Imperial College London

Department of Computing

A framework for modelling and reasoning about Lone Actor Terrorism

Author:
Oana Cocarascu

Supervisor:
Dr. Dalal Alrajeh

Second marker:
Dr. Francesca Toni

June 16, 2015

Abstract

Identifying key risk factors for offending is recognised as one of the main challenges facing criminological research. In this project, we investigate the use of machine learning techniques to identify the most relevant characteristics and antecedent behaviors of lone actor terrorists for explaining their target selection. The choice for an ensemble of classifiers is guided by their proven capabilities to outperform the standard algorithms the ensemble consists of. The classifier ensemble is based on Support Vector Machines, known for generating good models even when handling limited data (which is the case in lone actor terrorism), Decision Trees and K Nearest Neighbours. Features are grouped according to three themes (ideology, network membership and mental illness) and explanatory models are generated under each theme using a pairwise feature selection model based on information gain. Each generated model comprises only those features that are considered most relevant with respect to target choices. These are then compared with a model generated using all variables combined (i.e. not with respect to themes). A Bayesian network is used with the attributes obtained from the best model as a graphical representation and inference engine to model the knowledge about individual terrorists and the relationships between characteristics.

Acknowledgements

I would like to thank Dr. Dalal Alrajeh for accepting to supervise my project, for inspiration and guidance throughout. I would also like to thank Dr. Francesca Toni for her advice and constructive feedback and Dr. Paul Gill for providing the original data and for suggestions. I wish to thank my parents. For everything.

Contents

1

Introduction

1 Motivation

Since the early 2000s, terrorism has become a threat to both civilian security and national security. There has been an increasing research on terrorism since 9/11 in order to understand its context and political justifications. It has been argued that terrorists choose their targets based on capabilities and constraints and the frequency of a civilian target is high because the civilian population is more accessible to conduct attacks against or because the preferred target is too difficult or costly. This phenomenon should be investigated and analyzed in order to understand its mechanisms and limit its effects.

Research has been conducted in the area of terrorism and has focused on the two types of terrorism: lone actor and group terrorism. There have been comparisons between terrorists and other offenders such as school attackers and assassins [1] and studies have been conducted to understand the factors that move an individual from extreme opinions to violent action, from radical opinion to radical action [2] and to understand the differences between lone actor terrorists and groups [3]. The contributors to this field have advanced insights regarding **ideology** and **affiliation** and how these are attributed to terrorist acts. Despite the limited number of **lone actor terrorists**, it has been observed that they exhibit specific personality and social traits compared to group-based terrorists that vary significantly [4]. There are environmental, social and individual characteristics which have been shown to increase the likelihood of engaging in a terrorist act [4, 5]. Whilst research suggested that lone wolf terrorists had been previously diagnosed with **mental illness**, this finding received minimal attention. This is the result of the discrepancies across the literature that assume an act of violence is either a terrorist act or the action of an individual affected by mental illness [3].

Lone actor terrorists plan and carry out an attack without assistance from others, they consider themselves to represent a larger group or cause and may have experience in an organization related to the cause [4]. The factors that determine an individual to move to violence include, but are not limited to, grievance, risk and status seeking, and loss of social connection [6].

Research aimed at understanding the underlying causes of terrorism have focused on three main areas: politically motivated violence in relation to political, economic and social factors, group dynamics, and psychological characteristics that determine individuals to join a violent organization [4, 7]. Gill categorizes lone actors by **ideology** and **network connectivity** [8] and provides three different models of a group: the group as an ideological support network, the group as an operational support network and the group as an operational unit/psychological support network [9]. While it has been shown that groups provide security, researchers have not succeeded in providing an explanation to how terrorists carry out attacks without the security of an organization.

Several attempts have been made to integrate machine learning algorithms in the field of criminology in order to understand crimes. The current techniques used in profiling and behaviour analysis can be grouped into: techniques used to derive geographic information [10, 11] and patterns [12, 13], and machine learning techniques with decision trees being the most frequently used method [14, 15, 16].

The Global Terrorism Database [1], a database used in research, influences terrorism studies. Due to the available data which includes information on events as opposed to behavioral aspects of offenders [3], these studies generally focus on terrorist events rather than on individual offenders. The current criminology studies that attempt to reveal the criminal behaviour focus on statistical analysis rather than on the relationships between the characteristics of terrorists and how these variables interact. Limited research has been done in order to understand terrorism with respect to the target choice with [17] proposing a model to represent an airline hijacking and [18] predicting the likelihood of terrorist activities at critical transportation facilities.

2 Objectives

The objective of this project is to provide a better understanding of the characteristics and behaviour of lone actor terrorists using a machine learning approach. Machine learning algorithms are not novel within the area of criminology and have been previously applied to understand crimes and to identify patterns, but have not been widely applied in studying terrorism. Our approach identifies the terrorist's behaviour with respect to their target choice, high value or civilian, and the key factors identified in the literature: mental illness,

[1] http://www.start.umd.edu/gtd/

ideology and network organization.

Compared to other studies that have been conducted in the field of criminology, this project explores data related to individual offenders rather than terrorist events or other type of crime. Overall, the aim of the project is to provide a framework for identifying the most relevant features for explaning lone actor behaviour and using these features to classify lone actor terrorist target choice, *high value* or *civilian*, and to represent the relations between variables. The project seeks to provide a greater understanding of the terrorist behaviour and to understand the variables that form the characteristics of lone wolves.

The objectives of this thesis and how they are implemented are:

- Identifying the most important features. Feature selection is applied in order to eliminate low quality features. We propose a pairwise feature selection algorithm and we test whether a variable is more useful if taken with others using information gain ratio.
- Building a model that will classify lone actor terrorist target choice, *high value* or *civilian*. We suggest a supervised classification of high value/civilian target using an ensemble of classfiers consisting of Support Vector Machines, known to generalise well even in cases with limited training data, Decision Trees and K Nearest Neighbours on the entire dataset, as well as dividing the complete set of features into three groups: ideology, network, mental illness, and train an ensemble on each group, two techniques being used for the final classification, Majority Voting and Weighted Majority Voting.
- Building a probabilistic model with the most relevant features. We use a Bayesian network to model the relationships between the features identifed to be the most informative using our feature selection method. The network is able to represent causal and probabilistic relationships and to combine expert knowledge with data. It is used to model the relationships between lone wolf characteristics, how these variables relate to one another and to infer unobserved variables given any evidence.

The results obtained using machine learning algorithms are then compared with the key points found in the research conducted by criminologists outlining the psychology of terrorists. We review previous findings from Paul Gill's report [8] that used the data we modelled in this project. Apart from the theoretical findings that arise from studying terrorism, the project builds a model that will help understand various interactions between characteristics of lone wolf terrorists, which can be used in comparison analysis and to evaluate hypotheses such as the influence of certain variables on the target choice.

While preventing an act of terrorism still remains an uncertain task, this project seeks to explore the characteristics of terrorists, which, if used by experts in criminology, could provide a better understanding of lone wolves and hence reduce the risks of terrorist acts. The ultimate goal in this field is to be able to disrupt or even prevent attacks, whereas this project aims to explain the characteristics and behaviour of lone actor terrorists, guide researchers to the aspects they should focus on, refine knowledge and test theories.

3 Contributions

This projects presents a machine learning aproach for behaviour analysis of lone actor terrorists. The framework allows to determine important features that can explain lone-terrorist behaviour and to understand the relations between these features. It can guide researchers on what aspects they should focus on by allowing them to test hypotheses and theories, to refine knowledge and to identify key factors. The main contributions are:

- We developed a feature selection method that combines information gain with the idea that two insignificant features by themselves can be useful together. Our method tests whether a variable that is not informative by itself can provide a significant performance improvement when taken with others. The approach consists of identifying the most informative features using information gain ratio.
- We built a classification model for lone terrorist target selection. We implemented a supervised classification of a binary target, high value and civilian, using an ensemble classifier that consists of Decision Trees, K Nearest Neighbours and Support Vector Machines.
- Our classification approach provides a theoretical contribution as we used two strategies in performing terrorist target classification: firstly, using the data set with a single set of features and secondly, using the data set with features divided into categories and combined the outputs using Weighted Majority voting. The results of the experiments we conducted using the ensemble classifier underline how the proposed technique of splitting the feature set into groups outperforms in terms of accuracy the standard approach of using a classifier for a single group of features.
- We created a model using Bayesian networks that can handle the probabilistic aspect of behaviour analysis. We used the most informative features as identified when we modelled classification in order to generate the networks. We constructed three network structures for each of the three themes, network, ideology and mental illness from the data set. We used the networks for probabilistic inference in order to identify the most likely value of each variable based on evidence.

- The interaction between variables belonging to different themes can be obtained by combining the smaller networks. We combined the networks rather than learning the network from the data in order to obtain a more accurate model given the number of nodes for the combined network and the size of our data set.

- We improved the network by allowing an expert to modify the network, that is introducing expert knowledge into the network. We developed a web-based analysis graphical interface which allows experts to change a network and thus incorporate domain knowledge through an application that provides a simple and efficient visualisation of variables related to lone terrorist behaviour as well as how these variables interact and influence each other.

- We provided a framework that overcomes the limitations of the standard technologies used in criminology (i.e. Smallest Space Analysis) by considering pairs of features and representing the relations between variables.

2

Background

1 Terrorism Background

1.1 Terrorism

Terrorism represents the violence against people and the damage of properties, or facilitating any of these actions, in which the safety of the public is at risk and which is designed to influence the government, to intimidate the public or to advance a political, religious or ideological cause [8]. However there are over a hundred definitions of terrorism [19], all having three common elements: the use of violence, the objectives and the intent to intimidate the target group. There are five special characteristics of terrorism: premeditated, directed to a target group, violent, antisocial and that it can be influenced in order to achieve a goal. The contributors to this field have advanced insights regarding ideology and affiliation and how these are attributed to terrorist acts. There are environmental, social and individual characteristics which have been shown to increase the likelihood of engaging in a terrorist act [4, 5].

Research aimed at understanding the underlying causes of terrorism [4, 7] have focused on three main areas: politically motivated violence in relation to political, economic and social factors, group dynamics, and psychological characteristics that determine individuals to join a violent organization. The studies focus on two types of terrorism: group terrorism and lone actor terrorism.

1.2 Group terrorism

Gill provides four different models of a group: the group as a social movement, the group as an ideological support network, the group as an operational support network and the group as an operational unit/psychological support network as shown in Figure 1. The number of members of a group, how they are trained for a particular type of violence and the level of involvement depends on the group type. The members form a homogenous group, they have a shared religious and ideological identity and are influenced by the network to carry out an attack and to facilitate the attack.

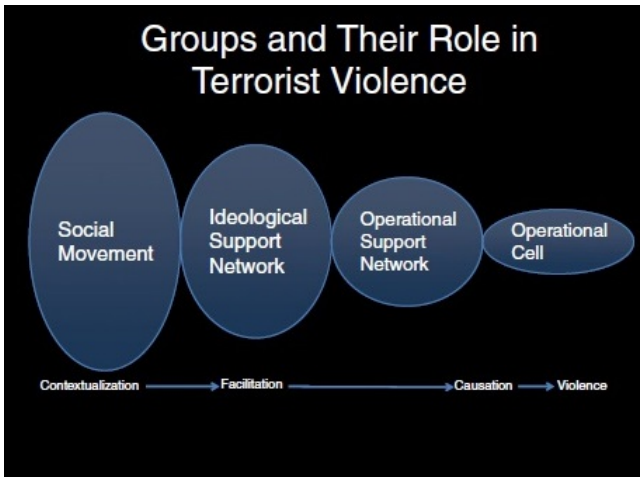


Figure 1: Group role in terrorism [9]

1.3 Lone actor terrorists

A lone actor terrorist is an individual who engages in illicit or high-risk behaviour [8]. Several other definitions of lone actor terrorists can be found in the literature. In [1] a lone-wolf terrorist is defined as an individual who plans and carries out an attack without organisational support, while Pantucci [20] defines the lone wolf as an individual who acts alone while having connections with a terrorist organisation. Pantucci [20] identifies four types of lone wolf terrorists: loner, lone wolf, lone wolf pack and lone attacker.

Despite the limited number of lone actor terrorists, it has been observed that they exhibit specific personality and social traits compared to group-based terrorists that vary significantly [4]. It has been reported that in absence of a group, an individual faces more challenges in organizing a terrorist attack and that he may need to have a source of income and may need to develop the necessary practical skills [3].

Research has focused on comparisons between assassins and school attackers in order to make hypotheses about lone actor terrorists with respect to grievance [1] and on similarities between lone terrorists and group terrorists [3]. Most of the studies focus on events rather

than individual offenders [3] due to the type of data that can be found in the Global Terrorism Database (GTD), an open-source database which includes information on terrorist events between 1970 and 2013 collected from media reports. It contains information relevant to the incident such as the weapons used and target. Each case contains minimum 45 variables, with recent attacks having more than 120 variables. Due to the type of data found in the database, terrorism studies generally focus on terrorist events rather than on individual attackers.

Key findings suggest that lone actor terrorists plan and carry out an attack without assistance from others. They consider themselves to represent a larger group or cause and may have experience in an organization related to the cause. The lone actor terrorist takes risks and makes sacrifices as a free choice and not because of social pressures. The lone actor is not influenced by a group, does not have support and the attack is prepared secretly. The factors that determine an individual to move to violence include but are not limited to grievance, risk and status seeking, and loss of social connection [4, 6].

Despite the fact that research suggested that lone wolf terrorists had been previously diagnosed with mental illness, this finding received minimal attention. This is the result of the discrepancies across the literature that assume an act of violence is either a terrorist act or the action of an individual affected by mental illness [3]. In [8] Gill et al categorize lone actors by ideology and network connectivity.

1.4 Thematic analysis

Thematic analysis is a commonly used qualitative analysis method in psychology and other domains. It is a way of identifying themes within data. A theme represents an important aspect in the data, a meaning within the data set and is a collection of characteristics.

2 Features Background

A feature is the specification of an attribute and its value. A feature vector consists of a list of features describing an instance [22]. Features can be grouped into two categories: *discrete* and *continuous*. Discrete variables take a finite number of fixed values, whereas continuous variables can take values within a range.

Dimensionality reduction seeks to reduce the number of attributes in the data set. There are two methods used for dimension reduction: *feature selection* which selects a subset of features from the initial set of features and *feature extraction* which creates newly derived combinations of attributes [23, 24, 25, 26]. Feature selection methods divide into *filters* and *wrappers*.

An algorithm is said to *overfit* if the performance on the training examples increases, while the performance on unseen data decreases. Eliminating redundant and noisy features decreases the computational complexity and decreases the overfitting in supervised learning, especially in the case of a limited number of training examples with a large number of features.

2.1 Information gain

The information gain is used to select the attribute that best splits the output. It is biased towards choosing attributes with a large number of values or states and this may lead to overfitting. Information gain, sometimes used synonymously with mutual information, measures dependency between variables and it is more general in determining non-linear relationships. For example the date of birth will have the highest information gain because of the number of states this variables has, but choosing this attribute will result in a poor generalization model.

Information gain (IG) is based on Entropy, which characterizes the impurity of a system of examples. Minimum impurity is achieved when the examples belong to a single class. It is defined mathematically:

$$Entropy(S) = \sum_{i \in classes} - \, p_i \; log_2 \; p_i \tag{1}$$

where p_i is the probability of class i.

Information gain represents the reduction in Entropy(S) caused by partitioning the system S of examples according to attribute A:

$$IG(S, A) = Entropy(S) \; - \sum_{v \in values(A)} (|S_v|/|S|) Entropy(S_v) \tag{2}$$

Information gain ratio (IGR) is a modification of the information gain that reduces the bias towards multi-valued attributes and is the ratio between the information gain and the intrinsic value (IV) which represents the distribution of instances into branches.

$$IV(S, A) = -\sum_i \frac{|S_i|}{|S|} \; log_2 \; \frac{|S_i|}{|S|} \tag{3}$$

$$IGR(S, A) = \frac{IG(S, A)}{IV(S, A)} \tag{4}$$

Features represent an important aspect of machine learning.

3 Machine Learning Classifiers Background

Machine Learning is a broad term which comprises a significant number of algorithms that automatically learn from data and improve with experience [30]. Nowadays, its applications are becoming more and more important as much of the work on visual processing, language and speech recognition relies on Machine Learning and Artificial Intelligence.

Definition: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E [31].

Machine learning algorithms are typically classified into two broad categories: *supervised* and *unsupervised*.

Supervised learning algorithms learn a general function from labelled training data. Each example in the labelled training data consists of an input vector and an output value known as a class. The aim of the supervised learning algorithms is to generalise the labelled examples and to correctly classify unseen inputs.

Unsupervised learning algorithms handle unlabelled training examples. The aim of these types of algorithms is to find hidden patterns and structure in the input data.

In *batch* learning the learner receives all the data at the start of the learning. In *online* learning the learner receives one example at a time, and assigns a label to an example before receiving the correct value. The current hypothesis is updated with each example [32].

3.1 Decision Tree

Decision Trees approximate discrete classification using a tree-based representation. Each node in the tree represents an attribute and each branch corresponds to a value of that attribute. Using a top-down search, the Decision Tree selects the attribute that classifies most of the examples using a statistical property called Information Gain. The root of the tree is represented by the feature that best separates the training data. In general, to avoid overfitting, the algorithm is stopped before reaching the point where it perfectly classifies the training data. Decision Trees can be translated into a set of rules where each rule is given by a branch and each class is represented by a disjunction of rules [30].

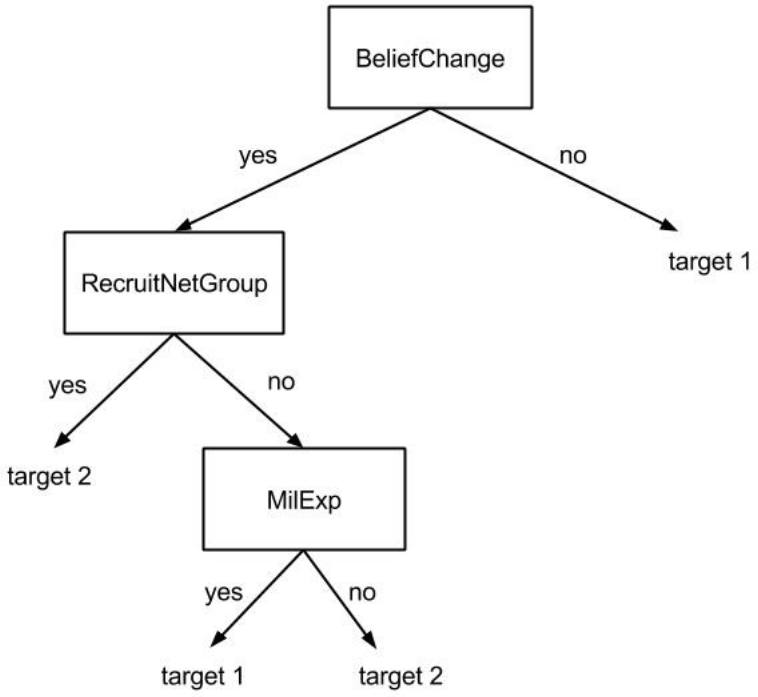


Figure 2: An example of Decision Tree for terrorism target choice.

3.2 K Nearest Neighbours

K Nearest Neighbours algorithm is an instance-based learning method that stores the training examples. It assumes the instances correspond to points in the n-dimensional space where each dimension corresponds to a feature, and that similar instances are in close proximity. The neighbours of an example are computed using the Euclidean distance. If an instance x_i is described by the feature vector $< x_{i1}, x_{i2}, ..., x_{in} >$ where x_{it} represents the value of the tth attribute of instance x_i, then the Euclidean distance between two examples x_i and x_j is defined:

$$d(x_i, x_j) = \sqrt{\sum_{t=1}^{n} (x_{it} - x_{jt})^2} \tag{5}$$

The new example is assigned the most frequent class label. The performance of the algorithm is dependent on the choice of k [30].

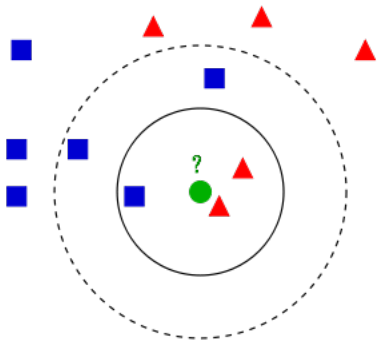


Figure 3: An example of KNN classification [33]. A new example can be assigned to a square or a triangle. If k=3 (first circle) the new example is assigned to triangle, whereas if k=5 (second circle) the new example is assigned to square since there are 3 squares and 2 triangles.

