TerrorMine: Automatically Identifying the Group behind a Terrorist Attack

Alan Falzon and Joel Azzopardi
Department of Artificial Intelligence, Faculty of ICT, University of Malta, Msida, Malta

Keywords: Terrorism, Data-mining, GTD, Machine Learning, Prediction, Clustering, Forecasting.

Abstract: Terrorism is a problem that provokes fear and causes death internationally. The Global Terrorism Database (GTD) contains a large number of terrorist attack records which can be used for data mining to help counter or mitigate future terror attacks. TerrorMine employs AI techniques to identify perpetrators responsible for terrorist attacks. Moreover, the effect of clustering beforehand is investigated, while also attempting to identify new (unknown) terrorist organisations, and predicting future activity of terror groups. Several experiments are performed. The Random Forest model obtains the highest Weighted F1-score when identifying responsible perpetrators. Furthermore, upon clustering the data using Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBScan) before classification, training time is reduced by more than 50%. Various techniques are used for the unsupervised identification of whether a terrorist attack was carried out by an unknown terrorist group. Nearest Neighbours gives the highest Macro F1-score when cross-validated. When forecasting the future impact of the different terrorist groups, Prophet achieved an F1-score higher than that of Autoregressive Integrated Moving Average (ARIMA).

1 INTRODUCTION

Terrorism refers to violent threats and acts aiming to spread fear among targets, and impose ideas onto a particular population (Jackson, 2018). It affects tourism and in turn the national general economy. For this reason, terrorism is one of the toughest challenges faced worldwide (Tolan and Soliman, 2015).

One of the most notable attacks of all time is the September 11 attack, more commonly known as '9/11 attacks', committed by the Al Qaeda group in the US in 2001. It caused almost 3000 deaths, spreading tremendous terror[1]. While the average frequency of terrorist attacks has slowly declined over time, the impact of terrorist activity has increased mostly due to mass media which reports terrorist activity and their widespread condescending behaviour in a dramatised fashion (Pfeiffer, 2012).

Data driven approaches can help counter-terrorism efforts whereby data about past terrorist attacks and related intelligence can be mined for knowledge that can help prevent future attacks. The Global Terrorism Database (GTD)[2] is an excellent source of such data and can be used to help train models that automatically identify perpetrators behind an attack or predict future terrorist activities.

Perpetrator identification is one of the initial counter-terrorism steps which lead to the prevention of future attacks (Talreja et al., 2017). Traditionally, the organisation responsible for a terrorist act either claims responsibility or is discovered through intelligence gathering activities such as email tracking, phone tracing or social network analysis (Talreja et al., 2017). Such methods are cumbersome, require specialised skill and dedicated hardware, and potentially infringe on privacy rights. This reveals the need for AI approaches to help in counter-terrorism work.

This paper aims to mine the GTD in order to discover knowledge that can help in counter-terrorism. Other than perpetrator identification, this work aims to potentially fill certain gaps such as the detection of a newly formed terrorist group. The ability to tackle such a problem would ultimately be useful to raise warnings about threats by terrorist organisations.

The overall aim of this work is to contribute to the efforts of counter-terrorism using a data-driven ap-

[1] https://abcnews.go.com/US/20th-anniversary-911-nears-questions-anger-death-linger/story?id=79606569 [Last Accessed: April 5th, 2022]

[2] https://www.start.umd.edu/gtd/ [Last Accessed: June, 2022]

Falzon, A. and Azzopardi, J.
TerrorMine: Automatically Identifying the Group behind a Terrorist Attack.
DOI: 10.5220/0011540300003335
In *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2022) - Volume 1: KDIR*, pages 221-228
ISBN: 978-989-758-614-9; ISSN: 2184-3228

proach. The aim was achieved through the following objectives:

- Automatically identifying the terrorist organisation responsible for a terrorist attack
- Investigating the effect of clustering on the terrorist group identification
- Automatically discovering new terrorist groups responsible for recent attacks
- Predicting the future activity of terrorist groups

2 LITERATURE REVIEW

This section consists of an overview of the various research carried out in relation to the objectives specified in Section 1. Section 2.1 provides an overview of the Global Terrorism Database (GTD) while Sections 2.2, 2.3, 2.4 and 2.5 provide a review of literature related to each of the four objectives, respectively.

