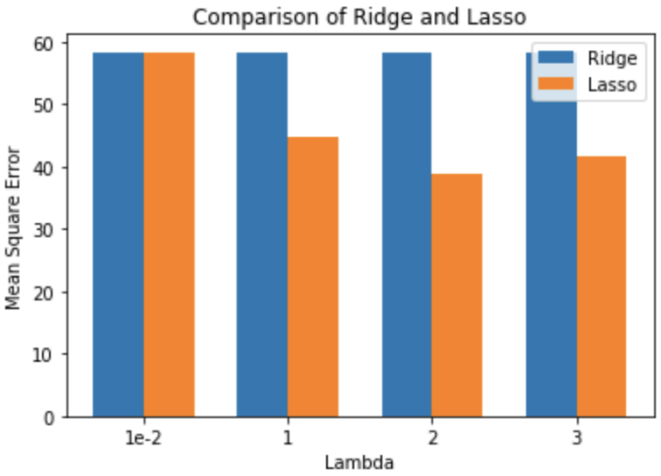

Fig. 10. Comparison of Ridge and Lasso

Fig.10 compares the mean square error of Ridge and Lasso regression algorithm for four different values of lambda. In the chart we see lasso has lower mean square error values than ridge for all four lambda values. We know lasso performs L1 regularization, which sets the irrelevant coefficient value to zero; where else ridge performs L2 regularization, which lowers the coefficient values but it does not minimize it to zero. Therefore, lasso is performing feature selection[12] by eliminating the feature that has less effect on predicting the target and thus show lower mean square error value than ridge for all the lambda values.

TABLE II
PARAMETERS FOR RIDGE AND LASSO

λ	MSE (Ridge)	MSE (Lasso)
0.01	58.3835	58.2775
1	58.3828	44.6677
2	58.3820	38.9745
3	58.3813	41.6087

D. *Random Forest*

We trained the random forest regression algorithm on different number of trees ranging from 45 to 50. To evaluate the predictive analysis we calculated the mean square error for each number of tree. Fig.11 shows that the mean square error value is lowest when the parameter n_estimater, which is the number of tree, is 46. As the number of tree is raised higher than 46 the mean square error value increase because random forest model over-fits the data and includes the data that are noisy.[4] By noisy we mean the data points that do not really represent the true properties. As a result, the predictive performance of the model is reduced and mean square error value is raised.

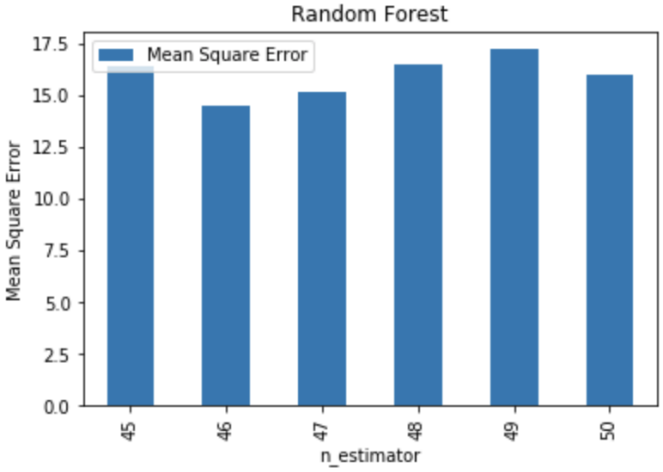

Fig. 11. Performance of Random Forest for different n_estimator

TABLE III
PARAMETERS FOR RANDOM FOREST

n_estimator	MSE
45	16.4279
46	14.5167
47	15.1927
48	16.4945
49	17.2176
50	19.9918

V. CONCLUSION

In this paper we first performed EDA to understand the data distribution. Following this we performed information extraction with the help of Natural Language Toolkit and

WordCloud libraries in python. This allowed us to know the most frequent words used surrounding a terror event.

We then provided an equation to check severity of the attacks and ended our experiments by addressing a regression problem to predict the number of casualties. We have compared the mean square values obtained by running our dataset on different Regression Algorithms. From Table 4, we can see that Random Forest (MSE 14.51) gave us the best results as compared to other Regression Algorithms. Random forest shows such high performance by reducing the weighted impurity in a tree. For regression the impurity is measured by feature variance. Therefore, when training a tree, it can be measured how much each feature reduces the weighted impurity in a tree. For a forest, it is possible to average the decline in impurity from each feature and rank the features according to this measure, and finally Random forest gives the best predictive performance by decreasing impurity. On the other hand, Ridge regression do not remove the impure data, thus learns noisy data points and gives the maximum mean square error on prediction. Lasso regression performs feature selection by arbitrarily selecting any one feature among the highly correlated ones and reduced the coefficients of the rest to zero. The selected variable also changes randomly as the model parameters changes, thus predictive performance of Lasso regression totally depend on the tuning of the penalty term. Performance of MLP is unpredictable and it varies for different dataset. For our dataset MLP shows a lower predictive performance than Lasso regression. Finally, it is clear that Random Forest is the best regression algorithm to perform predictive analysis for our dataset.

TABLE IV
MEAN SQUARE ERROR FOR DIFFERENT REGRESSION ALGORITHMS

Algorithms	MSE
MLP	40.49
Ridge	58.38
Lasso	38.97
Random Forest	14.51

VI. FUTURE WORK

For the future work we want to combine natural language processing and lethality score along with activity of terror group in the region to predict the likeliness of a terror attack in the region.