3.3 Support Vector Machines

The Support Vector Machine, introduced in [34], represents the state of art in binary classifications. A Support Vector Machine maps the examples into a high dimensional feature space [35]. For linearly separable data, a two dimensional feature space can be separated by a line. A three dimensional feature space can be separated by a plane, and a multi-feature space can be separated by a hyperplane. Given a set of examples $\mathbf{x_i}$ with labels $y_i \in \{1, -1\}$, the algorithm finds the parameters of the decision function $D(\mathbf{x})$ during learning and assigns the class 1 if $D(\mathbf{x}) > 0$ and -1 otherwise [34]. The decision function is:

$$D(\mathbf{x}) = \sum_{i=1}^{N} w_i \varphi_i(\mathbf{x}) + b \tag{6}$$

Support vectors represent the training examples closest to the hyperplane as seen in Figure 4. The highest generalisation ability of the classifier is achieved by the hyperplane with a maximum margin. The hyperplane represents the decision boundary.

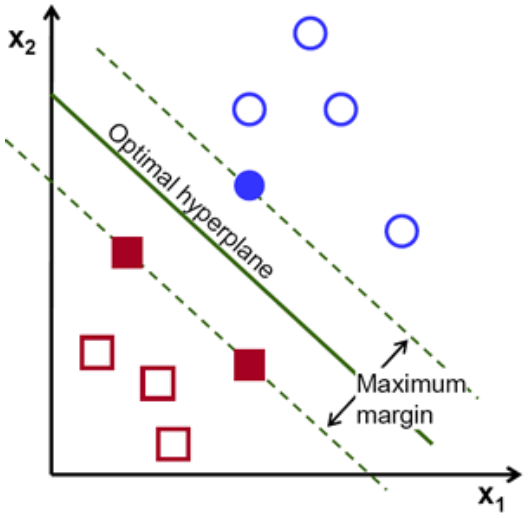


Figure 4: Support Vector Machines for binary classification: circle or square [36]. The support vectors are the geometric figures seen on the dotted lines. The optimal hyperplane maximizes the margin of the data.

More explicitly, the area of the feature space is separated into two, each point belonging to one of the classes. The aim is to maximise the margin between the two classes, that is the largest distance between the decision boundary and the nearest training data points. A new example will be classified by testing which side of the hyperplane the point lies on.

One of the main advantages of SVMs is that non linearly separable data can be mapped to a higher dimensional space which can then be separated using a linear hyperplane. This is known as the 'kernel trick'. Inner products are calculated directly using kernels, without performing the actual mapping.

Mathematically, ϕ maps points from a n-dimensional space to a m-dimensional space where m > n: $\phi : R^n \rightarrow R^m$. The kernel function computes the dot product in a higher dimensional space: $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$.

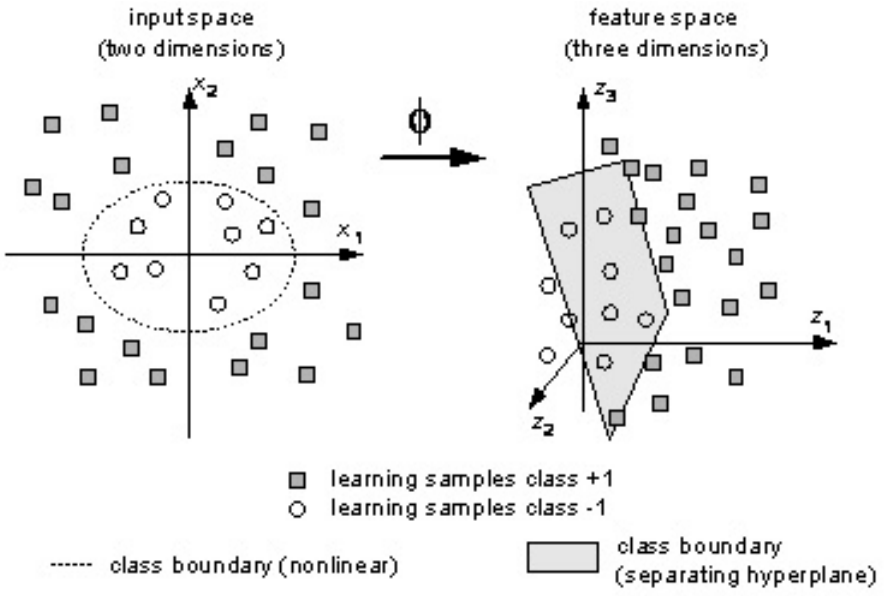


Figure 5: Non-linear two dimensional space mapped into a three dimensional space which can be linearly separated by a hyperplane [37].

3.3.1 RBF kernel for SVM

The radial basis function (RBF) is a popular kernel function used in Support Vector Machine classification. The radial basis function is given by the following equation:

$$K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2), \gamma > 0 \tag{7}$$

The RBF kernel can handle the case in which the features and labels are nonlinear by mapping the examples into a higher dimensional space. The RBF kernel has two parameters, C and γ. The goal is to identify C and γ so that the classifier can accurately predict new data.

C represents the penalty parameter of the error term and is the trade off between misclassification of training points and simplicity of decision surface. C represents the trade off between margin maximisation and error minimisation. A low value of C results in the decision surface being smooth that can lead to underfitting, whereas a high value of C results in a high penalty for nonseparable points and overfitting, that is correctly classifying all training examples by selecting more examples as support vectors [38].

γ represents the kernel coefficient and defines the influence a single example from the training set can have. The larger the value of γ, the closer other examples must be to be affected. If the value of γ is very large, the area of influence of the support vectors will contain only the support vector and the C parameter will not be able to overcome overfitting in this case.

If the value of γ is very small, the area of influence of the support vectors will contain the entire training set, resulting in a model that cannot capture the complexity of the data [38].

3.4 Ensemble classifiers

Ensemble classifiers use multiple learning algorithms in order to achieve a better performance than the one achieved by any of the constituent algorithms. The ensemble integrates multiple classifiers in order to make use of the strengths of one method to complement the weaknesses of another learning algorithm. The prediction of each model in the ensemble has equal weight in the final classification decision.

4 Evaluating a Machine Learning Classifier

An important aspect of a machine learning algorithm is estimating the performance of the system. Learning the parameters of a supervised machine learning algorithm and testing the model on the same data will result in a high prediction score but the model will fail to correctly classify unseen data. This situation is known as *overfitting*. A common practice is to train the classifier with training examples and test it on a different set of data, the test examples. The training set is usually further divided into training data and validation data. In order to avoid overfitting, the validation data is used to optimise the parameters of the classification model.

4.1 k-fold cross validation

The aim of cross validation is to ensure that the results will statistically generalise to an independent data set and it is used when the amount of data for training and testing is limited. The set of examples is divided into k folds and k training iterations are performed. Each iteration, k-1 folds are used for training and the resulting model is validated on the remaining data. The performance measure reported by the k-fold cross validation is the arithmetic mean of the individual measures obtained in each of the k iterations.

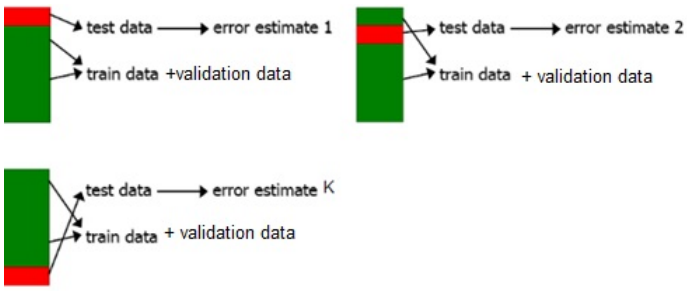


Figure 6: k-fold cross validation [39]

4.2 Confusion matrix

A confusion matrix is typically used as a visualisation tool to evaluate the output of a classifier. The rows of the matrix represent instances in an actual class, and the columns of the matrix represent instances in a predicted class.

		Predicted class		
		Cat	Dog	Mouse
Actual class	Cat	5	2	1
	Dog	2	2	1
	Mouse	0	0	3

Table 1: Confusion matrix

		Predicted class	
		Cat	Other
Actual class	Cat	5 (TP)	3 (FN)
	Other	2 (FP)	6 (TN)

Table 2: True Positives (TP), False Negatives (FN), False Positives (FP) and True Negatives (TN) for class Cat

4.3 Accuracy

Accuracy is defined using mean squared error (MSE). Given a vector of predictions $\hat{\mathbf{Y}}$ and the vector of true values $\mathbf{Y}$ the MSE is mathematically expressed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{\mathbf{Y}}_i - \mathbf{Y}_i)^2 \tag{8}$$

4.4 Recall and Precision rates

Recall rate describes the completeness of the retrieval and is represented by the percentage of positive labelled instances that were predicted as positive. Precision rate describes the accuracy of the retrieval and is represented by the percentage of positive predictions that are correct. Intuitively, recall expresses the classifier's ability to find all positive examples, whereas precision represents the ability to not label a negative example as positive.

$$Recall\ rate = \frac{TP}{TP + FN} \times 100\% \tag{9}$$

$$Precision\ rate = \frac{TP}{TP + FP} \times 100\% \tag{10}$$

4.5 $\mathbf{F}_\alpha$ measure

The F_α measure combines the recall and precision rates in a single equation. α defines how recall and precision rates are weighted. If recall and precision are equally weighted, α is 1.

$$F_\alpha = (1 + \alpha) \times \frac{precision \times recall}{\alpha \times precision + recall} \tag{11}$$

5 Bayesian Network Background

Bayes' theorem is named after Rev. Thomas Bayes and is stated mathematically as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{12}$$

where P(A) and P(B) are the probabilities of events A and B and P(B|A) is the conditional probability, the probability of B given that A is true.

The Bayesian Network (BN) is a graphical model where nodes represent features and arcs represent conditional dependencies between nodes. The network is also referred to as a directed acyclic graph (DAG) where each arc indicates the direction of the effect of one variable on another variable. Each link has an associated conditional probability table (CPT) which encodes the relationships between the connected nodes. Absence of arcs in the network implies conditional independence. Bayesian Networks are based on the Markov property, that is there are no other direct dependencies in the graph apart from the ones represented by arcs [27].

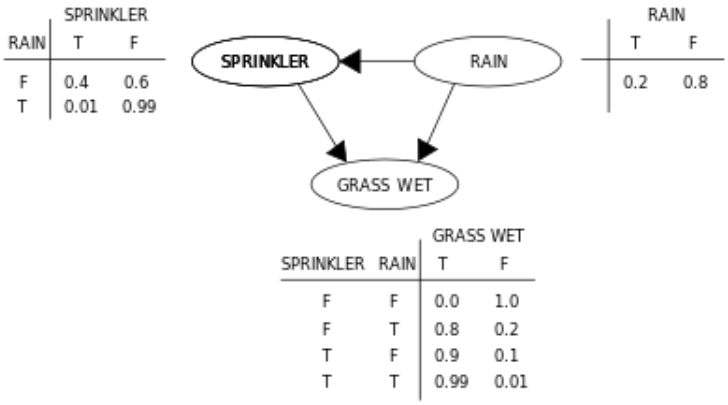


Figure 7: Bayesian network with associated conditional probability tables [28]. In this figure, the node "Rain" is the parent of both "Sprinkler" and "Grass wet".

The Bayesian network represents a joint distribution P over a set of random variables $X = X_1, .., X_n$:

$$P(X_1, .., X_n) = \prod_{j=1}^{n} P(X_j | parents(X_j)) \tag{13}$$

where $parents(X_j)$ represents the set of variables X_i such that there is an arc from node i to node j.

Bayesian inference represents a probability model which uses Bayes rule to update the parameters of the model as evidence is acquired. In Bayesian statistics, probability is used as a measure of uncertainty and the methods rely on a prior distribution [29]. Given the value of a subset of variables in the network, referred to as evidence, we update the probabilities to incorporate the new evidence. This is achieved through message passing: each node sends a message to its parents and its children. There are two types of networks: *singly connected* and *multiply connected*. The singly connected network is a directed acyclic graph whose underlying undirected graph has a single path between any given two nodes. A multiply connected network has more than one path between nodes. Propagation does not always terminate in the case of multiply connected networks.

Bayesian networks are used in bioinformatics, image processing, risk analysis. The Bayesian network can also be used to represent the probabilistic aspect of behaviour analysis.

5.1 Gibbs sampling

Given a Bayesian network and some evidence, we can estimate the posterior probabilities of the variables by drawing n samples and calculate the probabilities based on frequencies. This is known as a Monte Carlo Markov Chain method and one popular MCMC algorithm is Gibbs sampling. The main idea is that samples are not independent, but are correlated with each other and the distribution of a node X can be calculated using only the Markov blanket of X: children of X, parents of X and parents of children of X. Through sampling the values get closer to the posterior distribution.

The Gibbs method starts by sampling from the distribution not conditioned on the evidence. The unknown nodes are assigned a random state. It then iterates over the unobserved variables, selecting a node and calculating its state using its Markov blanket. The node is instantiated based on the calculated distribution and a new sample is recorded using the calculated state of the node and the current sample for all other variables.

3

Related work

1 Criminology related work

There exist a number of studies in criminology which attempt to explain crimes and criminal behaviour, most of them focusing on statistical analysis.

A paper related to our work is [3] which illustrates the impact of Gruenewald et al's [40] work on the current state of the art in criminology. The key findings are: the relation between mental illness and terrorist behaviour and the role the group plays in suicide terrorism by providing the necessary operational support to perform an attack and by influencing an individual's moral [9].

One important work, on which we based our decision is [8]. In this paper, Gill et al shift from the current approaches in criminology in order to focus on behavioral analysis. They examine the socio-demographic network characteristics, antecedent behaviour and ideology and provide statistical analysis results.

1.1 Multidimensional scaling

Multidimensional scaling (MDS) techniques are the ones that are most frequently used in psychology and other social sciences because they are relatively easy to implement and their visual output can be intuitive to interpret [41] by providing the level of similarity of individual cases within a data set. They can be used to represent relations between features that are non-linear or multidimensional. The output is a spatial representation of the data, representing the points in space. MDS procedures are the ones that are generally used in profiling [41, 42, 43].

Mokros and Alison suggest that the similarity between background characteristics of offenders influences the resemblance in their crime behaviour [42]. They tested a multivariate

correlative hypothesis using Smallest Space Analysis (SSA), a non-metric MDS procedure to represent the behaviour maps and centroid measure to obtain indexes of behaviour. Their simulation tests show that 64.5% were convicted as juveniles, 75.3% were convicted as adults and 93.8% had a previous conviction either as a juvenile or as an adult.

In [43] MDS procedures are used to test the radex hypothesis, a model which groups the features shared by all offences in the center while the specific features are moved towards the periphery.

The work of McCauley et al [1] starts from the assumption that lone wolf terrorists have common characteristics with other lone actor violent offenders, assassins and school attackers. They identified the following common characteristics: grievance, depression, unfreezing (personal crisis), and a history of weapons use outside the military, which suggest the importance of means and opportunity [2]. The authors reported that nearly half or more of school attackers and assassins exhibited these characteristics:

Characteristic	School Attacker	Assassin
Grievance	81%	67%
Depression	78%	44%
Unfreezing	98%	50%
Weapons use	63%	71%

Table 1: Characteristics results of school attackers and assasins

Most attempts on explaining crime and criminal behaviour are based on two assumptions: consistency (any offender's actions are consistent across offenses) and homology (background characteristics influence offense styles) [44]. There are two problems in these approaches: (a) lone wolf terrorist are heterogeneous [8] and (b) the inference theory is not well understood by criminologists and they make inferences about non-psychological characteristics such as age, gender, marital status etc.

2 Feature selection related work

Feature selection using mutual information and pairwise feature selection algorithms have been previously described in the literature [45, 46, 47, 48, 49].

In this paper [45] Lefakis and Fleuret propose a method for selecting groups of features that are jointly informative in classification. They combine the Gaussian models of the features with the maximised upper bounds of the individual variable's mutual entropy with the label and the joint entropy. They argue that whilst joint features rely on the individual mutual information of the variables and this method is computationally expensive, it is in fact better to compromise on the density model that allows for an efficient computation.

Their algorithms are comparable with state-of-the-art methods.

Hall presented a method for pairwise combinations of features in [46]. The Correlation-based Feature Selection assumes that informative features are uncorrelated with one another but correlated to the target class. He proposed an algorithm that considers pairs of features where a derived feature is considered for feature selection if the correlation with the target class is higher than its constituent variables. The original feature selection process, that is informative features are uncorrelated with one another but correlated to the target class, is then applied to these variables.

In this paper [47] a method using mutual information and based on combination of feature ranking and forward selection is proposed. The authors argue that mutual information of individual features and target class leads to features that are relevant together to be ignored. Their algorithm identifies the feature with the highest mutual information with the class and then considers pairs of features containing the selected variable and the other features. The next selected variable is the one from the pair with the highest mutual information. The method stops once the specified number of variables is reached.

Another pairwise feature selection procedure is described in [48] where it is stated that this method can be useful in cases that deal with small sample size.

The algorithm proposed in [49] combines Max-Relevance and Min-Redundancy criteria and uses mutual information to select features relevant to the target class that are not pairwise redundant, that is features that have high mutual information with the class and low mutual information between each other.

3 Machine Learning related work

Machine Learning algorithms are not novel within the area of criminology. They have been previously applied to understand crimes. The current techniques used in profiling and behaviour analysis can be grouped into: techniques used to derive geographic information [10, 11] and patterns [12, 13], and machine learning techniques with decision trees being the most frequently used method [14, 15, 16].

Dahbur and Muscarello [54] developed an unsupervised technique using Kohonen Neural Networks that can identify group of records as patterns for serial criminals. They focused their research on armed robberies. The features were split into four groups of similar features (crime vectors, offender vectors, victim vectors and vehicle vectors). Each group of features is presented to a Kohonen network and the classifications given by each individual network are then combined and fed into a Kohonen network that will provide the final classification. With a test data that contained ten patterns, they were able to predict this number of patterns in 64% of the cases.

The paper [16] compares forecasting performance of three classifiers: logistic regression, random forests, and stochastic gradient boosting, all of which were shown to perform well in criminal applications. They conducted their experiments on 20,000 observations of serious crimes (murder, attempted murder, arson etc) and predicted whether an individual is arrested for a serious crime within two years of release on probation. Their results are shown in the following table:

		Fail	No Fail	Model error
Logistic Regression	Actual Fail	378	302	0.444
	Actual No Fail	1385	2935	0.321
Random Forests	Actual Fail	427	253	0.372
	Actual No Fail	1196	3124	0.277
Stochastic Gradient Boosting	Actual Fail	396	284	0.418
	Actual No Fail	1361	2459	0.315

Table 2: Confusion table for serious crime

Their work has its limitations. They randomly split the training and testing data for each of the algorithms. They have shown that the date of birth is the most important factor in determining whether an individual will be arrested for a serious crime, but choosing this attribute will result in a poor generalization model:

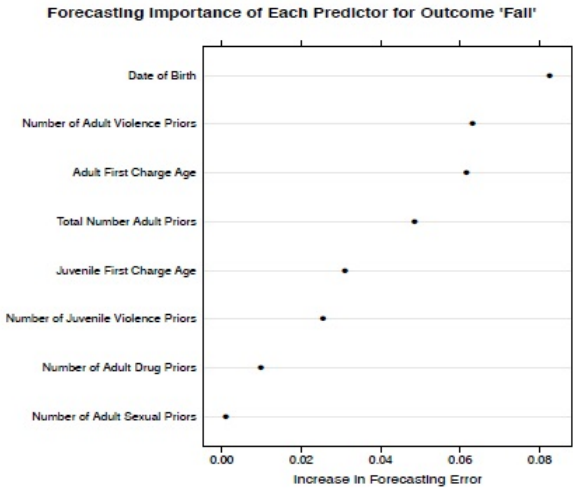


Figure 1: Feature importance in serious crimes [16]

One paper [15] provides perspective into tree-based machine learning algorithms with the goal of forecasting criminal behaviour. Classification trees have proved to be effective in criminal classification and criminal forecasting and the methods outlined by Berk are: random forests, stochastic gradient boosting and Bayesian additive regression trees. The importance of a feature is measured by the reduction in accuracy occurring when the feature is

removed from the prediction. Each time forecasts are made, a feature is shuffled, changing the information each attribute brings to the forecast:

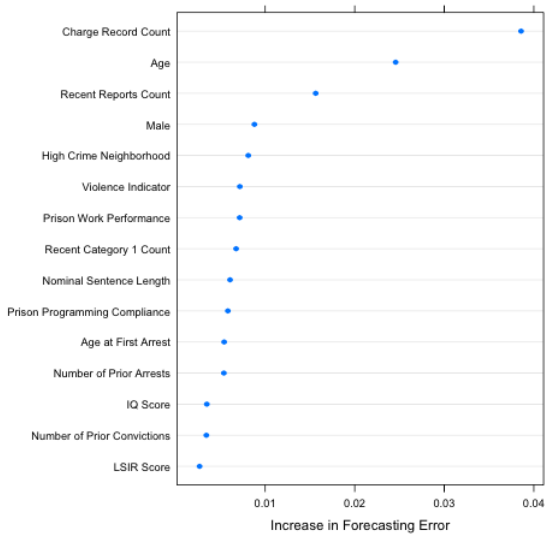


Figure 2: Feature importance measured by proportional reductions for violent crimes committed within 2 years of release on parole. When variables related to the number of prison misconduct (Charge Record Count, Recent Report Count) are shuffled, accuracy drops from 60% to 56% [15]

The paper [11] showed how Bayesian learning can be used to model serial crimes in order to predict the geographic location of a next crime. The features selected by Liao et al were only related to geographic information. Before the first offense from a crime series, each feature is given the same weight. The geographic distribution is approximated by a Gaussian distribution and each is given a weight based on the distance decay function. The developed method uses effect functions that are adjusted adaptively using Bayesian learning theory. The results presented are based on a case from China. There is no discussion in the paper on how the algorithm performs on a new data set. One other work that uses geographic information is [10], which detects patterns of criminal behaviour based on geographic location.