2.1 The Global Terrorism Database (GTD)

The GTD, maintained by the National Consortium of the START, is an open-source database that, at the time of writing, contains data describing over 200,000 terrorist attacks that happened from 1970 until 2019. For each terrorist incident, there are up to 120 variables including the date, location, information about the target and weapons used, the number of casualties, and the responsible group if identified (National Consortium for the Study of Terrorism and Responses to Terrorism (START), 2019).

Data from the GTD needs to be prepared by selecting the subset of features considered to be important, handling missing data, scaling and encoding data (Pagán, 2010), (Laite et al., 2019). Feature selection involves reducing the number of attributes to be used by classification models, as done by Pan using Extremely Randomised Trees (Pan, 2021). Various techniques are explored when it comes to handling missing data, including Listwise Deletion and Mode, Mean, Median, KNN and Multivariate Imputation (Pagán, 2010).

2.2 Objective 1: Automatically Identifying the Perpetrator

This section provides a review of the literature related to the automated identification of the terrorist organisation responsible for an attack. It tackles the various encountered missing data techniques, feature selection methods and classification models.

Missing Data
The most frequently used technique to handle missing GTD data is Listwise Deletion (Tolan and Soliman, 2015), (Pagán, 2010), (Khorshid et al., 2015b), (Laite et al., 2019). (Tolan and Soliman, 2015) compare the performance of the Listwise Deletion technique to that of Mode Imputation, arriving to the conclusion that the former method performs better overall. (Pagán, 2010) compares Listwise Deletion to Mean Imputation, Median Imputation and KNN Imputation and reports that the best results were obtained using the latter method.

Feature Selection
In order to perform classification of the data found in GTD, the features to be fed to the classification pipeline are most frequently hand-picked manually (Pagán, 2010), (Laite et al., 2019), (Khorshid et al., 2015b). (Pan, 2021), (Peng, 2018) and (Talreja et al., 2017) use automated approaches involving Extremely Randomised Trees, a feature correlation matrix and Factor Analysis of Mixed Data, respectively.

Classification
The most frequent ML classifiers used on terrorist attack data from GTD encountered in the performed research include Decision Tree, Logistic Regression, KNN, Naive Bayes and SVM. (Talreja et al., 2017), (Mohammed and Karabatak, 2018), (Tolan and Soliman, 2015) and (Khorshid et al., 2015a) all implement and compare both the Decision Tree and SVM classifiers. (Talreja et al., 2017) report that upon evaluating their different implemented models trained on GTD data ranging from 1990 until 2014 and tested on attacks that happened during 2015 where the location is India, SVM resulted in the highest accuracy. (Peng, 2018) and (Diab, 2019) also implement the Decision Tree and SVM ML models, respectively.

(Peng, 2018), (Tolan and Soliman, 2015) and (Khorshid et al., 2015a) perform classification using KNN and Naive Bayes, with the former classification technique also being utilised by (Pagán, 2010) and (Mohammed and Karabatak, 2018). (Tolan and Soliman, 2015) report that the KNN classifier resulted in the highest accuracy score. (Diab, 2019) applies the Logistic Regression, SVM and Perceptron classifiers for classification and they use them to compare the performance of Gradient Descent (GD) and Stochastic Gradient Descent (SGD) on the classification of terrorist attacks with respect to the type of attack based on textual data. They report that the best

result was obtained when using the Perceptron classifier.

2.3 Objective 2: Exploring the Effect of Clustering

As observed in the undertaken research, clustering is used in various ways to improve the classification process. These include: the removal of outliers detected through clustering, as done by (Meng et al., 2017); fitting a classifier on each cluster during the classification phase, as performed by (Mathivanan et al., 2018); and performing clustering to generate the cluster ID of each record as an additional feature to be fed to the classifier, as implemented by (Alapati and Sindhu, 2016).