VII. ACKNOWLEDGMENT

We would like to thank Iman Fadakar and Professor Haitham Amar for their guidance and helpful comments throughout the length of project.

VIII. APPENDIX

In this section we have few additional libraries used for the project and some additional visuals.

A. The Matplotlib Basemap Toolkit

This toolkit provides PROJ.4 C library for the functionality to transform coordinates to one of the 25 different map projections. The GEOS library in the toolkit is used internally to clip the coastline and polticial boundary features to the desired map projection region.

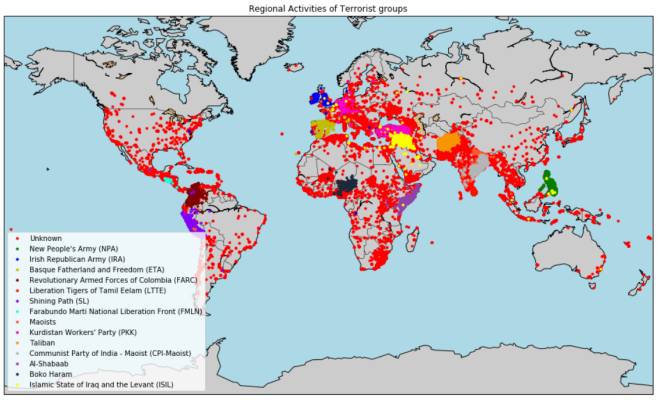

Fig. 12. Terror activities by different groups

B. Folium

Folium is a Python Library that allows to visualize spatial data in an interactive manner, straight within the notebooks environment. The library is highly intuitive to use, and it offers tools to creeate interactive leaflet maps.

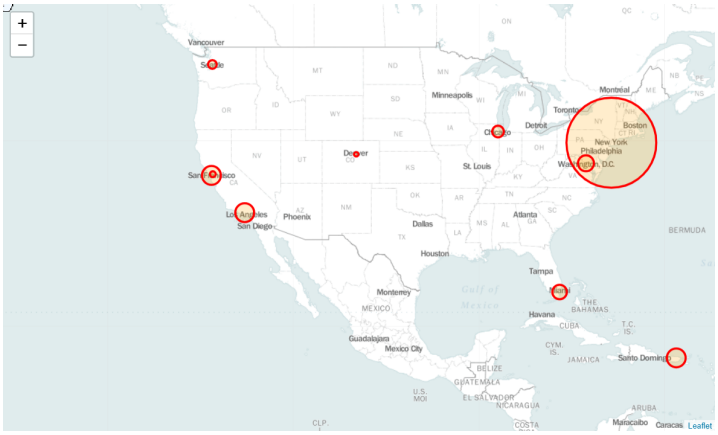

Fig. 13. Top ten targeted cities in United States

C. Wordcloud

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites. For generating word cloud in Python, modules needed are matplotlib, pandas and wordcloud.

We also referred to Natural Language Processing with Python. Written by Steven Bird, Ewan Klein and Edward Loper [13].

REFERENCES

[1] "National consortium for the study of terrorism and responses to terrorism (start) global terrorism database [data file]," 2018. [Online]. Available: Retrieved from https://www.start.umd.edu/gtd

[2] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *CoRR*, vol. abs/1708.05148, 2017. [Online]. Available: http://arxiv.org/abs/1708.05148

[3] S. Singh, "Natural language processing for information extraction," *CoRR*, vol. abs/1807.02383, 2018. [Online]. Available: http://arxiv.org/abs/1807.02383

[4] "Selecting the best machine learning algorithm for your regression problem," 2018. [Online]. Available: Retrieved from https://towardsdatascience.com/selecting-the-best-machine-learning-algorithm-for-your-regression-problem-20c330bad4ef

[5] "Modern machine learning algorithms: Strengths and weaknesses," 2019. [Online]. Available: Retrieved from https://elitedatascience.com/machine-learning-algorithmsregression

[6] "A complete tutorial on ridge and lasso regression in python," 2019. [Online]. Available: Retrieved from https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-ridge-lasso-regression-python

[7] "Practical machine learning: Ridge regression vs. lasso," 2017. [Online]. Available: Retrieved from https://codingstartups.com/practical-machine-learning-ridge-regression-vs-lasso/

[8] M. L. Vaughn, "Interpretation and knowledge discovery from the multilayer perceptron network: Opening the black box," *Neural Computing and Apllication*, vol. 4, pp. 72–82, 1996.

[9] "A beginner's guide to multilayer perceptrons (mlp)." [Online]. Available: Retrieved from https://skymind.ai/wiki/multilayer-perceptron

[10] E. Loper and S. Bird, "NLTK: the natural language toolkit," *CoRR*, vol. cs.CL/0205028, 2002. [Online]. Available: http://arxiv.org/abs/cs.CL/0205028

[11] J. Alison, L. Deng, and Z. B. Zhu, "Cs224w final project report: Uncovering the global terrorism network," 2017.

[12] "Why, how and when to apply feature selection," 2018. [Online]. Available: Retrieved from https://towardsdatascience.com/why-how-and-when-to-apply-feature-selection-e9c69adfabf2

[13] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, 1st ed. O'Reilly Media, Inc., 2009.