The study [57] compares five classification algorithms predicting the crime status, critical and non-critical. Two feature selection methods were used in this study: one in which attributes were manually selected by human experts, yielding 44 attributes, and one in which a Chi-square test was used to identify the correlated features (94 attributes). The following tables summarise the algorithm performance:

Method	Precision (%)		Recall (%)	
	44 features	94 features	44 features	94 features
Naive Bayesian	86.7	87.5	84.6	84.4
Decision Tree	84.6	85.7	85.0	86.3
Support Vector Machine	85.0	86.1	85.6	86.5
Neural Network	85.1	86.8	85.3	87.1
k Nearest Neighbour (k=10)	86.9	87.3	87.5	88.0

Table 3: Precision and Recall rates for serious crimes for two sets of attributes

Method	Accuracy (%)		AUC	
	44 features	94 features	44 features	94 features
Naive Bayesian	84.646	84.395	0.894	0.898
Decision Tree	84.997	86.251	0.731	0.727
Support Vector Machine	85.649	86.452	0.66	0.678
Neural Network	85.298	87.054	0.882	0.892
k Nearest Neighbour (k=10)	87.506	88.008	0.897	0.895

Table 4: Accuracy and AUC for serious crimes for two sets of attributes

Wright conducted a study on Western European terrorism activities between 1965 and 2005 [58]. He divides the targets into four categories: political leadership, security, civilian and rival terrorists and the ideology into four groups: nationalist/separatist, sectarian, left wing and right wing. He models the relationships between target, ideology and the government's response to attack which can be forceful, juridical or no action as a game, defining the parameters and equilibrium conditions. His findings suggest that using a multinomial logistic extension of statistical backwards induction to reach each decision node, ideology is the only informative predictor of target selection. The study only considers the ideology feature in choosing the target and does not compare how other features influence the target selection.

In [59] Gohar et al developed an ensemble technique for classification of the terrorist group responsible for the attack. Using data of terrorism incidents from 1970 to 2012, they show that the majority voting of individual classifiers gives better results than the base classifiers: Naive Bayes, K Nearest Neighbour, Iterative Dichotomiser 3. Their results are summarised in the following table:

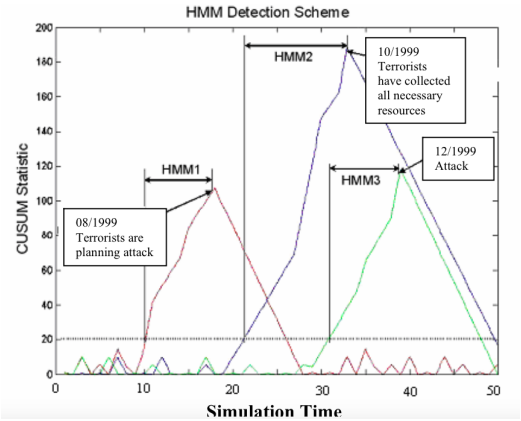
Classifier	Accuracy %
Naive Bayes	92.75
KNN	83.43
DS	91.30
ID3	84.97
Majority Voting	93.40

4 Bayesian Networks related work

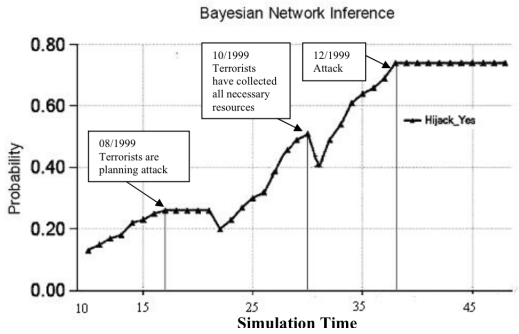
Ezell et al [14] discuss various techniques used in terrorism risk analysis. These include probability trees, event trees, and decision trees. The study also outlines the use of Bayesian networks not only in the development of anti-terrorism models, but also in predicting the distribution for lethal exposure to chemical agents. The main problems in these tree approaches is that probabilities are assigned by experts only, and the weights have to be changed with each new case. There is no discussion on how well a Bayesian Network performs. In [56] the authors provide perspective into how Dynamic Bayesian Networks can be used for investigating the interdependency of urban infrastructure during extreme events. It is also suggested that one of the early models developed that was using Bayesian Networks, Site Profiler, predicted that the Pentagon was a likely terrorist target.

The paper [50] proposed a statistical machine learning algorithm based on Relational Dependency Networks (RDN) with relational probability trees (RPT) for each attribute, and Gibbs sampling for inference in order to perform collective classification. Delaneyy et al used graphical representations from The Institute for the Study of Violent Groups (ISVG), a research group that maintains a database of terrorist and criminal activity from open-source documents. Their first contribution was predicting leadership roles of individuals within a group. Additionally, they predicted the outcome of hostage negotiations (released or killed) in kidnapping events. They reported an AUC performance (area under the receiver operating characteristic (ROC) curve) of 0.6725 on RPT and 0.7314 on RDN. Their final system was able to correctly detect hostage release in 70% of cases with a 20% false alarm rate.

A Hidden Markov Model (HMM) represents a probability distribution over a sequence of observations. It is a Markov model with hidden states. That is, future states depend on the current state only and the observation at time t is generated by a process whose state S_t is hidden. A Dynamic Bayesian Network (DBN) is a Bayesian network used to model time series data where arcs flow forward in time. In [17] Pattipati et al propose a model to represent an airline hijacking using HMMs to determine the state transition path, and a Bayesian network. The likelihoods of three independent HMMs representing the planning and strategy, resource allocation and preparations for hijacking are associated with boolean Bayesian nodes. The probabilities received from the HMMs are combined in order to provide the likelihood of a terrorist attack and the Bayesian network is updated only when the HMMs report significant findings. The model was tested on a single example, the case of Indian Airlines Hijacking. Figures 3a and 3b show their results.



(a) Detection of HMMs using CUSUM (cumulative sum control chart) test statistic, used to monitor change detection. That is, when the probability distribution of the time series changes. A terrorist activity is assumed with the starting point of each HMM. The peak probability represents the last state of the HMM [17].



(b) Probability of hijack for Indian Airlines. The Bayesian inference is associated to the detection of the HMM [17].

Figure 3: HMM and Bayesian inference.

In [51] HMMs represent different terrorist activities and these are associated to a subordinate Dynamic Bayesian network. The final probability of terrorist activity is a Dynamic Bayesian network applied to the subordinate networks. Results are shown in Figure 4.

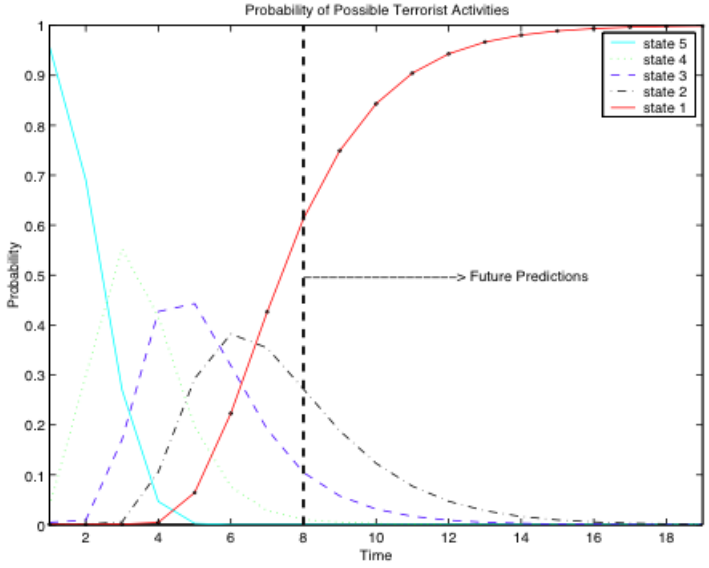


Figure 4: Probability of five state network HMM. Only one state is predicted to be a terrorist attack [51].

A novel approach was presented in the paper [52]. Baumgartner et al modelled a Bayesian network as an inference algorithm, providing confidence levels to represent the accuracy for each feature. Their Bayesian Network model was trained on data containing 57 binary features identified by criminal investigators and forensic psychologists and tested on 1,000 single-victim homicide cases with 21 features. The reported overall accuracy is impressive, reaching 80% in the single-victim homicide cases. Their simulation tests show that their use of confidence levels, that is considering only predictions with high confidence levels, increases the accuracy to 95.6%. In their work, the authors proposed a new approach based on Markov separation properties for inhibiting an arc in the network structure. K2' as they name it, uses these properties to simplify the structure of the network by inhibiting an arc between the evidence variables. When all variables are instantiated, a Bayesian Network is equivalent to one in which the arcs between the evidence variables are removed. The inference results will be the same even if the evidence variables are not independent. The performance of K2' in comparison with K2 can be seen in Figure 5.

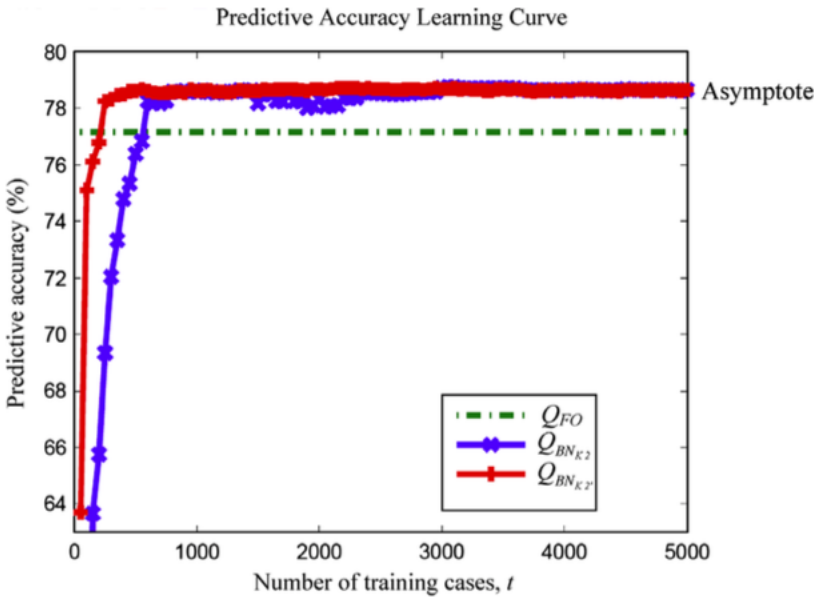


Figure 5: Accuracy for K2 and K2' algorithms where Q_{FO} is the average accuracy. On small data sets, t $<$ 800, K2' performs better than K2 [52].

In [53] the authors developed a three layer Bayesian network to calculate the threat of an attack. The three layers consist of a physical layer, comprising information such as organization, target, attack, weapon etc, a social layer with psychological and political motivations to join a terrorist group, and an economic layer. The social and economic layers are split into further layers representing initial indicators, exclusion factors, personal indicators and predictive behaviour, constructed in a way so that they can be analyzed in this order by experts. The authors identified 25 nodes and the conditional probability tables using experts' opinions from an initial set of 70 features. While the structure of the

network can be derived from expert knowledge, learning the conditional probability tables from a panel of experts may result in parameters that do not represent the true relations between features. The model was built in Matlab [1] using the Bayes Net Toolbox (BNT) [2] and the threat anticipation model was tested on three cases: when the likelihood and the consequence of attack are high, the threat is high at 97.347%, when both are low the threat is low at 6.634% and when the likelihood is low and the consequence is high, the threat of attack is moderate at 50.797%.

A similar approach is followed in [18] where a Dynamic Bayesian network is used to predict the likelihood of terrorist activities at critical transportation facilities. The probabilities are derived from FBI and immigration databases and the model is tested on two possible examples of an airplane hijack at an U.S. airport. The authors constructed a hypothetical scenario for which the likelihood of a terrorist event was 74%. If a hidden layer is activated, the likelihood of the terrorist event drops to 39%. They constructed a second example which contains information about three suspected individuals observed over n discrete time variables. For a time interval of 2h and n = 10, the likelihood of a terrorist event is 56%.

Reviewing the literature, there are no previous works aimed directly at analyzing the behaviour characteristics of lone wolf terrorists with respect to targets using a Machine Learning approach. Even though the paper by Baumgartner et al [52] follows a similar approach by constructing a Bayesian network for single-victim homicides, we use a different corpus given that our aim is to explore the terrorism and the extremist behaviour.

[1] http://uk.mathworks.com/products/matlab/
[2] https://code.google.com/p/bnt/

4

Framework overview

In this section we give information about the data set used in the project and explain the main components of our framework by giving a high level overview of the main parts. We review the tools used in implementing the framework.

1 Data set

This model is built using a data set of lone actor terrorists, who planned or organized and carried out an attack and were either convicted of their crimes or died during the attack in the U.S. and Europe since 1990. Additionaly, the data includes individuals who engaged in non-violent behaviours that facilitated attacks carried out by others. The data was gathered by researchers at the International Center for the Study of Terrorism (ICST) at the Pennsylvania State University (PSU). A data set of 111 lone actor terrorists with more than 180 variables is provided by Dr. Paul Gill from University College London. Whilst this may appear as a small data set, lone-wolf terrorism and terrorism in general are domains in which additional data cannot be easily acquired.

The variables analysed in the original data set include socio-demographic information (age, gender, employment, education, family characteristics etc), antecedent event behaviour (pre-event warning or changes in ideology and religion beliefs), event specific behaviour (attack methods), network variables (whether the individual interacted with members from a network or tried to recruit other members) and post-event behaviours (claim of responsibility). The information was collected as reported in the media. Since details about specific attacks varied across the media, the authors faced difficulties in distinguishing between missing data and an absent value [8].

1.1 Restructuring the data set

A data set of 111 lone actor terrorists with more than 180 variables is provided by Dr. Paul Gill from University College London. Firstly we restructure the data set by:

(a) eliminating variables that are not relevant for the current purposes of the project such as: age of first terrorist activity or the size of the town in which the attack took place.

(b) eliminating variables related to time such as: the time difference between planning the attack and execution of attack.

(c) extracting features by creating new variables from the current features if related. For example three binary variables (previous criminal conviction, juvenile arrest, and previously imprisoned) are combined to create a new discrete feature, criminal conviction.

(d) certain variables were coded as unknown in the original data set. In our model, the absence of a binary behaviour characteristic is encoded as a missing value (unknown = no).

1.2 Dividing the feature set into themes

In this project, we divide the complete set of features into three categories: ideology, network and mental illness. The themes identified and used throughout this project are guided by the themes identified in the criminology literature [21].

- Ideology contains variables such as: ideology, religion, whether other people were aware of the individual's ideology/religion, if the offender produced propaganda material etc.
- Mentall illness is composed of features representing the substance abuse and use, insanity, whether the individual was isolated etc.
- Network includes funds, whether the individual was involved or interacted with an organized group or tried to recruit other members etc.
- The rest of features that cannot be categorized such as level of education, employment etc, are included in each of the data set for the three themes.

1.3 Targets

A high value target is represented by a target required for the successful completion of the mission [60]. In this project we define a high value target as a person or institution that has an important role in society. We consider military, political targets, government buildings

etc to be a high value target. We define all other individuals and institutions to be civilian. This includes but it is not limited to: social security offices, newspapers publishing houses, companies, ethnic minorities, doctors.

2 Algorithm overview

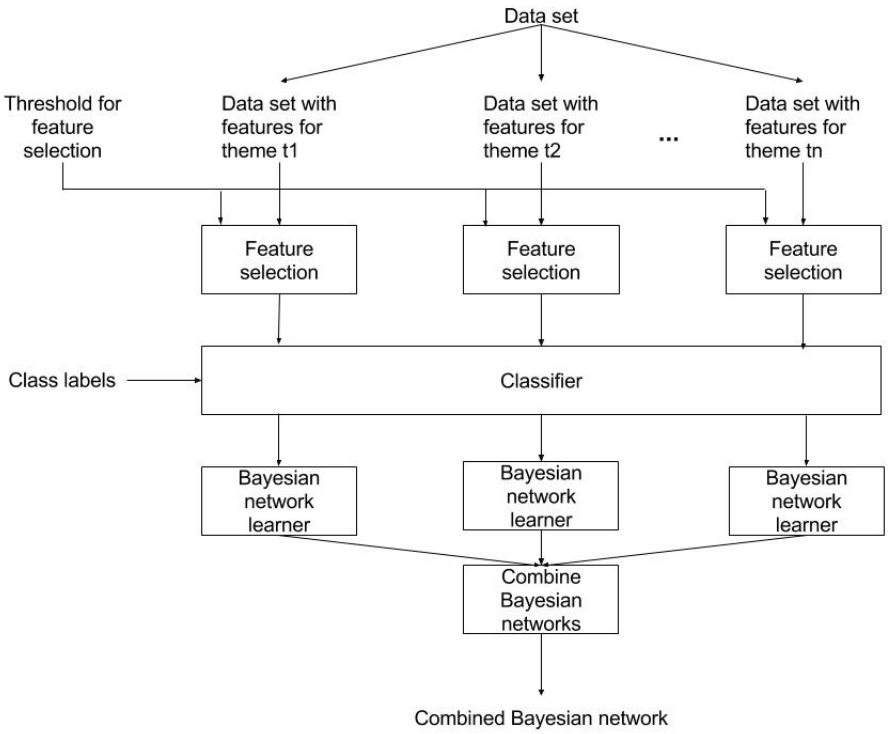


Figure 1: Framework overview.

Figure 1 shows the main parts of our framework. The feature set is split into several themes, with the same number of examples being used in all experiments. To determine the informative feature set, we implemented a feature selection algorithm based on information gain of feature pairs. We select the most informative features based on a predefined threshold. We apply feature selection in each theme and combine the individual outputs using two methods: Majority voting and Weighted Majority voting. The classifier tests the *combined* outputs of themes and not the individual accuracies of each theme, hence the same threshold is used in all categories to select the features. The performance of the various combinations of thresholds, machine learning algorithms and fusion methods are then compared in order to identify the method that achieves the highest accuracy. The results are computed using cross validation. It is important to note that due to the class imbalance we prefer a model with a lower accuracy but higher F1 measures for the target classes. By identifying the method that yields the highest accuracy, we also determine what features were used to

achieve this result. We use these most informative features to generate a Bayesian network for each feature set corresponding to a theme. Finally, we create a single network as a combination of the thematic networks and perform inference.

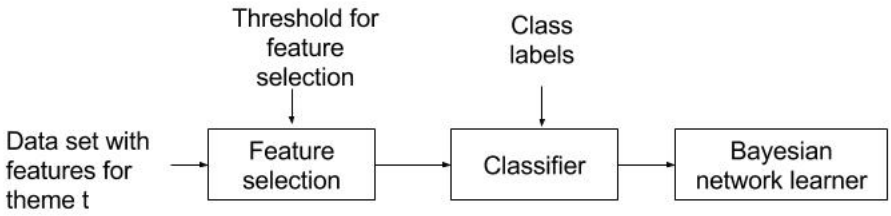


Figure 2: Method overview.

Figure 2 shows a more detailed view of how our method works on a thematic data set. We apply feature selection and determine the most informative features based on a predefined threshold. The data set created using these features is trained with a classifier, one associated to each theme. Using cross validation we determine the accuracy of the predictions. The final classifier determines the accuracy of the *combined* outputs of themes rather than the accuracy for each category. Identifying the threshold that gives the highest accuracy determines which features to be used in constructing the Bayesian network for the associated theme.

3 Collecting data

In order to test how our classification model generalises, we need to conduct our experiments on new, unseen data. We extended the current database with 10 examples since the domain of lone terrorism and terrorism in general is one in which additional data cannot be easily acquired.

The original database included socio-demographic information (age, gender, employment, education, family characteristics etc), antecedent event behaviour (pre-event warning or changes in ideology and religion beliefs), event specific behaviour (attack methods), network variables (whether the individual interacted with members from a network or tried to recruit other members) and post-event behaviours (claim of responsibility).