(Meng et al., 2017) use clustering as an unsupervised approach to detect outliers in GDT data. The first approach is to consider all instances of a cluster which is smaller than a specific threshold as outliers. The second approach is to consider those instances whose distance to their respective centroid is above a threshold as outliers. The latter approach resulted in a better classification performance.

Focusing on the e-commerce domain, (Mathivanan et al., 2018) compare K-Means and Hierarchical clustering as means to improve classification. They report that while classification performance was not improved, the classification process takes much less time when preceded by Hierarchical clustering. (Alapati and Sindhu, 2016) investigate the same two clustering methods where the cluster ID resulting from the pre-process is input to the ANN classifier as an additional field. As opposed to the experiments performed by (Mathivanan et al., 2018), their resulting classification accuracy is improved when preceded by clustering, with the highest increase resulting from hierarchical clustering before classification through ANN (Alapati and Sindhu, 2016).

2.4 Objective 3: Automatically Discovering New Perpetrators

Novelty detection is commonly treated as an outlier detection problem where outliers are considered to be only present in new data (Pedregosa et al., 2011). To date, we have not encountered systems that apply these techiques to the terrorism domain.

The most frequently encountered novelty and outlier detection techniques include One-Class Support Vector Machine (OCSVM), which is implemented by (Spinosa and Carvalho, 2005); Autoencoder, which is used by (Curia, 2020); Generative Adversarial Network (GAN), as utilised by (Zenati et al., 2018) and (Carrara et al., 2021); and Local Outlier Factor (LOF) – used by (Alsawadi and Bilal, 2021).

(Spinosa and Carvalho, 2005) use OCSVM to perform novelty detection on medical data, where a data point which falls outside the space enclosed by the hyperplane is considered to belong to an unseen class. (Curia, 2020) performs outlier detection on terrorist attacks from GTD as a means to predict whether an attack is successful or not using a hybrid approach consisting of an Autoencoder model and K-Modes clustering. Autoencoder is a specific type of ANN which encodes an input instance and decodes it back to its original form. If the loss between the two instances is greater than a threshold specified for each cluster of data resulting from K-Modes clustering, then the instance is considered an outlier, thus a successful terrorist attack (Curia, 2020).

(Zenati et al., 2018) and (Carrara et al., 2021) combine an Autoencoder with a GAN to perform anomaly detection. GANs are specialised types of ANNs made up of a Generator and a Discriminator. The Generator is responsible of generating fake data, while the Discriminator is responsible of determining whether data is real or generated when fed both types of data. When the GAN is used as an outlier detector, the Generator component is an Autoencoder whose input instance is considered an outlier if the Discriminator tags the output as having been generated.

LOF is an unsupervised technique applicable to outlier and novelty detection. (Alsawadi and Bilal, 2021) utilise LOF for safer navigation of autonomous vehicles by enabling it to detect new potentially dangerous situations during autonomous driving. The LOF method considers any data points as outliers if they are located in low-density areas in relation to their neighbours (Alsawadi and Bilal, 2021).

2.5 Objective 4: Predicting the Future Activity of Terrorist Groups

When it comes to the forecasting of future terrorist activity, the most frequently encountered technique is the ARIMA forecasting model, which is used by both (Sahin, 2018) and (Li et al., 2017). While not specifically used in the counter-terrorism field, Prophet, open sourced by Facebook in 2017, has also been gaining traction when it comes to forecasting (Taylor and Letham, 2017). Prophet is essentially an additive regression model which relies on the growth, seasonality, effect of any specified holidays and any custom white noise error.

(Sahin, 2018) proposes a system used to forecast the frequency of terrorist attacks taking place in different countries for a given month, utilising data

from the GTD. They compare the performance of the ARIMA and the Dynamic Regression models. The ARIMA method solely uses a linear combination of past terrorist attacks. On the other hand, the Dynamic Regression Model not only depends on past values of the variable being predicted, but also on other factors, such as the Gross Domestic Product (GDP) (Sahin, 2018). The Dynamic Regression model did perform better than ARIMA in certain cases, however the ARIMA model performed better overall. (Li et al., 2017) also make use of the ARIMA model based on GTD data to predict future terrorist activity occurring in different countries, aiming to predict the conditional probability of bombing tactics in different countries. (Li et al., 2017).