We gathered data based on the features from the original data set. The data was collected from databases such as murderpedia [1] and cage [2] as well as from academic and non-academic sources such as news reports. Each variable from the original feature set is checked against these sources and it must be recorded in multiple sources, not just one, in order to overcome

[1] http://murderpedia.org/
[2] http://www.cageuk.org/

source reliability issues. Where no information can be found about a variable, we assign a no (equivalent to 0) in the case of a binary feature or unknown in the case of a categorical feature. It is important to note that the quality of the data is highly dependant on the media reporting.

4 Tools

The programming language used throughout the project is Python, along with scikit-learn, a machine learning library in Python and Bayesian network libraries, libpgm and Ebay BBN. We also used IBM SPSS Statistics, a package generally used in social science.

Python
Python [3] provides both object oriented and functional features. Python code can be easily read and understood. NumPy [4] is a scientific package for Python and provides efficient numerical computations, fast operations on arrays, as well as basic statistical operations. Python is a production ready tool which represents an advantage for building an interactive interface.

Scikit-learn
Scikit-learn [5] is a machine learning tool written in Python, used for data analysis and built on libraries such as NumPy, SciPy [6], and matplotlib [7]. It provides various supervised learning algorithms and parameter tuning models and it uses C-libraries to enhance performance.

ligpgm
libpgm [8] is a Python library for creating a Bayesian network, learning parameters from data and performing inference. The algorithms are based on Koller and Friedman's *Probabilistic graphical models* [61]. The library can be used to learn Bayesian networks (structures and parameters) from data, to learn the structure of a discrete Bayesian network from data, to learn the conditional probability tables of a discrete Bayesian network given data and a structure and to perform Gibbs sampling in a discrete network given evidence.

Ebay BBN
Ebay BBN [9] is a Python library for creation and inference on Bayesian networks. It supports conversion to join trees and exact inference on cyclic graphs. It does not however support learning, hence we combined libpgm with Ebay BBN to handle network parameters learning and cyclic graphs.

[3] https://www.python.org/
[4] http://www.numpy.org/
[5] http://scikit-learn.org/stable/
[6] http://www.scipy.org/
[7] http://matplotlib.org/
[8] http://pythonhosted.org/libpgm/
[9] https://github.com/eBay/bayesian-belief-networks

D3

D3 (Data Driven Documents) [10] is a JavaScript library for producing visualisations and manipulating data based documents. It has a powerful visualisation API which makes it easy to bind the data. It takes advantage of JavaScript, HTML, CSS and SVG. The graphs created using D3 can be both static and interactive.

Flask

Flask [11] is a microframework designed for Python and is based on Werkzeug, a WSGI utility library for Python and template engine Jinja2. It is a good choice for small web applications.

highcharts

Highcharts [12] is a JavaScript charting library for producing interactive charts. It supports a wide range of visualisations such as column charts, bar charts and pie charts.

SPSS

SPSS is the most widely used package in social science for quantitative data analysis. It can perform data manipulation, analysis and has a vast number of statistical and mathematical functions. SPSS provides Multidimensional Scaling, which is used in order to find the structure in a data set consisting of distance measures between variables or cases.

[10] http://d3js.org/
[11] http://flask.pocoo.org/
[12] http://www.highcharts.com/

5

Feature selection

When using a filtering or a feature ranking algorithm, certain features are considered redundant and are dropped from the learning algorithm. However it might be the case that the variable that was previously identified as redundant by a feature selection or ranking algorithm can actually improve the accuracy of a classification algorithm if added to the feature set.

Having many variables makes it difficult to fully understand the data. The shortcoming is attributed to the complexity of behaviour and to the large number of variables, which limit the applicability of behaviour classification. We are interested in assessing the influence that features have on the model.

In our model, we select the most important features before classification, that is feature selection is independent of the prediction algorithm. The approach consists of identifying the most informative features using information gain ratio and implementing a model, using various techniques, to identify terrorist target type. To generalise the model, we split the data set into ten folds where each fold has a set of features. After identifying the most informative variables, features that appear in at least N% (50%, 70%, 90%) folds are selected. Using the attributes obtained after feature selection we perform terrorist target classification and determine which threshold for the set of features yields the best results.

1 Feature selection algorithm

Feature selection is applied in order to reduce the number of variables, but without losing the information that the original variables provide. The main advantages of feature selection are: improving the generalisation error, determining the relevant features which can be used for explanatory purposes, and reduce the dimensionality of the input space.

Feature selection is applied in order to eliminate low quality features from the data set. It can lead to an improvement in the classification accuracy as well as to an improvement in the computational efficiency. The main reason for choosing feature selection over other techniques such as dimensionality reduction is feature interpretability, which is required in the field of criminology. In dimensionality reduction, the linear combinations of the original features are usually not easily interpretable.

Gain ratio

In order to reduce the bias resulting from the information gain of variables with a large number of states, we use the gain ratio which adjusts the information gain to allow uniformity of the feature values [62]. The intrinsic value depends on the number of values a categorical feature has and how uniformly distributed these values are. The higher the value of the intrinsic value, the lower the gain ratio. Dividing the information gain by the intrinsic value reduces the bias towards selecting attributes with a large number of values.

We split the feature set into the selected themes: ideology, network, mental illness. The features that cannot be categorized such as level of education, employment etc, are included in each of the data set for the three themes.

The number of features after restructuring the attribute set and the number of features in each of the three categories is illustrated below:

Category	Number of features
Entire feature set	64
Network	35
Mental Illness	40
Ideology	48

1.1 Feature selection using pairs of features

There are methods that provide good classification performance, unfortunately they do not allow for a deeper understanding in the way the decision is made. Whilst some methods apply data transformation into a different dimensional space, they do not identify the features that are the most important for determining the target.

1.1.1 Pairwise Feature selection algorithm

The information gain ratio describes the relation between each feature and the target, and the main advantage of feature selection is that the information about the importance of individual attributes is not lost. However, in this approach, the information provided by pairs of features is ignored and it can lead to the case where two variables that are less informative by themselves can be useful when taken together. We test whether a variable that is less informative by itself can provide a significant performance improvement with respect to classification accuracy when taken with others. Thus, two features that are not as informative by themselves can be useful together. We consider pairs of features and create a new feature from these by joining them, each value of the new feature corresponding to a unique combination of the original feature values. For example, if variable X has values $\{x_1, x_2\}$ and variable Y has values $\{y_1, y_2\}$ then the derived feature has values $\{x_1y_1, x_1y_2, x_2y_1, x_2y_2\}$. The derived feature is considered for feature selection if its information gain ratio is higher than the information gain ratio of both of its constituent features. The final derived features selected are the ones whose maximum difference between the information gain ratio of the derived feature and its constituent factors is greater than the average difference across all derived features. In the final step, we select the most informative N derived features based on a previously defined threshold: 50%, 70% and 90% respectively. We compare multiple thresholds to determine whether the number of features selected has an impact on the classification accuracy in which these features are used. While in the algorithm, a pair of features is replaced by one feature, the final set of attributes will include the constituent features of the selected attributes.

We split the data set into ten folds. In each fold, we calculate the information gain ratio of each variable and the information gain ratio of the derived variable obtained as a pairwise combination of features from the original feature set. From the newly derived variables, we choose the features whose information gain ratio is higher than the ones of its constituent factors. We assign to each of these features the maximum difference between its information gain ratio and the ones of its constituent factors. We assign the maximum difference as we want the derived variable to have the greatest increase in information when compared with the constituent factors. We calculate the average of these values, `avg_increase_difference`, and we select the derived variables whose difference between its value and the information gain ratio of its constituent features is greater than `avg_increase_difference`. The set of features is given by the constituent factors and the final set of features is obtained by choosing the features that appear in at least N% folds.

Algorithm 1: Feature selection algorithm

input : All features: $features$

output: Most informative features: $informativeFeatures$

```
for i ← 1 to 10 do
    featuresGain ← ∅;
    for feature ∈ features do
        featuresGain[feature] ← InformationGainRatio(feature);
    informativeFeatures ← ∅;
    derivedFeaturesGain ← ∅;
    jointFeatures = combinationsOfFeatures(features);
    for derivedFeature ∈ jointFeatures do
        derivedFeaturesGain[derivedFeature] ←
        InformationGainRatio(derivedFeature);
    potentialInformativeFeatures ← ∅;
    for derivedFeature ∈ jointFeatures do
        if checkDerivedFeatureContext(derivedFeature) then
            // choose the features whose information gain ratio is higher that the ones of
            its constituent factors
            potentialInformativeFeatures[constituentFeatures] ←
            max(derivedFeaturesGain[derivedFeature] −
            featuresGain[constituentFeature1], derivedFeaturesGain[derivedFeature] −
            featuresGain[constituentFeature2];
    avgIncreaseDifference ← average(potentialInformativeFeatures);
    for derivedFeature ∈ potentialInformativeFeatures do
        if potentialInformativeFeatures[derivedFeature] −
        featuresGain[constituentFeature1] > avgIncreaseDifference and
        potentialInformativeFeatures[derivedFeature] −
        featuresGain[constituentFeature2] > avgIncreaseDifference then
            informativeFeatures ← constituentFeatures
Final set of features is obtained by choosing the features that appear in at least N% folds.
```

1.1.2 Most informative features

The following plots represent the most informative features from the entire feature set as well as from each thematic set of variables before the final step of the pairwise algorithm: choosing the features that appear in at least N% folds. It is not necessary the case that the union of the selected features in the three themes is the same as the set of most informative variables when feature selection is applied on the entire feature set.

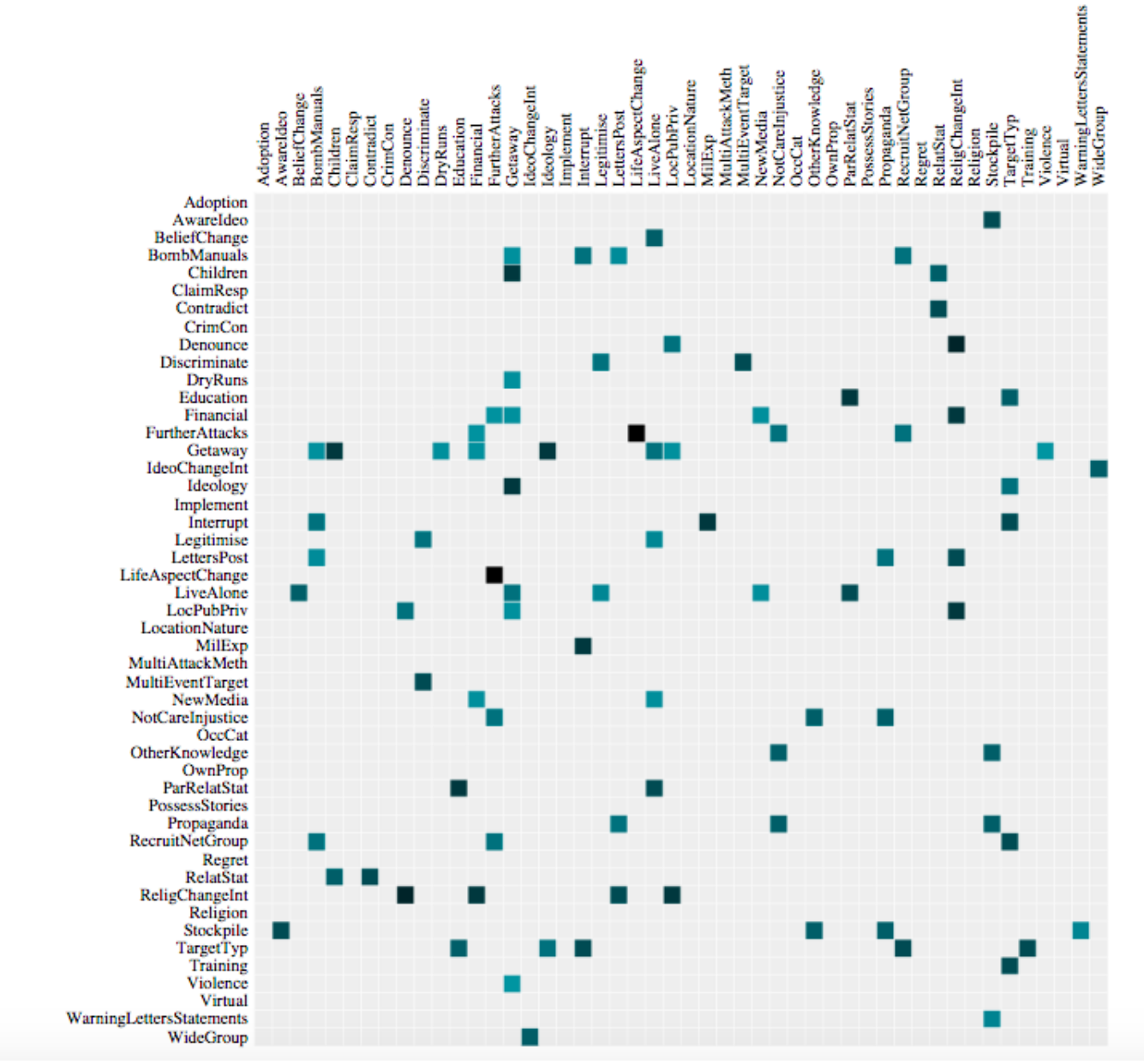


Figure 1: Feature matrix for ideology variables. Colour intensity is proportional to the gain in the difference between the derived feature and its constituent variables.

Listing 5.1: Initial set of most relevant features from the ideology feature set

```
python feature\ context/join_attributes.py
ideo
set(['MilExp', 'Discriminate', 'OtherKnowledge', 'FurtherAttacks', 'Training', '
    Religion', 'AwareIdeo', 'MultiEventTarget', 'RelatStat', 'Denounce', '
    IdeoChangeInt', 'Interrupt', 'LifeAspectChange', 'WarningLettersStatements', '
    LettersPost', 'Violence', 'BombManuals', 'Contradict', 'ParRelatStat', '
    Legitimise', 'Getaway', 'Ideology', 'Stockpile', 'RecruitNetGroup', 'Propaganda',
     'NewMedia', 'Financial', 'LocPubPriv', 'LiveAlone', 'WideGroup', '
    NotCareInjustice', 'ReligChangeInt', 'TargetTyp', 'DryRuns', 'Education', '
    Children', 'BeliefChange'])
Nr selected features 37
Nr total features 48
Features eliminated set(['Regret', 'CrimCon', 'OccCat', 'OwnProp', 'Virtual', '
    LocationNature', 'ClaimResp', 'Adoption', 'PossessStories', 'MultiAttackMeth', '
    Implement'])
```

Listing 5.2: Initial set of most relevant features from the entire feature set

```
python feature\ context/join_attributes.py
all
set(['MilExp', 'Discriminate', 'InteractNet', 'OtherKnowledge', 'FurtherAttacks', '
    Isolated', 'Religion', 'OtherInv', 'AwareIdeo', 'MultiAttackMeth', 'Insanity', '
    Obsess', 'Stress', 'MultiEventTarget', 'RelatStat', 'Denounce', 'IdeoChangeInt',
    'ClaimResp', 'Interrupt', 'LifeAspectChange', 'WarningLettersStatements', '
    MentalIll', 'LettersPost', 'Violence', 'Training', 'BombManuals', 'Contradict', '
    ParRelatStat', 'Legitimise', 'Getaway', 'Ideology', 'HurtOthers', 'DryRuns', '
    Stockpile', 'RecruitNetGroup', 'Implement', 'Propaganda', 'SubstanceUse', '
    NewMedia', 'Financial', 'LocPubPriv', 'LiveAlone', 'CrimCon', 'WideGroup', '
    NotCareInjustice', 'Involve', 'ReligChangeInt', 'TargetTyp', 'AwareGriev', '
    Education', 'Children', 'BeliefChange'])
Nr selected features 52
Nr total features 64
Features eliminated set(['Regret', 'OccCat', 'OwnProp', 'Virtual', 'SubAbuse', '
    LocationNature', 'Adoption', 'PossessStories', 'Funds', 'HarmVictimHelpless', '
    PersRelat', 'Tipping'])
```

Figure 2: Feature matrix for all variables. Colour intensity is proportional to the gain in the difference between the derived feature and its constituent variables.

Listing 5.3: Initial set of most relevant features from the network feature set

```
python feature\ context/join_attributes.py
net
set(['Discriminate', 'InteractNet', 'NotCareInjustice', 'FurtherAttacks', '
    MultiAttackMeth', 'MultiEventTarget', 'RelatStat', 'DryRuns', 'ClaimResp', '
    Interrupt', 'LifeAspectChange', 'WarningLettersStatements', 'LettersPost', '
    Violence', 'BombManuals', 'ParRelatStat', 'Getaway', 'Stockpile', '
    RecruitNetGroup', 'Implement', 'Financial', 'LiveAlone', 'CrimCon', '
    OtherKnowledge', 'Involve', 'NewMedia', 'TargetTyp', 'Education', 'Children'])
Nr selected features 29
Nr total features 35
Features eliminated set(['Regret', 'Funds', 'Virtual', 'OtherInv', 'OccCat', '
    PersRelat'])
```

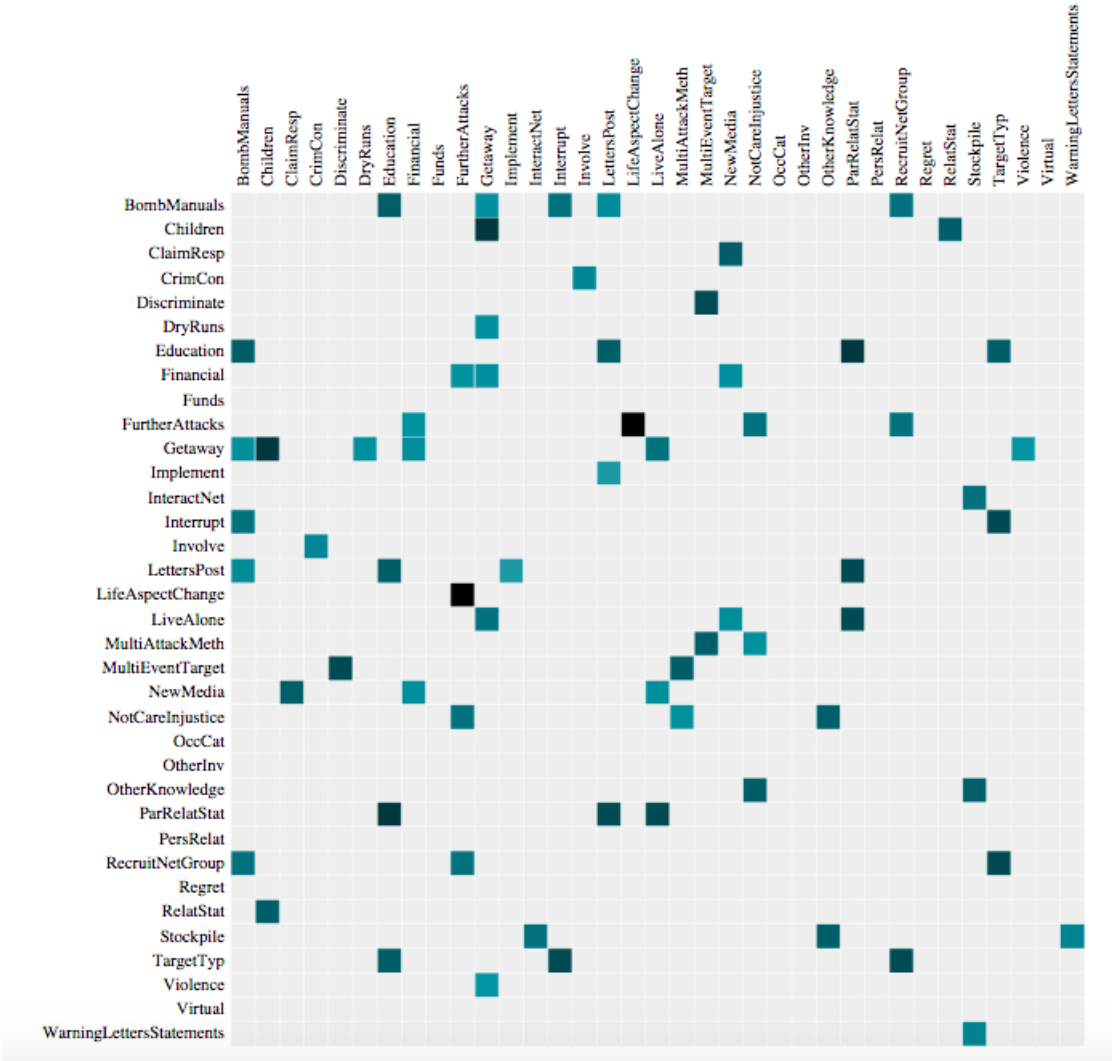


Figure 3: Feature matrix for network variables. Colour intensity is proportional to the gain in the difference between the derived feature and its constituent variables.