(Battineni et al., 2020) use Prophet to perform 60-day forecasting of the total number of COVID-19 infections in the USA, Brazil, India and Russia, using a logistic growth model. On the other hand, (Jha and Pande, 2021) compare the ARIMA and Prophet models by forecasting Supermarket sales data, ultimately concluding that Prophet performs better, resulting in a better fit, more accurate predictions and a smaller error rate.

3 METHODOLOGY

The methodology of TerrorMine consists of the conduction of various experiments to identify the terrorists behind an attack, explore the effect of clustering on the identification, detect the emergence of new groups and forecast terrorist group activity. The experiments, performed in line with the objectives, are respectively described in Sections 3.1, 3.2, 3.3 and 3.4.

3.1 Objective 1: Automatically Identifying the Perpetrator

For the automated identification of the responsible terrorist groups, the initial step is to select the most salient features. This is done through extremely randomised trees using the ExtraTreesClassifier model provided by SKLearn [3]. During the fitting of the classifier, the feature importance is computed for each of the tables based on the impurity score for the splitting of data based on the features. After fitting, these impurity scores for all the input features are extracted from the trained model. All features with an impurity score higher than average are then selected. The selected features are *iyear*, *imonth*, *country*, *region*, *provstate*, *city*, *latitude*, *longitude*, *targsubtype1*, *corp1*, *target1*, *natlty1* and *weapsubtype1*.

The classifiers implemented for this objective are Decision Tree, Random Forest, KNN, Logistic Regression, SVM (also trained using SGD) and Feed-forward Neural Network (FNN), which are some of the best encountered performing classification techniques in this domain. For the Decision Tree and Random Forest classifiers, the categorical features are ordinal encoded and the continuous features are left as is, since high non-standardised values do not impact tree-based classifications. On the other hand, for the remaining classification techniques, categorical features are one-hot encoded and continuous features are standardised. All classifiers, excluding ANNs, are fine-tuned through a grid search using a wide variety of parameters.

After selecting the optimal classification model, two additional experiments are carried out in attempt to enhance the performance of the model which include handling of missing data and the use of textual features. For the former, Listwise Deletion, Mode Imputation and Multivariate Imputation are performed before classification. For the inclusion of textual data we compute the structured TF-IDF features of the attack *summary* field in GTD. Such improvements are implemented based on their positive impact in the experiments encountered in the reviewed literature.

3.2 Objective 2: Exploring the Effect of Clustering

Two different approaches are taken to explore the effect of clustering on classification. The first approach is to use the assigned cluster label as an additional feature to the classification pipeline. The second approach is to cluster the data and then fit a separate classification model on each cluster. The second approach involves performing hyper-parameter tuning on each model so that each model is optimised to classify instances belonging to its respective cluster. Clustering experiments are carried out using K-Means, K-Modes and HDBScan.

K-Means operates using Euclidean distance, thus continuous features are standardised and categorical features are one-hot encoded. Since K-Means does not perform well with high dimensional data, possibly resulting in meaningless clusters, the four textual features that mostly contribute to data sparsity are excluded, namely: *provstate*, *city*, *corp1*, *target1*. The K-Means and HDBScan techniques both operate on the same feature subset. K is set to 6 for K-Means,

[3] https://scikit-learn.org/ [Last Accessed: May 13th, 2022]

determined through the Elbow method using Distortion score. HDBScan, on the other hand, automatically infers the number of clusters due to its density-based mechanism. K-Modes is set up to only operate on the categorical feature subset since it is a technique used for clustering categorical attributes. A K of 11 is determined by considering the least cost, which is the sum of dissimilarities between all of the resulting clusters.

3.3 Objective 3: Automatically Discovering New Perpetrators

The problem of detecting new perpetrators is treated as an outlier detection problem where outliers are only assumed to be present in unseen data. The experiments for this objective use unsupervised Nearest Neighbours, LOF, OCSVM, Autoencoder and GAN for outlier detection, which were found to be some of the most effective techniques in the related work. The same feature subset described in Section 3.1 is considered, except for the Autoencoder and GAN, which exclude the highly dimensional attributes that cause the DNN model training to exceed the available hardware's capacity. The Nearest Neighbours, LOF and OCSVM models are fine-tuned using Grid Search.

For the implementation of Nearest Neighbours, the model is fitted on the training data and for any new unseen instance, its Euclidean distance to the closest K neighbours is calculated. If the mean distance is greater than a specified threshold determined through Grid Search, then the instance is considered an outlier. When using Autoencoder, the decoded output of an instance is compared to the input using Mean Absolute Error (MAE). If the loss is larger than the standard deviation of the list of (MAE) values of the training data subtracted by their mean, then the instance is considered an outlier, as done by (Rajan, 2021). When it comes to the GAN technique, the Generator is replaced by an Autoencoder and once the Discriminator is trained, the network is fed the unseen data. Any instances which the Discriminator labels as "fake" are considered to be outliers, thus assumed to be attacks performed by unseen terrorist groups.

3.4 Objective 4: Predicting the Future Activity of Terrorists Groups

The impact of each terrorist group for a specific year and month is predicted using ARIMA and Prophet, where, according to the review literature, the former technique has been shown to perform well in this field and the latter has been proven to even surpass ARIMA in other fields. The impact is predicted by forecasting the number of civilian fatalities (*nkill* - *nkillter*), the number of those wounded (*nwound* - *nwoundte*) and then summing them up, as performed by (Singer and Golan, 2019). The impact is then categorised through quantile-based discretisation. As an additional experiment, the attack frequency is also forecast.

In order to find the optimal ARIMA model for each required measure for each terrorist group, the *auto_arima* behaviour of the arima class developed by (Smith et al., 2017) is used; a tool intended to discover the optimal parameters of this forecasting technique. This enables the discovery of the optimal value for the order of the first-differencing, auto-regressive and moving-average models. It then proceeds to fine-tune the values of the order of the auto-regressive and moving-average components of the seasonal model, and seasonal differencing. When evaluating how well a set of parameters results in forecasting, Mean Squared Error (MSE) is used. Similarly, the optimal Prophet model is estimated in terms of growth, upper-bound and seasonality pattern. For Prophet, in order to evaluate the performance of each set of parameters, the Root Mean Square Error (RMSE) is used.

4 RESULTS & DISCUSSION

This section presents the achieved results for each objective. For the first three objectives, data was ordered by *date* and *eventid* and split into 80% training and 20% testing sets. The training set is used to fine-tune and select the optimal model through time-independent cross-validation. The test set is used to test the selected model on unseen time-dependent data. For the fourth objective, data ranging from January 2010 until May 2016 is used as training data and data from June 2016 until December 2017 for validation data in order to fine-tune the forecasting models. The models are then tested on the unseen data which ranges from January 2018 until December 2019.

4.1 Objective 1: Automatically Identifying the Perpetrator

This section includes the results for perpetrator identification involving the Decision Tree, KNN, Logistic Regression, SVM and FNN classifiers, in addition to missing data handling techniques and the use of unstructured textual data. Table 1 shows that, based on the Weighted F1-score of 0.818, the optimal model for the automated identification of the perpetrator is the Random Forest model operating on both structured data and unstructured textual data from the *summary* field of terrorist attack records. Furthermore, it

also shows that handling missing data using Listwise Deletion, Mode Imputation and Multivariate Imputation does not improve the results.

The scores decreased when tested on the unseen test data, resulting in an F1-score of 0.800. This is expected since the model is fine-tuned on the training data. The decrease is slight however, which implies that the model is not over-fit on the training data. The table also shows that our model performs substantially better than a weighted random (naive) model. The model is also trained on GTD data ranging from 1990 until 2014 and tested on data for 2015 and 2016, to compare it with the classification results obtained by (Peng, 2018). The table shows that our model performed better, achieving an accuracy of 0.693.

Table 1: Table showing the perpetrator identification Accuracy and Weighted Precision, Recall and F1-score for the different models: Decision Tree (DT), KNN, Logistic Regression (LR), SVM, SVM using SGD (SVM-S), FNN, Random Forest (RF), Random Forest in addition to Listwise Deletion (RF-LD), Mode Imputation (RF-MI) and Multivariate Imputation (RF-MVI), Random Forest operating only on text features (RF-T) and Random Forest operating on both structured and text features (RF-ST).