Listing 5.4: Initial set of most relevant features from the mental illness feature set

```
python feature\ context/join_attributes.py
ill
set(['Discriminate', 'NotCareInjustice', 'FurtherAttacks', 'Isolated', '
    MultiAttackMeth', 'Stress', 'MultiEventTarget', 'RelatStat', 'DryRuns', 'Insanity
    ', 'ClaimResp', 'HurtOthers', 'LifeAspectChange', 'WarningLettersStatements', '
    MentalIll', 'LettersPost', 'Violence', 'BombManuals', 'ParRelatStat', 'Getaway',
    'Interrupt', 'Stockpile', 'Implement', 'Tipping', 'SubstanceUse', 'Financial', '
    LiveAlone', 'OtherKnowledge', 'NewMedia', 'TargetTyp', 'AwareGriev', 'Education',
     'Children'])
Nr selected features 33
Nr total features 40
Features eliminated set(['Regret', 'CrimCon', 'OccCat', 'Virtual', 'SubAbuse', '
    HarmVictimHelpless', 'Obsess'])
```

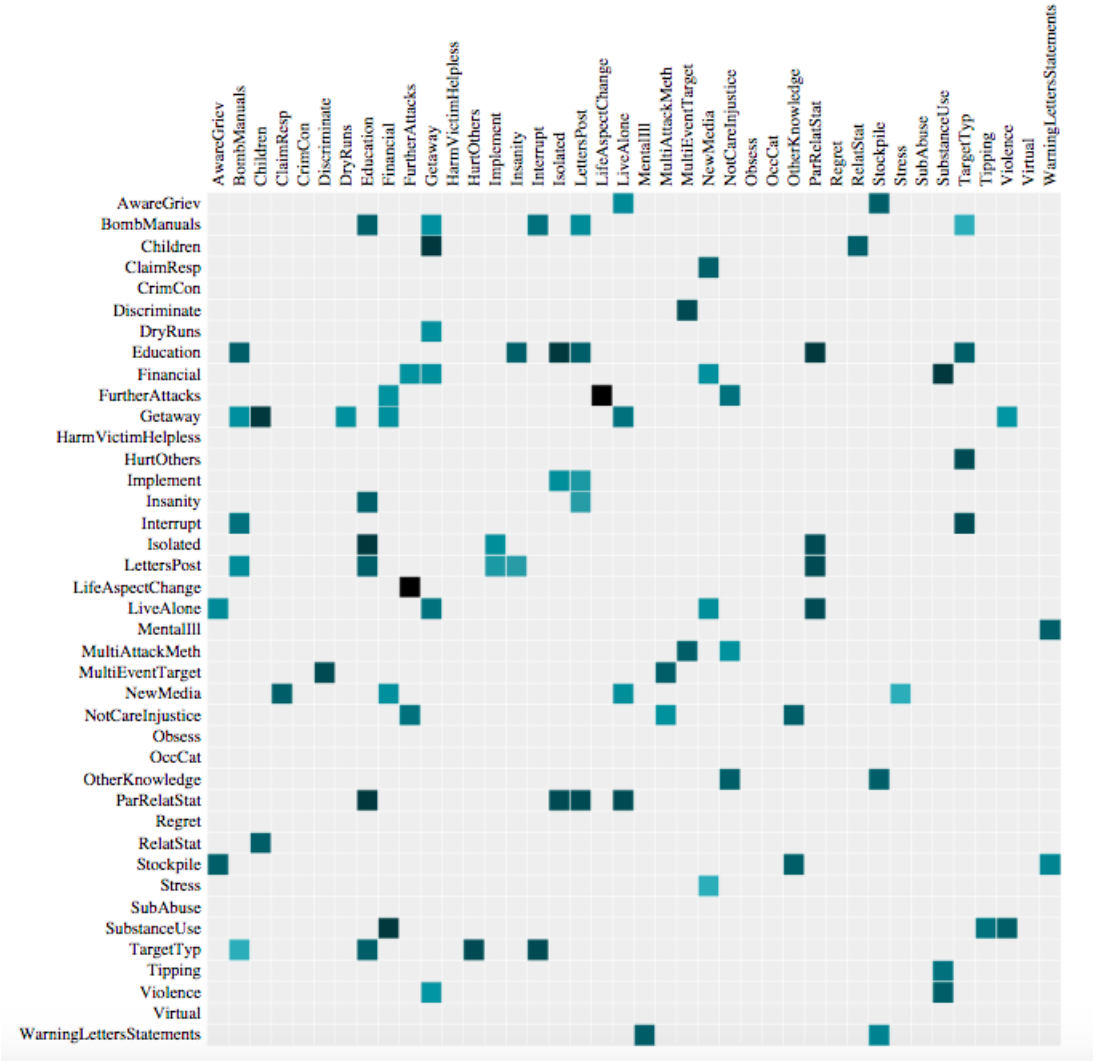


Figure 4: Feature matrix for mental illness variables. Colour intensity is proportional to the gain in the difference between the derived feature and its constituent variables.

The number of the most informative features can be found in the following table:

Data set	Nr Features 90% folds	Nr Features 70% folds	Nr Features 50% folds
Entire data set	40	50	53
Network	19	24	28
Ideology	25	34	39
Illness	20	30	33

1.1.3 Computation time

The feature selection algorithm evaluates all pairwise combinations of features. It is an exhaustive search, quadratic in the number of features $O(n^2)$. Compared to other methods, the pairwise feature selection algorithm has an additional complexity by estimating the information gain of the derived features. The computation can be parallelised since the evaluations are independent of each other.

2 Feature selection with Proxscal

2.1 Proxscal

The most widely used package in social science for analysis of quantitative data is IBM SPSS Statistics. It can perform data manipulation, analysis and has a vast number of statistical and mathematical functions.

SPSS provides Multidimensional Scaling, which is used in order to find the structure in a data set consisting of distance measures between variables or cases. This is done by finding the least squares representation of points in a low dimensional space so that the distances between points represent the given dissimilarities as closely as possible. It can be viewed as an exploratory data analysis. The output is represented as a map in which clusters can represent a subdomain of the problem. One of the advantages of MDS is that it can handle nonlinear relationships between attributes. Generally, the MDS maps are used in criminology to provide a way of identifying behaviours that occur together in crimes.

PROXSCAL(proximity scaling) represents the Categories Multidimensional Scaling. The difference between PROXSCAL and other procedures is that it minimises the normalised raw stress rather than the S stress. The normalised raw stress measure is generally preferred because it is based on the distances rather than on the squared distances.

The main limitations of the MDS model is that it cannot be used to test hypotheses and whilst it provides a useful representation, identifying clusters is the result of subjective judgement. The main drawback of descriptive statistics is that it only reports observations,

but does not correlate or associate the data and conclusions cannot be drawn beyond the available data set. The aim of the inference is to generalise beyond the available data.

2.2 Using Proxscal for selecting features

We divide the data set into two categories based on the target choice: high value and civilian. We use PROXSCAL for the high value data set and civilian data set respectively to determine the closest points to this target. The main purpose of Multidimensional Scaling is to provide a visual representation of similarities between the data points or to identify clusters in the data set. The interpretation of a Multidimensional Scaling map is that variables that are close are considered to be related, whereas variables that are placed far from each other are considered to be different.

We use MDS, a data analysis technique, to create a map representation from which we can choose the most informative features. MDS is used to represent the data, and the typical number of dimensions chosen is two. Having three or more dimensions makes the output more difficult to analyse. We define a scaling model in SPSS, specifying the minimum and maximum number of dimensions (2 in this case) and the proximity matrix. To generate the proximity matrix, we calculate the Jaccard coefficient for each pair of variables, the standard proximity measure used in criminal behaviour studies. The Jaccard coefficient represents the level of association between two variables and is calculated as the number of co-occurences over the number of times at least one variable occurs [63].

Given two arrays of variables, X and Y, the Jaccard coefficient is defined as:

$$J = \frac{M_{11}}{M_{11} + M_{01} + M_{10}} \tag{1}$$

where M11 represents the number of examples where X and Y are both present (both have value 1), M01 represents the number of examples where X has value 0 and Y has value 1 and M10 represents the number of examples where X has value 1 and Y has value 0.

The Jaccard distance is:

$$J = 1 - J_{coefficient} = \frac{M_{01} + M_{10}}{M_{11} + M_{01} + M_{10}} \tag{2}$$

Contrary to other methods of measuring associations such as Pearson's correlation coefficient, the Jaccard index does not assign the same value to joint non-occurrences and to joint occurrences by ignoring the cases when neither variable occurs. This is useful in criminology field since it is difficult to know whether a certain variable did not occur or was not identified.

Given our data, we need to transform all discrete variables into binary in order to represent presence or absence of each variable. We compute the similarity matrix using the Jaccard coefficient. We use the Ratio transformation so that the transformed proximities are proportional to the original proximities. For this, the SPSS input is the dissimilarity matrix which is obtained by subtracting the Jaccard index from 1. The Smallest Space Analysis output suggests that two points that are close on the map are likely to co-occur.

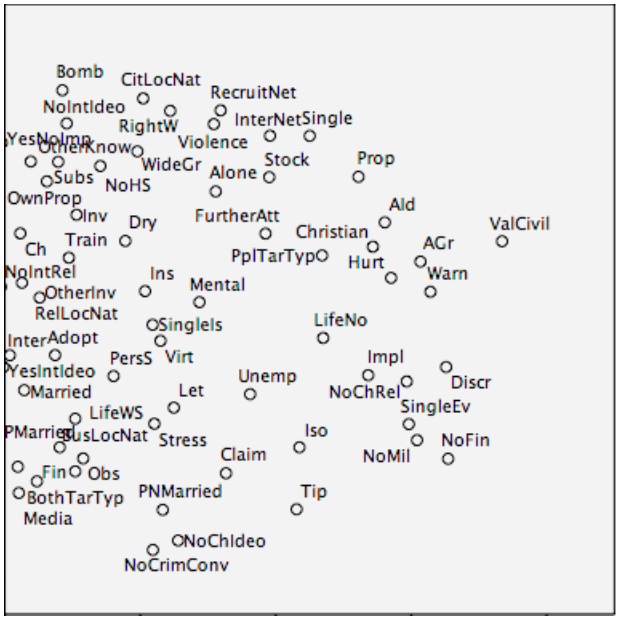
From the MDS plot, we select 70% closest features to target (high value and civilian respectively). We use each of these two sets of features in classification. We are interested in knowing whether the features selected for civilian and high value target respectively result in high precision, recall and F1 values related to their respective class.

Whilst we aim at obtaining the best representation of the data using the smallest number of possible dimensions, having only two dimensions enables readability of the MDS plot at the expense of a poor data representation.

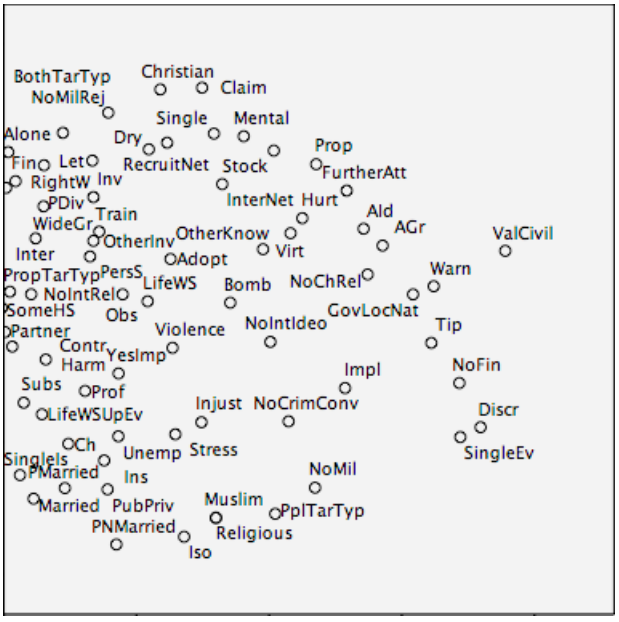
The axes from the MDS plot do not necessary represent the coordinate system and their label is interpreted by experts, which can be considered a disadvantage of the MDS model. For this project we assume that they represent the x and y axes.

2.3 Most informative features using Proxscal

The following plots represent the closest features to each of the two targets, high value and civilian, obtained using Jaccard distance. In each case, the data set used contains only the examples associated with the class under test with *ValCivil* set to 1. For example, for *civilian* case, only the examples that are labelled civilian are used in Proxscal with the feature *ValCivil* being set to 1, whereas in the *high value* case, only the examples labelled high value are used with the feature *ValCivil* being set to 1.

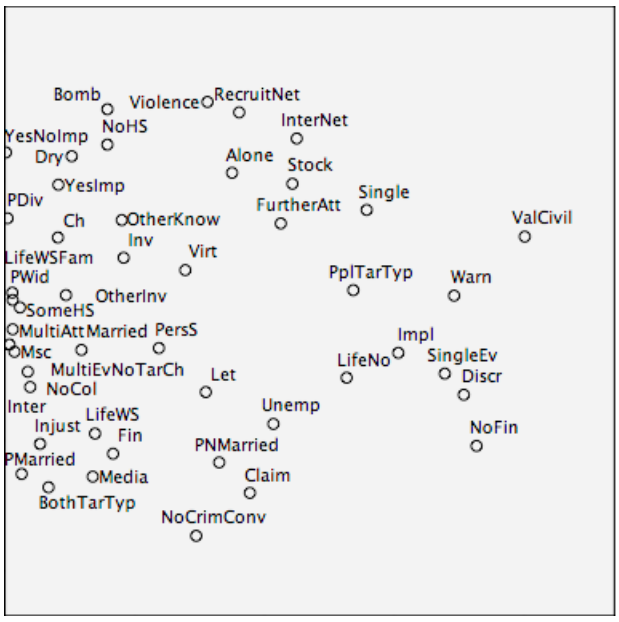


(a) Civilian all variables

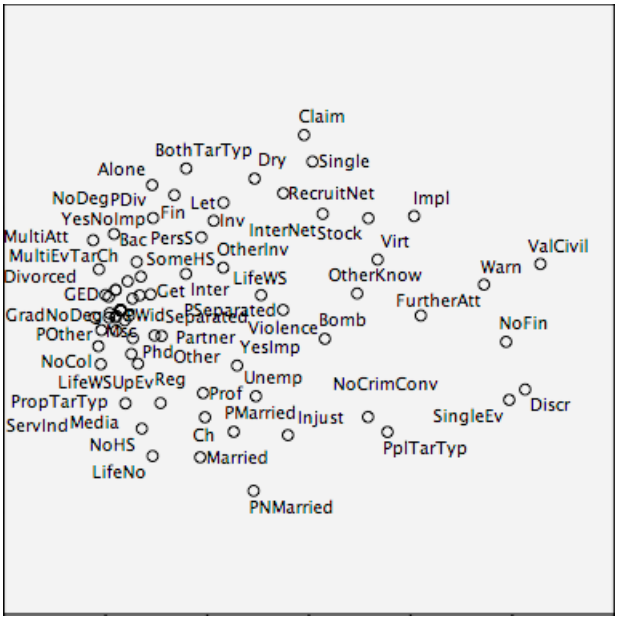


(b) High value all variables

Figure 5: All variables

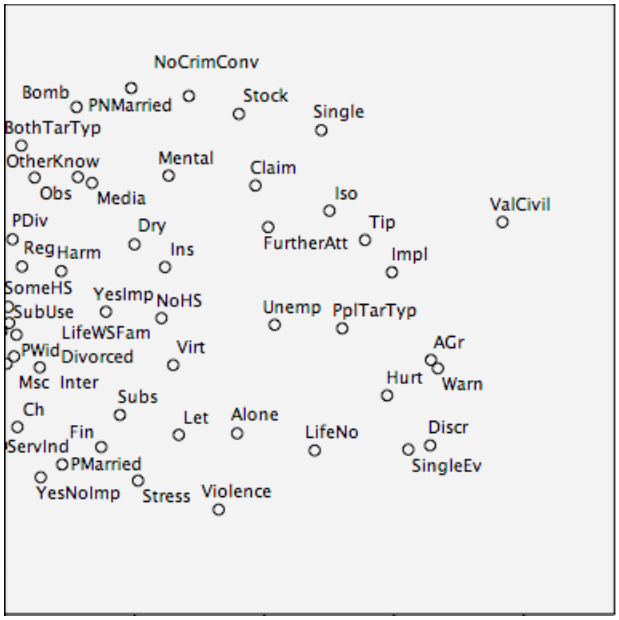


(a) Civilian network variables

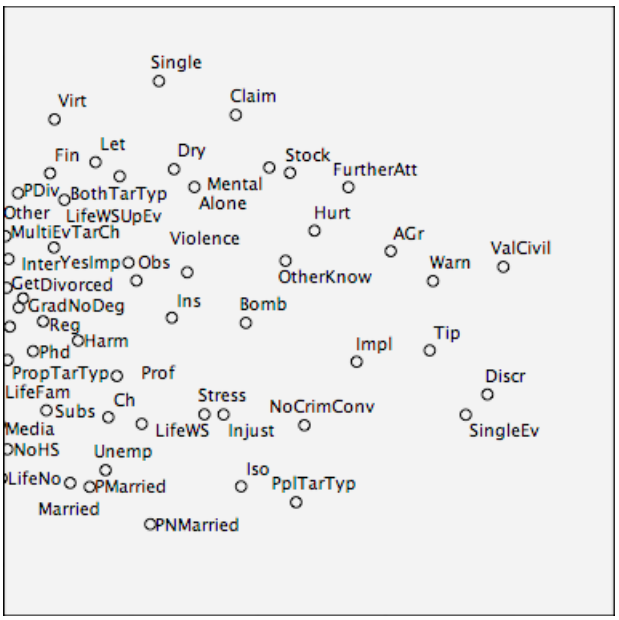


(b) High value network variables

Figure 6: Network variables

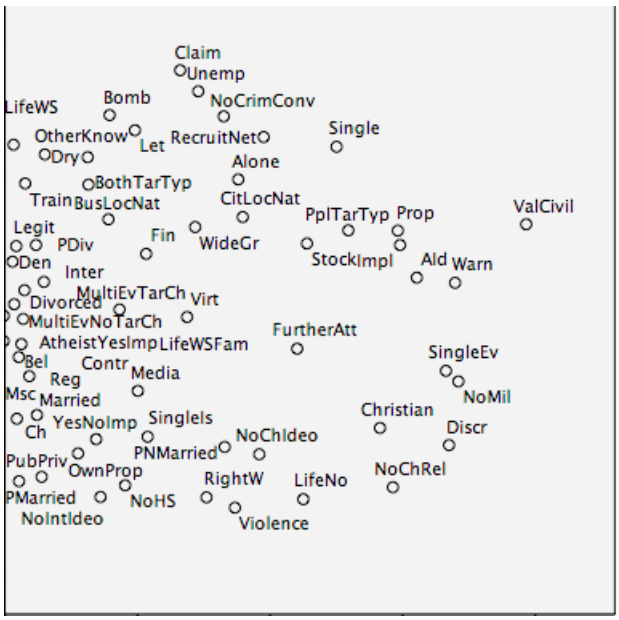


(a) Civilian mental illness variables

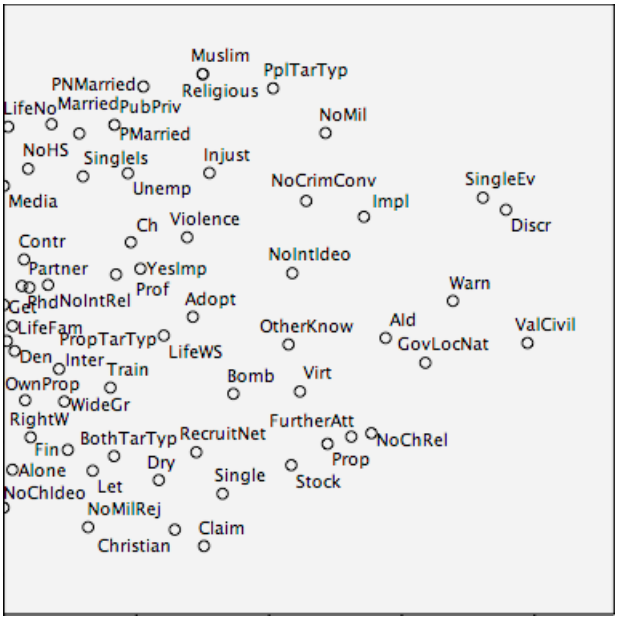


(b) High value mental illness variables

Figure 7: Mental illness variables



(a) Civilian ideology variables



(b) HIgh value ideology variables

Figure 8: Ideology variables

3 Discussion of results

The high value citizens and civilians classification model can provide a better understanding of how terrorists operate and an indication of the attributes that are correlated to each of the two outcomes. We identify differences between the proposed feature selection method and the standard approaches.

We apply the Multidimensional Scaling on the entire feature set, as well as on the sets of features related to each of the three proposed themes: network, mental illness and ideology, with respect to each of the target choices, high value and civilian, and in each case we select the 70% features closest to the target.