	Accuracy	Precision	Recall	F1
Cross Validation Data (2015 - 2018)				
DT	0.718	0.795	0.718	0.737
KNN	0.797	0.771	0.797	0.777
LR	0.827	0.810	0.827	0.811
SVM	0.801	0.814	0.801	0.797
SVM-S	0.829	0.807	0.829	0.808
FNN	0.620	0.718	0.620	0.625
RF	**0.830**	0.811	**0.830**	0.813
RF-LD	0.808	0.791	0.808	0.788
RF-MI	0.829	0.813	0.829	0.812
RF-MVI	0.829	0.811	0.829	0.812
RF-T	0.696	0.651	0.696	0.663
RF-ST	0.828	**0.828**	0.828	**0.818**
Test Data (2018 - 2019)				
Random	0.075	0.101	0.075	0.082
RF-ST	**0.817**	**0.817**	**0.817**	**0.800**
Test Data (2015 - 2016)				
Random	0.072	-	-	-
RF-ST	**0.693**	-	-	-
Peng (2018)	0.583	-	-	-

4.2 Objective 2: Exploring the Effect of Clustering

This section includes the clustering performance results for the K-Means, K-Modes and HDBScan techniques. Table 2 shows that the hybrid approach involving the fine-tuning and fitting of a Random Forest model on each cluster produced by the HDBScan technique, resulted in the best Weighted F1-score of 0.830 during cross-validation, surpassing the standard Random Forest model. When tested on unseen data however, the standard Random Forest model produced a slightly higher F1-score. The most probable reason for this is the fact that cross-validation was done on data which is clustered beforehand, offering an unfair advantage. The time taken to evaluate the hybrid pipeline using cross-validation takes less than half the time to train the standard one, which is highly advantageous.

Table 2: Table providing the Accuracy and Weighted Precision, Recall and F1-score for classification when preceded by the different clustering techniques. The entries for K-Means, K-Modes and HDBScan refer to the traditional classification model which also utilises the cluster label as an additional feature. K-Means*, K-Modes* and HDBScan* refer to the hybrid approach where a separate classification model is fit on each of the clusters.

Clustering	Acc.	Prec.	Rec.	F1
Cross Validation Data (2015 - 2018)				
None	0.828	**0.828**	0.828	0.818
K-Means	0.824	0.827	0.824	0.814
K-Modes	0.829	**0.828**	0.829	0.819
HDBScan	0.822	0.827	0.822	0.814
K-Means*	**0.849**	0.823	**0.849**	0.821
K-Modes*	0.841	0.819	0.841	0.822
HDBScan*	0.848	0.826	0.848	**0.830**
Test Data (2018 - 2019)				
None	**0.828**	**0.828**	**0.828**	**0.818**
HDBScan	0.804	0.793	0.804	0.783

4.3 Objective 3: Automatically Discovering New Perpetrators

This section includes the new perpetrator detection results for the LOF, OCSVM, Nearest Neighbours, Autoencoder and GAN techniques. Table 3 shows that the unsupervised Nearest Neighbours model resulted in the highest Macro F1-score during cross-validation when it comes to detecting new terrorist groups. This model is then tested on the unseen test set and compared to a constant model which classifies all attacks as not having been carried out by unseen groups. While the obtained result is not outstanding, our model performs better than this constant, implying that it manages to detect some of the attacks as having been carried out by new groups.

Table 3: Table showing the novel class detection Accuracy and Macro Precision, Recall and F1-score for the LOF, OCSVM, Nearest Neighbours (NN), Autoencoder (AE) and GAN results.