More than half of the features resulted from the intersection of the variable sets from both targets obtained from the SPSS analysis can be found in the attribute set identified by the proposed feature selection algorithms. From the thematic point of view, more than half of the features related to network, mental illness and ideology respectively appear in the variables set resulted from our algorithm.

The differences in the sets of features identified by our proposed method and the Smallest Space Analysis is related to the fact that we consider whether two variables that are less informative by themselves can be useful when taken together, thus analysing the information provided by pairs of features rather than considering the relation between each separate feature and the target. While analysing the importance of each individual feature, we also check whether two insignificant features by themselves can be useful together.

Variables from the entire feature set

Considering the whole feature set and comparing Figure 2 with Figure 5, our method shows that there is a potential relation between the choice of location of attack, public or private, and whether the offender had other individuals involved in procuring the weaponry, if the offender had members of the family or close associates involved in political violence, criminality or a wider movement and whether the religious beliefs of the offender have changed or intensified.

Other potential relations identified by our method are between whether the individual was living alone at the time of the event and the relationship status of the parents as well as whether other people knew about the individual's grievance prior to the event. An offender living alone at the time of the event is also likely to have experienced a change in his beliefs.

Financial issues are associated to planning further attacks as well as getaway. Whether the individual had a stockpile of weapons is linked to whether at least one other person knew about the research or preparation prior to the event as well as whether the offender recently joined a wider group, organization or movement.

Whether the ideological beliefs of the offender have changed or intensified provides more information regarding the target choice when taken together with the feature related to whether the offender joined a wider group as well as the relationship status of the parents. Isolation is associated to education and the relationship status of the parents.

Another relation identified by our method is between an individual obsessed with a specific event or phenomena and whether the individual interacted face-to-face or virtually with members of a wider network. There is a link between whether individual's close associates were involved in a movement and previous criminal conviction and a relation between dry runs and getaway.

Variables from the network feature set

Considering the features related to network and comparing Figure 3 with Figure 6, our method shows there is a link between evidence of bomb-making manuals in the individual's home and getaway. There is also a connection between the getaway and the individual engaging in dry runs. Another potential relation identified when looking at pairs of features is represented by the one between the individual providing a public claim or responsability and whether the individual became recently exposed to media. Whether the offender had other individuals involved in procuring the weaponry or had members of the family and close associates involved in political violence provides more information related to the target choice when considered together with previous criminal convictions.

Financial issues are associated to planning further attacks as well as getaway. Education is associated to the relationship status of the parents. Other potential relations identified by our method are between whether the individual was living alone at the time of the event and the relationship status of the parents as well as getaway.

Variables from the mental illness feature set

From the features related to mental illness and comparing Figure 4 with Figure 7, we identify a relation between financial issues and two other features: getaway and whether the individual was recently exposed to media. Our method also identifies a link between the relationship status of the individual's parents and the offender's isolation as well as a relation between substance use and a tipping point that precipitated the offender's movement on a pathway to engage in terrorist activities. Education is linked to isolation and whether insanity was suggested during the course of the trial.

Another potential relation is represented by the one between the individual providing a public claim or responsability and whether the individual became recently exposed to media. Our method identifies a connection between the getaway and the individual engaging in dry runs. Whether the individual had a stockpile of weapons is linked to whether at least one other person knew about the research or preparation prior to the event. There is a link between evidence of bomb-making manuals in the individual's home and getaway, the choice of target and education respectively.

Variables from the ideology feature set

Comparing Figure 1 with Figure 8, features belonging to the ideology group show that there is a relation between the choice of the location of the attack, public or private, and whether the offender had planned a getaway or if the religious beliefs of the offender have changed or intensified. The feature related to stockpile of weapons is also marked as providing more information about the target choice when considered with the feature related to whether other individuals were aware of the offender's extremist ideology.

Another potential relation identified by our method is between whether the individual sought legitimation from leading religious or political leaders prior to the event and whether the individual was living alone at the time of the event. There is a connection between the getaway and the individual engaging in dry runs and the offender's ideology respectively. An offender living alone at the time of the event is also likely to have experienced a change in his beliefs. We identify a potential relation between financial issues and whether the religious beliefs of the offender have changed or intensified. Our method suggests there is a link beetwen whether the offender received hands on training for the event and the target type.

4 Summary of Feature selection

Feature selection is applied in order to eliminate low quality features from the data set. The main reason for choosing feature selection over other techniques such as dimensionality reduction is feature interpretability, which is required in the field of criminology. Our feature selection method tests whether a variable that is less informative by itself can provide a significant performance improvement when taken with others, that is two insignificant features by themselves can be useful together. The approach consists of identifying the most informative features using information gain ratio. In order to investigate the performance of the feature selection algorithm, we generate three different sets of features based on a predefined threshold. We describe the differences identified by our method compared to using MDS plots. Using the features obtained with the feature selection algorithm we implement a model, using various techniques, to identify terrorist target type.

6

Terrorist target classification

We use the variables obtained from the feature selection algorithm described in the previous chapter in classification of terrorist target choice: high value and civilian. We then compare the results with the ones obtained when performing classification using all available features.

A supervised classification on high value/civilian target is implemented using a Support Vector Machine technique on the entire data set. The complete set of features is split into three groups: ideology, network, mental illness, and a SVM training algorithm is applied on each group. The final classification of the system combines the individual classification obtained in each group to produce the final result. The techniques used to fusion the classifiers are: Majority Voting and Weighted Majority Voting. We optimize the C and γ parameters using the RBF kernel, a common function used in SVM classification. SVM with RBF kernel will be used in all experiments in this project.

1 SVM

Support Vector Machines generalise well even in cases with limited training data. We use C Support Vector Classification, whose implementation is based on libsvm. The fit time complexity of SVC is more than quadratic with the number of samples [64]. We set the `class_weight` of the SVC function to 'auto', that is adjusting the weights of the points inversely proportional to the class frequencies. Thus, in an imbalanced data set, classes will have different importance, and the label of each class will be set to C*value.

The main advantage of SVM is that non linearly separable data can be mapped to a higher dimensional space by applying the 'kernel trick', which can then be separated by a linear hyperplane. The performance of the SVM is sensitive to the choice of kernel and parameters, but this can be overcome by performing cross validation while optimising the parameters.

1.1 Data scaling

scikit requires data sets to be standardised, meaning that the features need to be normally distributed: Gaussian with zero mean and unit variance. This is achieved using: `preprocessing.scale(X)`. This function transforms the data to center by removing the mean value of each feature, and then scales it by dividing the features by their standard deviation.

This is done because the RBF kernel assumes that features are centered around zero. If a feature has a variance that is orders of magnitude larger than others, it might dominate the function and thus the estimator will not be able to correctly learn the features. Thus we avoid the situation in which features in greater numeric ranges dominate the features in smaller numeric ranges.

1.2 Class imbalance

The main disadvantage of using only accuracy as a performance measure occurs in data sets with class imbalance. By predicting the target value of the majority class, a model can report a high classification accuracy. However this would not be useful in any problem domain. Hence, it is the case that it may be more useful to select a model with a lower accuracy that would have higher F1 scores for each class.

Whilst the goal is maximising the accuracy, machine learning algorithms applied to imbalanced data sets result in poor classifiers. The reason for this is that if 90% of the examples are from one class and under the assumption that the new instances are drawn from the same distribution as the training data, a learning algorithm that labels all instances with the majority class can achieve a 90% accuracy, but the model would not be valuable. When dealing with imbalanced data sets, a common practice is upsampling, that is replicating cases from the minority class. In our model, we upsampled the minority class in order to overcome the class imbalance problem.

1.3 Optimising parameters

The main reason for choosing a Support Vector Machine is that it can generalise small data sets and is known to be able to avoid overfitting by choosing the right kernel and tuning the kernel parameters. Splitting the initial feature set into themes and thus reducing the size of the training set results in a decrease in the complexity and computation time of the optimisation problem. The kernel parameters of each model are found based on the corresponding feature set.

1.3.1 Grid search

We optimise two parameters, C and γ using the RBF kernel, a common function used in SVM classification. Both parameters affect the model complexity. One strategy is to modify one variable at a time while having a default value for the other factor. The disadvantage of this method is that it assumes no interactions between the factors. Our approach is to use a factorial design, also known as 'grid search', in which factors are varied together.

The parameters are tuned using cross validation, we divide the data set into k folds of roughly equal size. We train a Support Vector Machine on $k-1$ folds and report the accuracy of the model on the remaining fold. This process is repeated k times. For this, we use a grid search in two dimensions. We choose the values from the interval obtained by specifying the lower and upper bound of the search space by looking at the average accuracy resulted from the cross validation technique.

In order to reduce the size of the data space we select the features and then learn the parameters of the machine learning algorithm independently.

We determine the best parameters as follows:

- grid space of (C, γ)
- for each hyperparameter pair, apply 10 fold cross validation
- choose the hyperparameter pair that leads to the lowest error
- use this best hyperparameter to create the prediction model and train the entire data set

1.3.2 Computation time

We optimize the C and γ parameters which influence the performance of the SVM. A finer tuning with a large set of parameters is achieved at the expense of computation time. For some values of γ, increasing the value of parameter C will result in equally performing models [66]. A low value of C uses less memory and less time for prediction.

The grid search evaluates all parameter combination. It is an exhaustive search through the specified hyperparameter space. It uses the cross validation technique and the best combination is selected based on the average accuracy score of the model. Each pair (C, γ) in the cartesian product of the two specified sets is used in training a Support Vector Machine for each fold. The pair that results in the highest score of the left out data in the cross validation procedure is selected. One disadvantage of this technique is that is suffers from the curse of dimensionality because it considers a large number of combinations of parameters. The computation can be parallelised since the hyperparameter evaluations are independent of each other.

Listing 6.1: Optimising parameters

```
param_grid = dict(gamma=gamma_range, C=C_range)
cv = StratifiedShuffleSplit(known_targets, random_state=42)
grid = GridSearchCV(SVC(class_weight='auto'),
param_grid=param_grid, cv=cv, scoring='accuracy')
grid.fit(dataset, known_targets)
print("The best parameters are %s with a score of %0.2f"  % (grid.best_params_, grid.
    best_score_))
```

2 Classification

Two strategies are used in performing terrorist target classification: firstly, using the data set with a single set of features and secondly, using the data set with features divided into categories. In both cases, we compare the results obtained when using all the available features with results obtained after performing feature selection as outlined in the previous section. It is important to note that in all cases the number of examples used in training is the same and that only the feature set is adjusted.

2.1 Classification using the single feature set

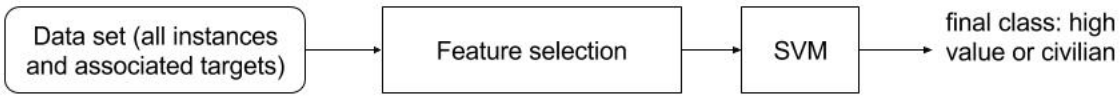


Figure 1: SVM for data set with single feature set.

We use two strategies in performing classification with the entire data set.

We train a Support Vector Machine using all the available feature set. We use the cross validation technique, dividing the data set into 10 folds, and report several performance measures. We use `StratifiedKFold`, a variation of k-fold cross validation which returns stratified folds. This means that each fold contains approximately the same percentage of examples of each of the target class as the complete training set. Hence we ensure that the distribution is respected in the training and testing sets from each fold.

We compare the results from training the models on the entire feature set with the results when a Support Vector Machine is trained on the data set obtained after selecting the most informative features using the algorithm described in the previous section. The two SVMs used on the entire feature set and on the feature set obtained after feature selection have different parameters C and γ.

2.2 Classification using themes

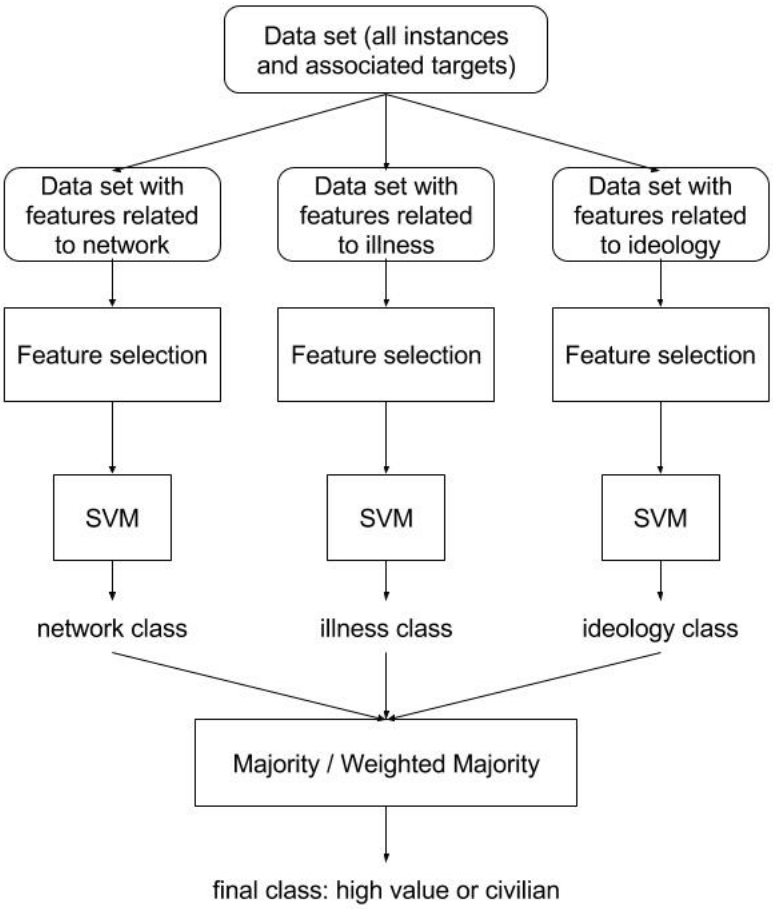


Figure 2: SVM for data set with feature set divided into the three themes.

To conduct thematic analysis we split the available data set into three themes: network, ideology and mental illness. Decomposition of the data set into several subsets yields parallel classifiers of lower complexity.

We define two tasks, one in which we train a Support Vector Machine on each theme using all the associated features and one in which we train a Support Vector Machine on the themes using the variables obtained after performing feature selection.

Each data set corresponding to a theme is trained using a Support Vector Machine. We report several performance measures using cross validation and we use `StratifiedKFold` so that each fold contains approximately the same percentage of examples of the targets as the complete training set. We combine the results of variants of the same classifier. In the case of SVM, each group of features has its own classifier with different values for the two parameters C and γ. We then perform feature selection in each of the three data sets correspoding to themes and we train a Support Vector Machine on these data sets.

In both cases, we have three outputs corresponding to the network data set, ideology data set and mental illness data set. These outputs are combined using a Majority Voting method and a Weighted Majority Voting method.

3 Fusion labels

The complete set of features is split into three groups: ideology, network, mental illness, and a Support Vector Machine algorithm is applied on each group. The features that cannot be categorized such as level of education, employment etc, are included in each of the data set for the three themes. The final classification of the system combines the individual classification obtained in each group to produce the final result.

The possible ways of combining the outputs of 3 classifiers depend on the information we obtain from individual classifiers. Xu et al distinguish three types of classifier outputs [65]. In this project we deal with the Type 1, the Abstract level. Each classifier produces a class label. Thus for any example to be classified, the 3 classifier outputs define a vector s = [s_1, s_2, s_3]. At the abstract level, there is no information about the certainty of the assigned label. The classifier outputs are assumed to be independent and the techniques used to fusion the classifiers are: Majority Voting and Weighted Majority Voting.

With **majority voting** there is no need to handle ties since there are three possible outputs and two labels. The number of classes, L, is odd (three in our case based on the themes). Thus, the class that appears twice will be assigned. For example, if two classifiers predicted the label to be high value then the final output will be high value.

In the case of classifiers not having the same accuracy, it is better to give the more performant classifier more influence in determining the final prediction. The **weighted majority algorithm** was proposed by Littlestone and Warmuth. If the classifiers in the ensemble are not of identical accuracy, we give the more competent classifiers more power in making the final decision. Initially, the weights are all 1 as there is no reason to prefer any classifier. We then compute the prediction and after each iteration, we update the weights. When a misclassification occurs, we multiply the weights of incorrect algorithm with ϵ where $0 < \epsilon < 1$. We compute the final prediction with updated weights. The final weights represent the average of the weights computed during cross validation over each theme and over a single group of features respectively.

Algorithm 2: Weighted majority algorithm

input : Outputs of 3 classifiers

output: weights

$w_i^1 \leftarrow 1$ for all i=1,...,n //weights initialised to 1;

$\epsilon > 0$;

for $r \leftarrow 1, ...n$ **do**

    f_i^r prediction of classifier i;

    $\hat{y}_r \leftarrow round\left(\frac{\sum_i w_i^r * f_i^r}{\sum_i w_i^r}\right)$

    if $f_i^r != y_r$ **then**

        $wi^{r+1} \leftarrow wi^r(1-\epsilon)$ // update weights

4 Evaluation of terrorist target classification using cross validation

The performance of the prediction models is determined by the size of the available data set. One limitation of the model is attributed to overfitting. This may appear because the underlying distribution of the training data set is undersampled. In order to overcome the overfitting problem, we apply cross validation, a technique which measures the generalisation ability of the model. Without gathering more data, there is little that can be done to overcome this problem and the domain of lone terrorism and terrorism in general is one in which additional data cannot be easily acquired.

We evaluate the performance of several learning algorithms and perform a comparative analysis. We randomly select the training and testing sets, we repeat the 10 fold cross validation process 10 times and report the average of 10x10 cross validation results. This is done in order to provide a more stable estimation of the algorithm's performance and to decrease the variations in performance caused by the choice of the training and testing sets. We report the quality of predictions using various performance measures such as: accuracy, precision, recall and F1 score. We compare the performance of the SVM trained on the single set of features with the SVMs trained on the thematic sets of features. In both cases we report the differences in performance when feature selection is applied.

Sets of features	Accuracy	High value Precision	High value Recall	High value F1	Civil Precision	Civil Recall	Civil F1
Single set	0.644	0.648	0.595	0.601	0.664	0.688	0.664
Thematic sets Majority	0.613	0.577	0.826	0.631	0.814	0.430	0.500
Thematic sets Weighted Majority	0.648	0.624	0.847	0.673	0.853	0.472	0.544
Single set 90% Feature selection	0.659	0.801	0.384	0.485	0.634	0.900	0.737
Single set 70% Feature selection	0.615	0.584	0.805	0.630	0.764	0.447	0.500
Single set 50% Feature selection	0.628	0.726	0.301	0.385	0.614	0.914	0.717
Thematic sets Majority 90%	0.653	0.733	0.478	0.541	0.650	0.807	0.708
Thematic sets 90% Weighted Majority	0.722	0.775	0.601	0.641	0.731	0.829	0.761
Thematic sets Majority 70%	0.637	0.736	0.334	0.427	0.618	0.902	0.726
Thematic sets 70% Weighted Majority	0.700	0.807	0.482	0.571	0.677	0.893	0.762
Thematic sets Majority 50%	0.613	0.673	0.256	0.340	0.599	0.926	0.717
Thematic sets 50% Weighted Majority	0.693	0.769	0.443	0.530	0.670	0.912	0.764

Table 1: Results of terrorist target classification using a single set of features with no feature selection, a single set of features with feature selection, thematic sets of features with no feature selection and thematic sets of features with feature selection.

Results show that when using a single group of features, having fewer variables results in a higher accuracy. Whilst the difference is not significant, the results show that feature selection improves the classification with 1% in the case of a single set of features if we select the features that appear in at least 90% of the folds.

In the case of thematic features, the Weighted Majority fusion algorithm shows a higher accuracy performance than the Majority voting method irrespective of whether feature selection is applied before classification. Selecting the most informative variables within thematic sets of features has a higher classification accuracy than the case in which we do not apply feature selection, the improvement varying between 5% and 8%. The best results are achieved when splitting the feature set into themes, combining the individual outputs using Weighted Majority voting and selecting the most informative features that appear in at least 90% of the folds with 72.2% classification accuracy and a precision greater than 73% for both classes, high value and civilian.

5 Summary of terrorist target classification

We performed classification of terrorist target choice, high value and civilian, using Support Vector Machines. We used two strategies in performing terrorist target classification: firstly, using the data set with a single set of features and secondly, using the data set with features divided into categories and combined the outputs using two approaches: Majority voting and Weighted Majority voting. In both cases, we compare the results obtained when using all the available features with results obtained after performing feature selection. Using cross validation, the best results are achieved when splitting the feature set into themes, combining the individual outputs using Weighted Majority voting and selecting the most informative features that appear in at least 90% of the folds with 72.2% classification accuracy.

7

Terrorist target ensemble classification

Having performed classification using Support Vector Machines known to generalise well even in cases with limited training data, we examine whether an ensemble of classifiers that includes the Support Vector Machine previously modelled can increase the classification accuracy.

We investigate various learning algorithms to determine the constituent methods of our ensemble. A supervised classification of a binary target, *high value* and *civilian*, is implemented using both a linear predictor and nonlinear predictors. We compare several classification algorithms applied to the entire feature set as well as to the set of attributes obtained after feature selection. We examine four classification algorithms in order to predict the terrorist target choice. The classifiers are Logistic Regression, Decision Tree, K Nearest Neighbours and Support Vector Machine. We compare the learning algorithms using the entire feature set to the case when we split the set of attributes into *ideology*, *network* and *mental illness*. Each data set obtained after splitting the initial examples into the three themes is processed independently by performing classification using Logistic Regression, Decision Tree, K Nearest Neighbours and Support Vector Machine respectively. Finally, the outputs from each theme are combined using Majority Voting and Weighted Majority Voting and are then used in the ensemble classification.

Integrating multiple classifiers in order to make use of the strengths of one method to complement the weaknesses of another learning algorithm, that is using an ensemble of classifiers, can have a better performance than using a single classifier. We propose a classifier ensemble based on Decision Trees, K Nearest Neighbours and SVM.

1 Ensemble Classification

The ensemble classifier consists of Decision Trees, K Nearest Neighbours and Support Vector Machines.

1.1 Support Vector Machine

Support Vector Machines generalise well even in cases with limited training data. They are known to be able to avoid overfitting by choosing the right kernel and tuning the kernel parameters. The kernel parameters of each model are found based on the corresponding feature set: single set of variables and thematic split respectively. We use the Support Vector Machines identified as the best in terms of accuracy from the previous section.

1.2 Decision Tree

The main advantages in using a decision tree are computation time and interpretability. The data does not need to be normalised since the split at each node is done based on the variable's values and thus the algorithm is invariant to transformations applied to features. The prediction complexity is logarithmic in the number of training data points. Decision trees are also easy to interpret as they can be converted to rules.

The leaves of the tree represent the two classes: high value and civilian, whilst the other nodes represent the features with a branch for each possible value. The split is performed using the entropy measure of features. To avoid overfitting, we do not expand the tree if the feature on which the split is done is not statistically significant, that is the tree expansion stops when there is no feature resulting in at least a number of examples on each branch. We achieve this in scikit by setting `min_samples_split` which represents the minimum number of samples required to split the node. In order to select the best tree, we measure the performance using cross validation over the data set.

1.3 K Nearest Neighbours

KNN algorithm stores instances of the data set rather than creating a model. Classification is given by the majority vote of the k nearest neighbours of the point currently being tested. Each feature represents a different dimension in some space and each value of the feature represents a coordinate in this space. The points are weighted using the Euclidean distance, such that closer points will have a greater influence on the point under test. The value of k is selected based on experimental results using cross validation over the data set and is set to 5.

1.4 Other methods

Logistic Regression is a discriminative linear model used in binary classification. Given a training set, the Logistic Regression algorithm learns the conditional probability distribution [67]. In our method, we make use of the L2 penalised logistic regression. The regularised logistic regression is implemented in scikit using the liblinear library. The samples of each class are weighted inversely proportional to the class frequencies by selecting `class_weight='auto'`. Whilst we tuned the parameters, the algorithm showed poor performance and the Logistic Regression method was not included in the ensemble. The results of Logistic Regression can be found in Appendix.

1.5 Ensemble overview

The classification using the ensemble of algorithms is performed on all data configurations analysed in the previous section: using a single set of features, using the thematic split of the feature set, with and without feature selection in both cases. The framework for the ensemble classifier is as follows:

1. each classifier is trained using the same training set
2. each classifier predicts each instance from the testing set
3. the predicted outputs of the classifiers are then combined to produce the final output using majority voting

1.6 Classification using the single feature set

Figure 1 shows the overview of the ensemble classifier on the data set consisting of the entire feature set. Experiments are conducted on two data sets: the initial set of features and the set of variables obtained after feature selection. We train a Decision Tree, a K Nearest Neighbours and a Support Vector Machine on the same data set. When a new instance is presented to the ensemble classifier, each learning algorithm predicts an output. The final prediction of the new example is represented by the majority class of the three proposed outputs.

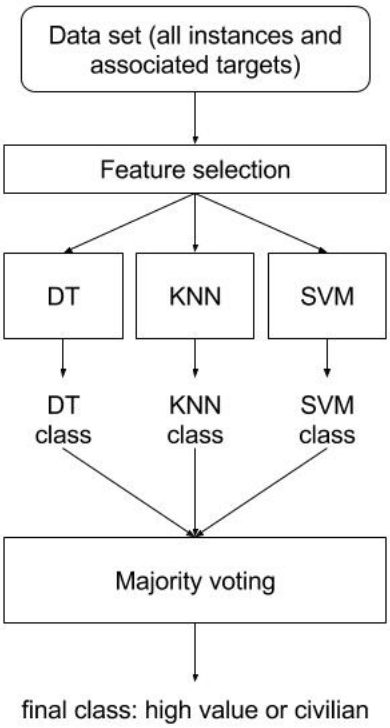


Figure 1: Classifier ensemble for data set with single feature set.

1.7 Classification using themes

Figure 2 shows the overview of the ensemble classifier on the data set consisting of the feature set split into the three themes: network, mental illness and ideology. Two data sets are used in the experiments: one in which the themes contain the original sets of features and one in which we applied feature selection in each theme. We train a Support Vector Machine on each of the three themes. When a new instance is presented to the ensemble classifier, the SVM predicts an output. The outputs related to each category are then combined with one of the two fusion algorithms: Majority Voting and Weighted Majority Voting. Using the same data sets, the same approach is applied to the other two learning algorithms: Decision Tree and K Nearest Neighbours. The same fusion algorithm is used in all cases. After this stage, there are three outputs, each resulted from one of the learning algorithms from the ensemble classifier. The final prediction of the new example is represented by the majority class of the three proposed outputs.

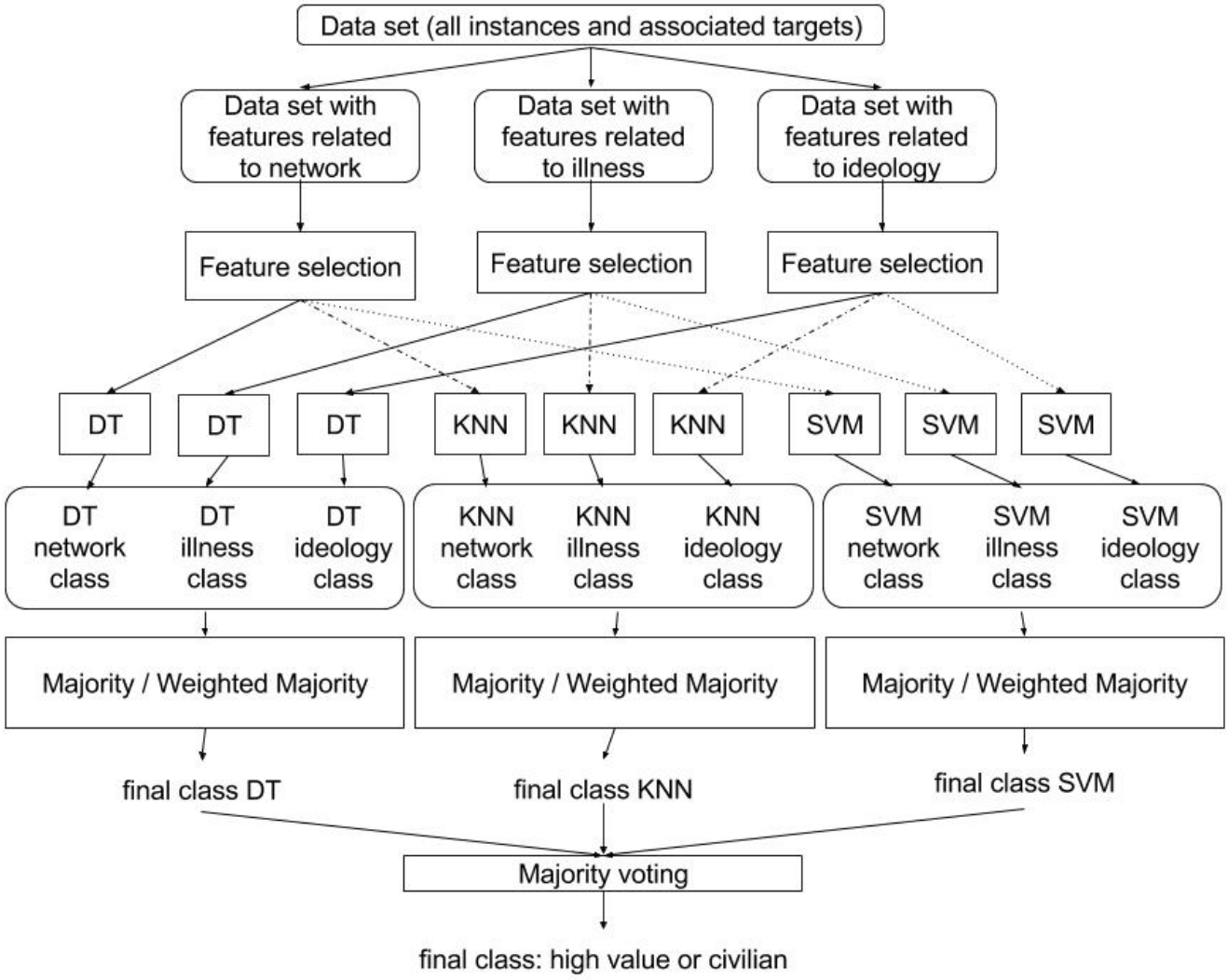


Figure 2: Classifier ensemble for data set with feature set split into themes.

2 Fusion labels

In the case of the ensemble classifier, the **majority voting** is applied to each of the constituent algorithms: Decision Tree, K Nearest Neighbours and Support Vector Machine respectively where each theme has an associated learning algorithm. The classifier outputs are assumed to be independent and there is no need to handle ties since there are three possible outputs and two labels. The number of classes, L, is odd (three in our case based on the themes). Thus, the class that appears twice will be assigned.

With **Weighted majority voting** we give the more performant classifier more influence in determining the final prediction. The Weighted Majority voting is applied to each of the constituent algorithms: Decision Tree, K Nearest Neighbours and Support Vector Machine respectively corresponding to each of the three themes. Initially, the weights are all 1 as there is no reason to prefer any classifier. We then compute the prediction, update the weights after each iteration and label a new instance using the updated weights.

3 Evaluation of terrorist target ensemble classification using cross validation

We evaluate the performance of several learning algorithms and perform a comparative analysis. We randomly select the training and testing sets, we repeat the 10 fold cross validation process 10 times and report the average of 10x10 cross validation results. This is done in order to provide a more stable estimation of the algorithm's performance and to decrease the variations of performance due to the choice of training and testing sets. We report the quality of predictions using various performance measures such as: accuracy, precision, recall and F1 score.

We test the classification performance using cross validation on the entire data set. We also report the classification obtained after selecting the most informative features as follows: we use the feature selection algorithm to select the best features and we then test the classification performance using cross validation. We compare the results in the case when all data is used in the ensemble classifier and in the case when the ensemble is trained for each of the three groups in which the set of attributes was split: ideology, network, mental illness. We also compare the results of the thematic models obtained after feature selection.

3.1 Ensemble classification on entire feature set

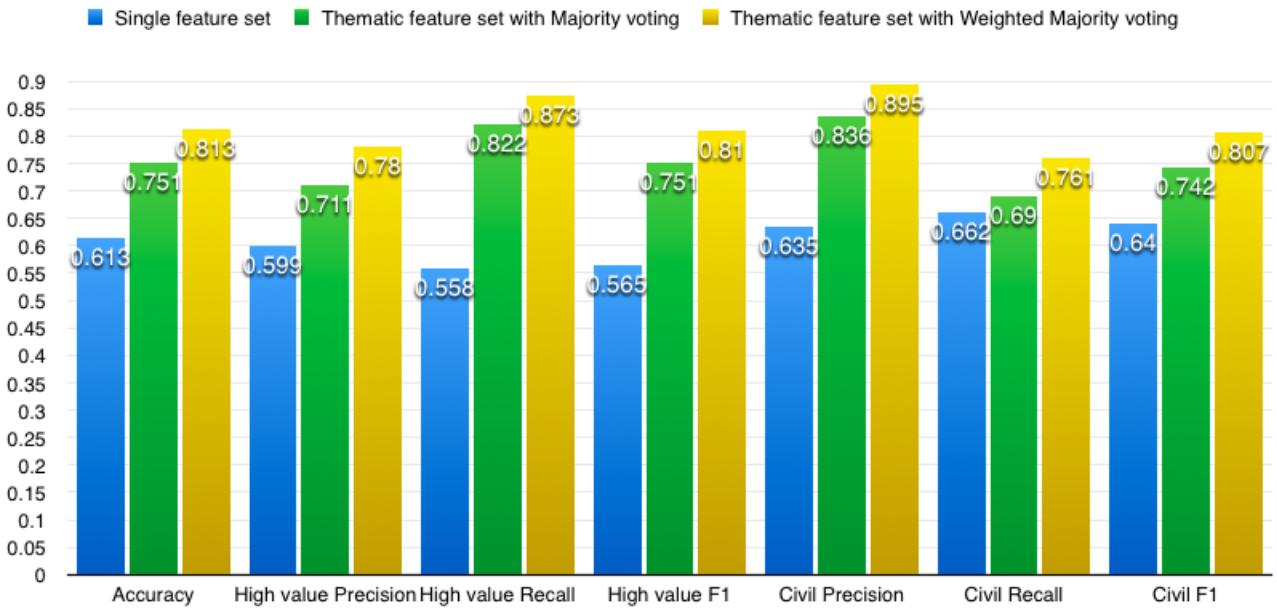


Figure 3: Comparison between single set of features and thematic set of features using all variables

The thematic split results show an improvement compared to the standard use of one group of features. The improvement varies between 14% and 20% depending on the fusion method. We not only reduce the training complexity, but we also achieve better F1 measures. In the case of the thematic split of features, the precision of the accuracy for each class is above 71% and the F1 measure for each class is above 74%.

The weighted majority algorithm gives the highest accuracy when compared to majority voting. Using this fusion method, the accuracy is improved by 6% compared with using the majority voting method. With weighted majority we also achieve a precision above 78% for each class, a recall measure greater than 76% for the two classes as well as a F1 measure above 80% for the two target choices, high value and civilian.

Comparing the ensemble method with Support Vector Machines, there is no significant difference when using the single set of features. The thematic split of variables gives better results when used with an ensemble classifier than with SVM. When feature selection is not applied, the ensemble on thematic features has an accuracy of 81.3%, 16.5% higher than using a Support Vector Machine. The individual results of the constituent algorithms of the ensemble can be found in Appendix.

3.2 Ensemble classification on subset of features

3.2.1 Single feature set

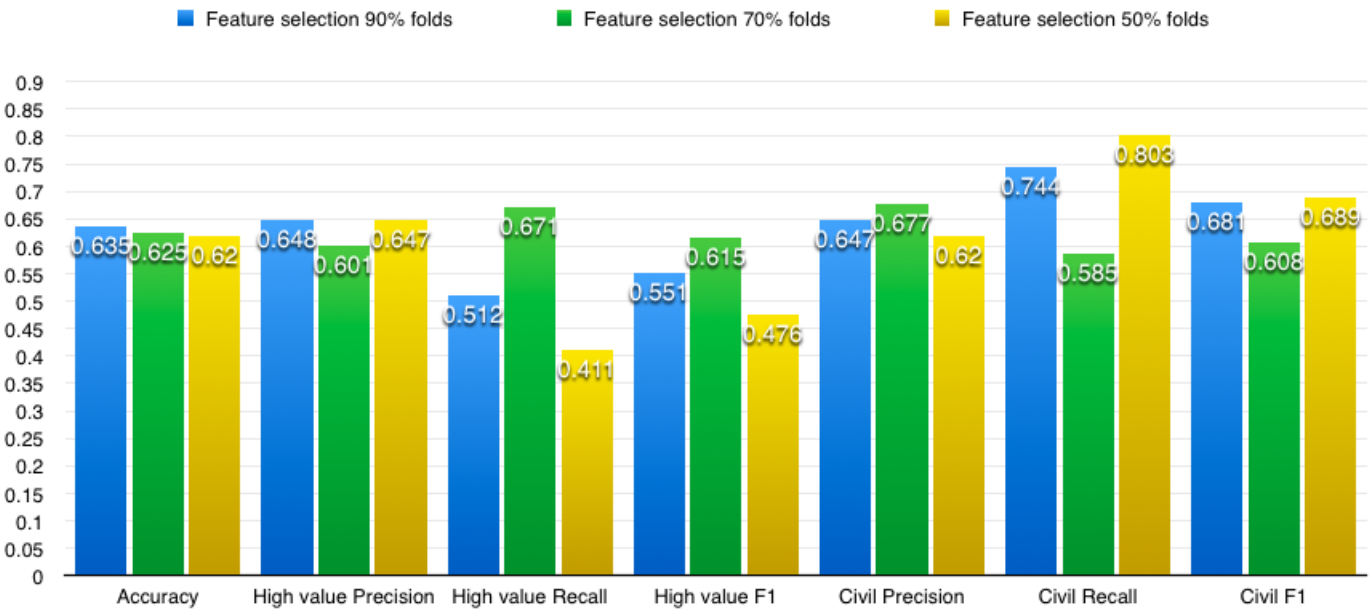


Figure 4: Single feature set using 90%, 70% and 50% features

Whilst the difference is not significant, the results show that feature selection improves the classification up to 2% in the case of a single set of features. The improvement in accuracy when applying feature selection varies between 1% in the case of feature selection in 70% and 50% of the folds, and 2% in the case of selecting the most informative features that appear in at least 90% of the folds. The best results when using a single set of features after selecting the most informative variables are achieved when selecting features that appear in at least 90% of the folds with an accuracy of 63.5% and a precision for each of the two classes greater than 64.7%.

There is no significant improvement in accuracy when using an ensemble of classifier instead of a Support Vector Machine. The individual results of the constituent algorithms of the ensemble can be found in Appendix.

3.2.2 Thematic sets of features

	Accuracy	High value Precision	High value Recall	High value F1	Civil Precision	Civil Recall	Civil F1
Majority	0.649	0.649	0.563	0.580	0.671	0.725	0.684
Weighted Majority	0.742	0.765	0.671	0.694	0.755	0.805	0.767

Table 1: Ensemble classification on subset of features with 90% thematic sets of features

	Accuracy	High value Precision	High value Recall	High value F1	Civil Precision	Civil Recall	Civil F1
Majority	0.652	0.678	0.493	0.553	0.650	0.791	0.707
Weighted Majority	0.742	0.785	0.637	0.687	0.734	0.835	0.774

Table 2: Ensemble classification on subset of features with 70% thematic sets of features

	Accuracy	High value Precision	High value Recall	High value F1	Civil Precision	Civil Recall	Civil F1
Majority	0.645	0.664	0.479	0.540	0.645	0.790	0.703
Weighted Majority	0.730	0.788	0.599	0.659	0.718	0.846	0.769

Table 3: Ensemble classification on subset of features with 50% thematic sets of features

When using the thematic feature split, the ensemble generally achieves the best accuracies irrespective of the number of features.

The Weighted Majority fusion is compared with the standard Majority Voting. The results show that using Weighted Majority voting as a fusion algorithm outperforms the standard Majority voting algorithm. With the Weighted Majority fusion, accuracies across classifiers are improved up to 10%. The best results were obtained when selecting features that appear in 90% of the folds with an accuracy of 74.2% and a precision greater than 75.5% for each of the two classes. Comparing the fusion algorithms when using a different number of features, the classifiers have similar performance. Whilst feature selection applied to thematic sets of features that appear in 90% and 70% of the folds respectively achieved the same accuracy of 74.2% and the differences in the F1 measures for each class are less than 1%, we prefer the 90% features as the ensemble classifier achieves a higher recall score for the high value case, that is the minority class as well as a higher F1 score.

The proposed approach, combining the results of classifiers applied to groups of similar features, outperforms the other tested classification methods. The results show the improvements of splitting the feature set into groups and combining the outputs of the individual classifiers assigned to each group. The fact that dividing the features into themes gives better accuracies than a standard single group features might be useful in reducing the fieldwork carried out by experts in order to understand relationships between behavioural attributes. The accuracy improvement when using categories of features compared to using a single set of features is of 20% when using all variables without applying feature selection and 11% when selecting the most informative variables.

Results show there is a small improvement in accuracy when using an ensemble of classifier rather than a Support Vector Machine in the case of selecting the most informative features from the thematic sets of variables. The individual results of the constituent algorithms of the ensemble can be found in Appendix.