	Accuracy	Precision	Recall	F1
Cross Validation Data (2015 - 2018)				
LOF	0.941	0.516	0.517	0.516
OCSVM	**0.951**	0.568	0.564	0.564
NN	0.930	**0.571**	**0.605**	**0.579**
AE	0.759	0.500	0.380	0.428
GAN	0.782	0.500	0.391	0.400
Test Data (2018 - 2019)				
Constant	**0.976**	0.488	0.500	0.494
NN	0.911	**0.537**	**0.616**	**0.547**

4.4 Objective 4: Predicting the Future Activity of Terrorists Groups

This section includes the new terrorist group forecasting results for the ARIMA and Prophet techniques. When predicting the future impact of the twenty most impacting terrorist groups, Table 4 shows that the Prophet model has the edge, resulting in slightly better Accuracy and Weighted Recall and F1-score.

Table 4: Table showing the Accuracy and Weighted Precision, Recall and F1-score for the Impact classification carried out using both Prophet and ARIMA.

	Accuracy	Precision	Recall	F1
ARIMA	0.535	**0.668**	0.535	0.581
Prophet	**0.602**	0.653	**0.602**	**0.620**

5 CONCLUSION AND FUTURE WORK

TerrorMine is a system which builds upon existing research focusing on counter-terrorism using data-driven methods. The first objective was to identify the groups responsible for terrorist attacks. The performed experiments show that the Random Forest seems to be the optimal classifier for this objective. While classification is not improved when handling missing data using specialised techniques, it seems to slightly improve when including textual data from the summary of attacks. The second objective involved exploring the effect of clustering on the perpetrator identification. Clustering the data before classification seems to have a negative impact on the result metrics when tested on unseen data, however model training and fine-tuning is made faster. The third objective involved the detection of attacks which were carried out by new groups. The results produced for this objective are not outstanding, however the Nearest Neighbours technique shows potential. The fourth objective was to forecast the terrorist activity of the different groups, with the performance of Prophet surpassing that of ARIMA in terms of weighted F1-score.

This work contributes to current research by comparing an extensive range of ML techniques on global and more recent GTD data to automatically identify terrorist groups. We also compare the sole utilisation of structured data, text data and ultimately a combination of both for classification. Additionally, we explore the niche of possibly detecting whether terrorist attacks were carried out by an unobserved group, by employing diverse techniques. To our knowledge, this has never been tackled before in this domain. Furthermore, we provide terrorist activity forecasts using Prophet, a tool which was not encountered in this field.

5.1 Future Work

A possible future improvement is to use the Gower distance instead of Euclidean where plausible since it considers both categorical and continuous data. The challenge with this approach is that the computation of a Gower matrix is computationally expensive. Furthermore, when it comes to clustering data to improve classification, the K parameter could alternatively be determined by varying it according to the classification results. Additionally, the clustering process could be used for outlier detection and removal, which could improve the results of all the objectives. Oversampling records related to minority terrorist organisations could also improve the results of the first two objectives. Another improvement is to use the terrorist organisation identification pipeline produced for the first objective to identify the terrorist organisations which are actually unknown, resulting in a more complete dataset which could improve the results of the remaining objectives. Furthermore, the terrorist groups which are determined to be inactive during the period of the unseen test data through the fourth objective could also be eliminated from the training data used for the identification of terrorist groups in the first objective.

REFERENCES

Alapati, Y. K. and Sindhu, K. (2016). Combining clustering with classification: a technique to improve classification accuracy. *Lung Cancer*, 32(57):3.

Alsawadi, H. and Bilal, M. (2021). Measuring novelty in autonomous vehicles motion using local outlier factor algorithm. *arXiv preprint arXiv:2104.11970*.

Battineni, G., Chintalapudi, N., and Amenta, F. (2020). Forecasting of covid-19 epidemic size in four high hitting nations (usa, brazil, india and russia) by fb-prophet machine learning model. *Applied Computing and Informatics*.

Carrara, F., Amato, G., Brombin, L., Falchi, F., and Gennaro, C. (2021). Combining gans and autoencoders for efficient anomaly detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3939–3946. IEEE.

Curia, F. (2020). Unsupervised hybrid algorithm to detect anomalies for predicting terrorists attacks. *International Journal of Computer Applications*, 975:8887.

Diab, S. (2019). Optimizing stochastic gradient descent in text classification based on fine-tuning hyper-parameters approach. a case study on automatic classification of global terrorist attacks. *arXiv preprint arXiv:1902.06542*.

Jackson, R. (2018). *Writing the war on terrorism: Language, politics and counter-terrorism*. Manchester University Press.

Jha, B. K. and Pande, S. (2021). Time series forecasting model for supermarket sales using fb-prophet. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 547–554. IEEE.

Khorshid, M., Abou-El-Enien, T., and Soliman, G. (2015a). A comparison among support vector machine and other machine learning classification algorithms. *IPASJ International Journal of Computer Science (IIJCS)*.

Khorshid, M. M., Abou-El-Enien, T. H., and Soliman, G. M. (2015b). Hybrid classification algorithms for terrorism prediction in middle east and north africa. *International Journal of Emerging Trends & Technology in Computer Science*, 4(3):23–29.

Laite, R., Lozano, M., and Sankaranarayanan, K. (2019). Terrorist group classification of historic terrorist attacks from the global terrorism database. In *2019 14th International Conference on Computer Science & Education (ICCSE)*, pages 237–242. IEEE.

Li, S., Zhuang, J., and Shen, S. (2017). Dynamic forecasting conditional probability of bombing attacks based on time-series and intervention analysis. *Risk analysis*, 37(7):1287–1297.

Mathivanan, N. M. N., Ghani, N. A. M., and Janor, R. M. (2018). Improving classification accuracy using clustering technique. *Bulletin of Electrical Engineering and Informatics*, 7(3):465–470.

Meng, X., Mo, H., Zhao, S., and Li, J. (2017). Application of anomaly detection for detecting anomalous records of terroris attacks. In *2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 70–75. IEEE.

Mohammed, D. Y. and Karabatak, M. (2018). Terrorist attacks in turkey: An evaluate of terrorist acts that occurred in 2016. In *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, pages 1–3. IEEE.

National Consortium for the Study of Terrorism and Responses to Terrorism (START) (2019). Global terrorism database. https://www.start.umd.edu/gtd.

Pagán, J. V. (2010). Improving the classification of terrorist attacks a study on data pre-processing for mining the global terrorism database. In *2010 2nd International Conference on Software Technology and Engineering*, volume 1, pages V1–104. IEEE.

Pan, X. (2021). Quantitative analysis and prediction of global terrorist attacks based on machine learning. *Scientific Programming*, 2021.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peng, A. (2018). An integrated machine learning approach to studying terrorism. In *Undergraduate Thesis*. Yale University.

Pfeiffer, C. P. (2012). Causalities and casualties: Media attention and terrorism, 1970–2010. Working Paper 127/2012, Helmut Schmidt University, Hamburg.

Rajan, S. (2021). Anomaly detection using autoencoders: A walk-through in python. https://tinyurl.com/2p92wm4p. Analytics Vidhja. Accessed June 2022.

Sahin, Y. (2018). *Forecasting the Monthly Occurrence of Terrorist Incidents Based on the GPI indicators and the GDP*. PhD thesis, Tilburg University.

Singer, G. and Golan, M. (2019). Identification of subgroups of terror attacks with shared characteristics for the purpose of preventing mass-casualty attacks: A data-mining approach. *Crime Science*, 8(1):1–11.

Smith, T. G. et al. (2017). pmdarima: Arima estimators for Python. http://www.alkaline-ml.com/pmdarima. Accessed June 2022.

Spinosa, E. J. and Carvalho, A. (2005). Support vector machines for novel class detection in bioinformatics. *Genet Mol Res*, 4(3):608–15.

Talreja, D., Nagaraj, J., Varsha, N., and Mahesh, K. (2017). Terrorism analytics: Learning to predict the perpetrator. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1723–1726. IEEE.

Taylor, S. J. and Letham, B. (2017). Prophet: Forecasting at scale. https://research.facebook.com/blog/2017/02/prophet-forecasting-at-scale/. Meta Research. Accessed June 2022.

Tolan, G. M. and Soliman, O. S. (2015). An experimental study of classification algorithms for terrorism prediction. *International Journal of Knowledge Engineering-IACSIT*, 1(2):107–112.

Zenati, H., Foo, C. S., Lecouat, B., Manek, G., and Chandrasekhar, V. R. (2018). Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*.