4 Evaluation of terrorist target ensemble classification on new data set

An important aspect of a machine learning algorithm is estimating the performance of the system. Given the limited data set of 111 lone actor terrorists, we want to ensure the model generalizes well on unseen data.

We select the percentage of feature selection that gave the best results using cross validation as well as the fusion algorithm with highest scores and test them on a new data set collected by us. Results show that the fusion algorithm with the highest performance is Weighted Majority. Our feature selection algorithm gives the highest accuracy when selecting variables that appear in at least 90% of the folds. Missing values from the testing set are replaced by the mean as the data set does not contain variables with a high magnitude that could influence the results.

We report the results on the new data set when the label is the target choice of lone terrorists: high value and civilian. We use two approaches in selecting the variables: our pairwise feature selection method and the closest 70% variables from the MDS plot. We check how our classification method generalises by testing the same examples with two different labels: *Discriminate* and *Violence* instead of the *Target choice*.

The following results were obtained from a data set of 10 samples using a Weighted Majority voting as a fusion algorithm and selecting features that appear in at least 90% of the folds.

4.1 Target choice

	Accuracy	High value Precision	High value Recall	High value F1	Civil Precision	Civil Recall	Civil F1
Single feature set	0.800	0.500	0.500	0.500	0.875	0.875	0.875
Single feature set Feature selection	0.800	0.500	0.500	0.500	0.875	0.875	0.875
Thematic feature set	0.700	0.400	1.000	0.571	1.000	0.625	0.769
Thematic feature set Feature selection	0.700	0.000	0.000	0.000	0.777	0.875	0.823

Table 4: Results using *target* as label.

4.1.1 Features from Proxscal

Using the ensemble classifier, we report the precision, recall and F1 scores for each of the two targets, high value and civilian, using the features identified to be the most informative using Proxscal.

	High value Precision	High value Recall	High value F1	Civil Precision	Civil Recall	Civil F1
Single Feature set	0.646	0.444	0.504	0.615	0.794	0.680

Table 5: Results using the entire feature set as resulted from Multidimensional Scaling

Thematic Feature set	High value Precision	High value Recall	High value F1	Civil Precision	Civil Recall	Civil F1
Majority	0.660	0.515	0.558	0.614	0.748	0.667
Weighted Majority	0.771	0.625	0.669	0.700	0.797	0.732

Table 6: Results using the thematic feature set as resulted from Multidimensional Scaling

4.2 Discriminate

	Accuracy	High value Precision	High value Recall	High value F1	Civil Precision	Civil Recall	Civil F1
Single feature set	0.600	0.750	0.500	0.600	0.500	0.750	0.600
Single feature set Feature selection	0.700	0.800	0.666	0.727	0.600	0.750	0.666
Thematic feature set	0.600	0.666	0.666	0.666	0.500	0.500	0.500
Thematic feature set Feature selection	0.800	0.833	0.833	0.833	0.750	0.750	0.750

Table 7: Results using *discriminate* as label.

4.3 Violence

	Accuracy	High value Precision	High value Recall	High value F1	Civil Precision	Civil Recall	Civil F1
Single feature set	0.600	0.400	0.666	0.500	0.800	0.571	0.666
Single feature set Feature selection	0.400	0.285	0.666	0.400	0.666	0.285	0.400
Thematic feature set	0.900	1.000	0.666	0.800	0.875	1.000	0.933
Thematic feature set Feature selection	0.500	0.375	1.000	0.545	1.000	0.285	0.444

Table 8: Results using *violence* as label.

The results of the experiments we conducted on the new data set using the ensemble classifier underline how the proposed technique of splitting the feature set into groups outperforms or achieves similar results in terms of accuracy compared to the standard approach of using a classifier for a single group of features. This is irrespective of the label chosen for the examples and the results are independent of whether we perform feature selection before classification.

Using variables from the MDS plots, we obtain better precision, recall and F1 measures in the case of the thematic split of features compared to using a single set of variables.

5 Summary of terrorist target ensemble classification

Integrating multiple classifiers in order to make use of the strengths of one method to complement the weaknesses of another learning algorithm can have a better performance than using a single classifier. We examined whether an ensemble of classifiers that includes the Support Vector Machine previously modelled can increase the classification accuracy. A supervised classification of a binary target, high value and civilians, is implemented using an ensemble classifier that consists of Decision Trees, K Nearest Neighbours and Support Vector Machines. We used two strategies in performing terrorist target classification: firstly, using the data set with a single set of features and secondly, using the data set with features divided into categories and combined the outputs using two approaches: Majority voting and Weighted Majority voting. In both cases, we compare the results of the ensemble obtained when using all the available features with results obtained after performing feature selection. The ensemble outperforms the Support Vector Machine or achieves similar performance and the results suggest that it is better to split the features in categories, rather than using the entire set of available features.

Using cross validation, the best results are achieved when splitting the feature set into themes with 81.3% accuracy in the case of using all features and 74.2% accuracy when selecting the most informative features that appear in at least 90% of the folds and combining the results using Weighted Majority voting.

8

Modelling terrorist behaviour

Since the behaviour in terrorism is uncertain, there is a need for a model that can handle the probabilistic aspect of behaviour analysis. We use the most informative features as identified in the first part where we modelled classification in order to generate the networks. Thus, we are not interested in learning a classification function using a Bayesian network, but a general model of how these features interact.

We use an objective approach to construct three network structures for each of the three themes, network membership, ideology and illness from the data set. Splitting the feature set into themes results in sets that can be analyzed separately. The interaction between variables belonging to different groups can be obtained by combining the smaller networks. We use the networks for probabilistic inference in order to identify the most likely value of each variable based on evidence, that is observations of at least one variable.

One way of improving the network is to allow an expert to modify the network, that is introducing expert knowledge into the network. We developed a web-based analysis graphical interface which allows experts to change a network and thus incorporating domain knowledge through an application that provides a simple and efficient visualisation of variables related to lone terrorist behaviour as well as how these variables interact and influence each other.

1 Bayesian network

A Bayesian network is a graphical model that represents probability relationships among variables. It is a directed acyclic graph in which nodes represent features and links represent direct influences between the features. Learning the structure of a Bayesian network consists of two tasks: learning the directed acyclic structure of the network and learning the parameters of the network. The structure can also be found by learning the conditional in-

dependence relationships between variables and then use these independencies to construct the Bayesian network [68]. The network learnt can then be used in inference, using evidence propagation to predict the probable values of each feature.

Traditional approaches require experts to determine the relationships between attributes and the conditional probability tables [69]. However, this approach is highly subjective, it can be difficult to express knowledge in terms of probability which may be biased and can lead to ignoring relationships between variables that have not been previously identified by experts.

The main advantages of the Bayesian network are that it can handle probability and uncertainty, it can distinguish between unconditional independence and conditional independence which may be difficult to be observed by criminologists. The Bayesian Network is useful in analysing the influence a variable has on the other attributes. One advantage of a Bayesian network is that it allows for incorporating the domain knowledge by defining prior information about the model. This is in contrast with other machine learning algorithms such as neural networks or decision trees [68].

The main limitation is related to the complexity of the problem, the limited data set and that it is not feasible for data sets with many features [68]. However, in our case, most of the attributes are binary thus conditional probability tables have fewer values and the computation is faster than in the case of discrete variables with more than two states.

1.1 Approach

A Bayesian network representing the lone terrorism behaviour is modelled by linking variables related to the offender's profile with the offender's actions pre and during the attack, one for each of the three categories: connections to a network, mental illness and ideology. To identify the relationships between these variables, we apply structural and parameter learning on our data set. Whilst there is an ongoing debate as to whether the objective or subjective approach is more appropriate, we apply an objective approach and learn the structure of the Bayesian network from the available data set. The main issue is that there are many possible dependencies between variables and estimating the probabilities of these relations is known to be NP-complete. Learning a network using all the features available is not possible as it will result in a huge model and a long computation time.

We construct three network structures for each of the three themes, network membership, ideology and mental illness. Splitting the feature set into themes results in sets that can be analyzed separately. This follows the type of reasoning used by criminologists, considering groups of variables. The interaction between variables belonging to different groups can be obtained by combining the smaller networks. Each network that was constructed displays variables that are disconnected from the graph. This is because no relationships were identified in the data set during the network structure learning. Another reason for the lack

of links of certain variables in the graph is that a relationship does simply not exist. We use the networks for probabilistic inference in order to identify the most likely value of each variable from evidence.

1.2 Koller-Friedman network learning algorithm

In [61] Koller and Friedman propose an algorithm for constructing the structure of a network based on the variable's independence properties, method known as a constraint-based approach. The Bayesian network is defined as a pair B=(G,P) where the distribution P factorizes over graph G and P is defined as:

$$P(X_1, .., X_n) = \prod_{i=1}^{n} P(X_i | Pa_{X_i}^G) \tag{1}$$

Even in the case of the variables being all binary, 2^n examples are required to compute the joint distribution and to estimate the network parameters robustly, which is impossible to achieve. Koller and Friedman propose an algorithm that at the core uses an independency map for P, consisting of the independence assertions of the form $(X \perp Y|Z)$ given G, perfect maps for a set of independencies which represent graphs that capture all the independencies in a distribution P and the concept of d-separation which states that two variables, X and Y, are d-separated if X can influence the value of Y in the presence of evidence of variable Z. If X and Y are not d-separated given Z in G, then X and Y are dependent given Z in a distribution P that factorizes over G [61]. Formally, this is expressed as:

$$If\ X \rightarrow Y\ in\ G\ then\ P \not\models (X \perp Y|U)\ for\ any\ U\ that\ does\ not\ include\ X\ and\ Y. \tag{2}$$

The undirected graph is constructed by considering pairs of variables and testing for independence given another variable. This is based on the idea that if X and Y are adjacent in G, then they cannot be separated with any variable. Using the undirected graph, the algorithm then identifies the edge direction.

A joint distribution P defines the probability $P(Y = y|E = e)$ of an event y given observations e. This is calculated by conditioning the joint distribution on the observation E=e, eliminating entries in the joint distribution not consistent with e and then normalizing the resulting entries to sum to 1.

1.2.1 Modifying the Koller-Friedman algorithm

The method for constructing the Bayesian network has two stages. Firstly the directed acyclic graph is constructed from the data. The second part of the algorithm estimates the parameters of the Bayesian network given the directed acyclic graph in order to maximise the probability of the data.

The algorithm runs independence tests for each pair of variables calculating the independencies and conditional independencies between features. In our case, each pair of variables, X and Y, is conditioned on a single feature, U. If known, this feature U leaves the variables X and Y independent. If there is a feature that separates the two variables under test then they will not be connected, otherwise they will appear as adjacent in the graph. The independence tests of variable pairs are run against a single other variable, otherwise a large number of examples is needed.

We modified the initial algorithm as found in the libpgm library to use **Kullback Leibler divergence**, a non-symmetric measure of the difference between two probability distributions instead of chi-square test. The Kullback-Leibler divergence of Q from P is stated mathematically as:

$$D_{KL}(P||Q) = \sum_i P(i) ln \frac{P(i)}{Q(i)} \tag{3}$$

We changed the method from using the chi-squared test because in cases where there is a small number of observations a G-test is more appropriate [70]. The G-test value is proportional to the Kullback-Leibler divergence measure (also known as information gain) of the difference between two probability distributions [71]. For each pair of variables, we generate the Kullback Leibler divergence of the two arrays: the expected distribution if X and Y are independent given U and the actual distribution of X, Y and U. If the value is less than a predefined parameter, the pair of variables are dependent given the feature under test.

Once all dependencies are found, the algorithm constructs the directed acyclic graph by considering triplets of nodes $X_i - X_j - X_k$ to identify the edge directions.

1.2.2 Structure learning complexity

Given n variables, the algorithm checks n^2 pairs and for each pair there are $(n-2)$ independence tests. The complexity of the algorithm is $O(n^3)$. The worst case is when variables are connected in the graph because no feature is found to separate the variables and all independence tests have to be run.

1.3 Gibbs sampling

We estimate the posterior probabilities of the variables given some evidence using Gibbs sampling by drawing n samples and calculate the probabilities based on frequencies. Initially the Gibbs method draws a sample from the distribution not conditioned on the evidence. The unknown nodes are assigned a random state. Iterating over the unobserved variables, a node is selected and its state is calculated using its Markov blanket. The node is instantiated based on the calculated distribution and a new sample is recorded using the calculated state of the node and the current sample for all other variables.

1.4 Creating a single network

When splitting the feature set into themes we included the features that cannot be categorized such as level of education, employment etc in each of the variable set for the three themes. Having constructed three network structures for each of the three themes, network membership, ideology and mental illness from the data set, we extend the networks and thus identify the interactions between variables belonging to different groups by combining the smaller networks. We combine the networks rather than learning the structure of the network with all features because there are many possible dependencies between variables and estimating the probabilities from our data set will not result in an accurate model given the number of nodes for the combined network and the size of our data set. However, we learn the structure of the thematic networks using fewer variables and hence the thematic graph will be more accurate.

In order to construct the single network, we use the set of nodes resulted from the three networks combined. We use the edges from each network to construct the graph of the combined network as follows: we iterate over the edges of each network and add the edge (X_i, X_j) to our final set of edges only if there is no path from node X_j to node X_i in the current graph. This is necessary in order to overcome a case such as the one depicted in Figure 1 where the message passing will not terminate.

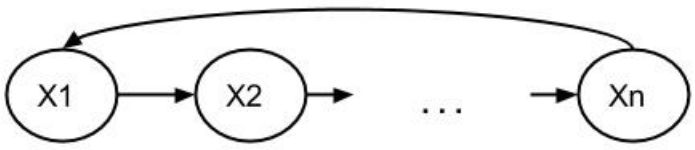


Figure 1: Edge $X_n - X_1$ is checked against the graph to determine whether there exists a path from X_1 to X_n.

Using the directed acyclic graph, the parameters of the Bayesian network are computed in order to maximise the probability of the data.

Listing 8.1: Combining networks

```
edges = set_of_edges(create_graph(edges, nodes))

skel = GraphSkeleton()
skel.load_skel(nodes, edges)
skel.toporder()

learner = PGMLearner()
bn, passed = learner.discrete_mle_estimateparams(skel, max_learning_data)
```

1.5 Incorporating expert knowledge

One way of improving the network is to allow an expert to modify the network, that is introducing expert knowledge into the network. An expert in the field can modify the model and thus find an improved model by changing the current one, adding or removing links between features and deleting nodes from the network. An important aspect in Bayesian networks is incorporating the expert's knowledge from the domain. In the domain of terrorism, it is difficult to have sufficient training data to learn both the structure and the parameters of a Bayesian network. Whilst a purely objective approach does not seem feasible to provide an accurate representation given the size of the available data and a purely subjective approach can be biased given that it relies solely on the the expert's knowledge, we feel that the best approach is to learn the structure and parameters of the network from the data set and then allow an expert to modify the network to incorporate the knowledge from the specific domain such as relations suggested by the existing literature. Thus, models can be enhanced using the criminology experts.

1.6 Cycles in network

The structure of each network is represented by a directed acyclic graph. An issue that arises when allowing a user to modify the network is related to introducing cycles in the network. To overcome this problem we used Ebay BBN, a python library that supports conversion to join trees and exact inference on cyclic graphs. Hence we combined the structure learning capabilities of libpgm with inference on multiply connected networks supported by BBN.

A multiply connected network can be replaced by a singly connected network using a join tree in which nodes are not single variables, but clusters of variables. A Bayesian network is transformed into a join tree by:

- moralising the graph by joining unjoined parents
- triangulating the graph to obtain a graph in which the maximum cycle length is three

- identifying the cliques
- ordering the cliques based on the running intersection property which states that if a node appears in two cliques, then it appears in all the nodes on the path that connects the cliques [72]

The message passing occurs between clusters and once it converges, the marginal distribution of each variable can be obtained and thus identify the likelihood of the states of each variable in the clique.

2 Graphical interface for representing networks

We developed a web-based analysis graphical interface which allows experts to change a network and thus incorporating domain knowledge. A user can see the most probable states of the variables in the network by selecting the evidence, thus allowing probability propagation in the network to compute the probabilities of the other attributes. Our model consists of three networks, each related to one of the three themes identified in the previous section: relations with members from a network, mental illness and ideology. The application provides a simple and efficient visualisation of variables related to lone terrorist behaviour as well as how these variables interact and influence each other.

We use Bayesian Networks to model the behaviour of the lone wolves in order to provide a better representation of the relations between characteristics and to infer unobserved variables given any evidence. The Bayesian Network is learnt from the available data that contains the attributes obtained after feature selection. The application provides a way of visualising the networks, updating the structure by adding or removing edges and deleting nodes, performing inference based on evidence and displaying the most probable state of each variable as well as displaying the likelihood of the states of each variable through pie charts.

2.1 Implementation details

When learning the networks associated to each of the three themes: individual's relations to members of a wider network, mental illness and ideology, the network structures are saved as JSON[1] files. To generate the original networks, the data is initially read from these files.

We used Flask to create a RESTful [2] API that uses GET and POST requests for performing inference and reading and updating the network structures respectively. The network

[1] http://json.org/
[2] http://en.wikipedia.org/wiki/Representational˙state˙transfer

visualisations were implemented using the d3 library and the pie charts representing the inference results were implemented using the highcharts library.

2.2 Illustrating the Bayesian networks

The graphical interface presents a Bayesian network for each of the three themes as well as the network combined from these:

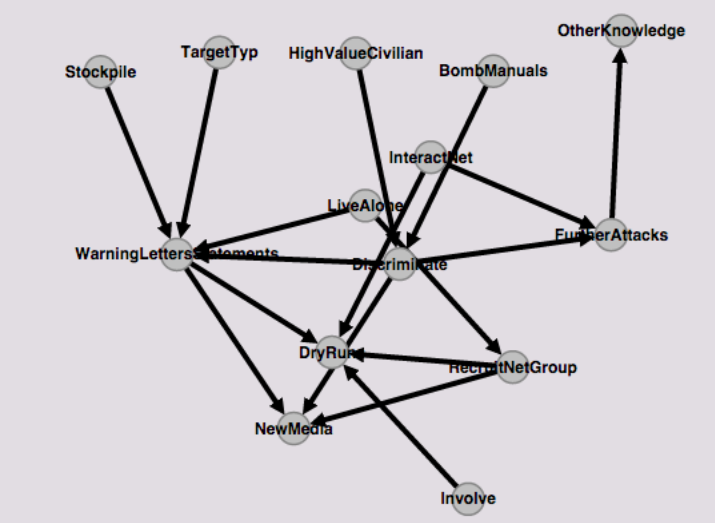


Figure 2: Bayesian network with features related to network membership.

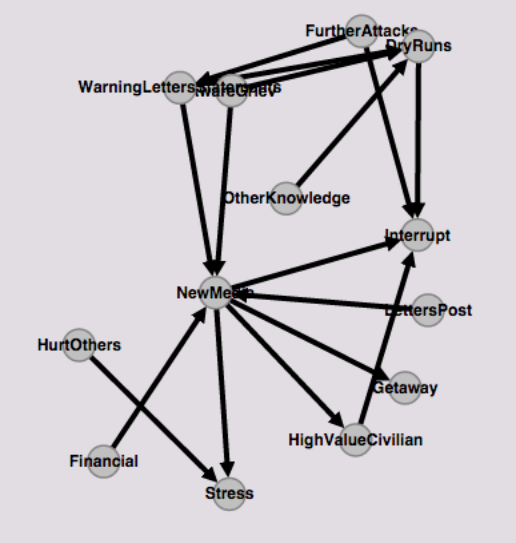


Figure 3: Bayesian network with features related to mental illness.

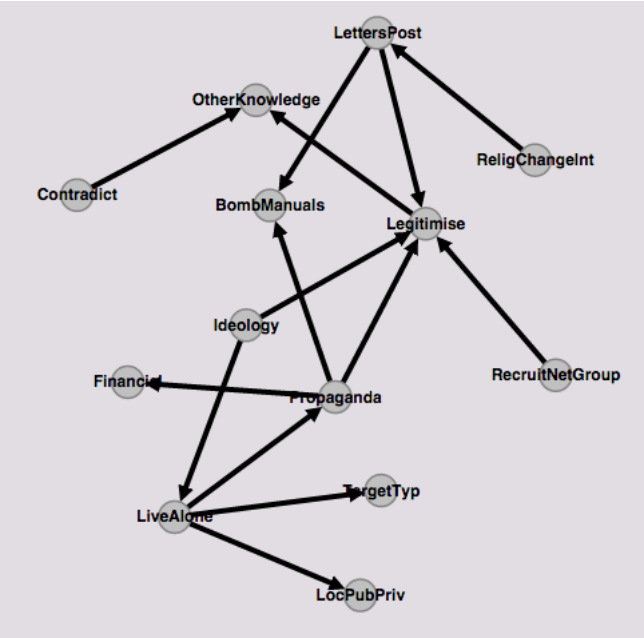


Figure 4: Bayesian network with features related to ideology.

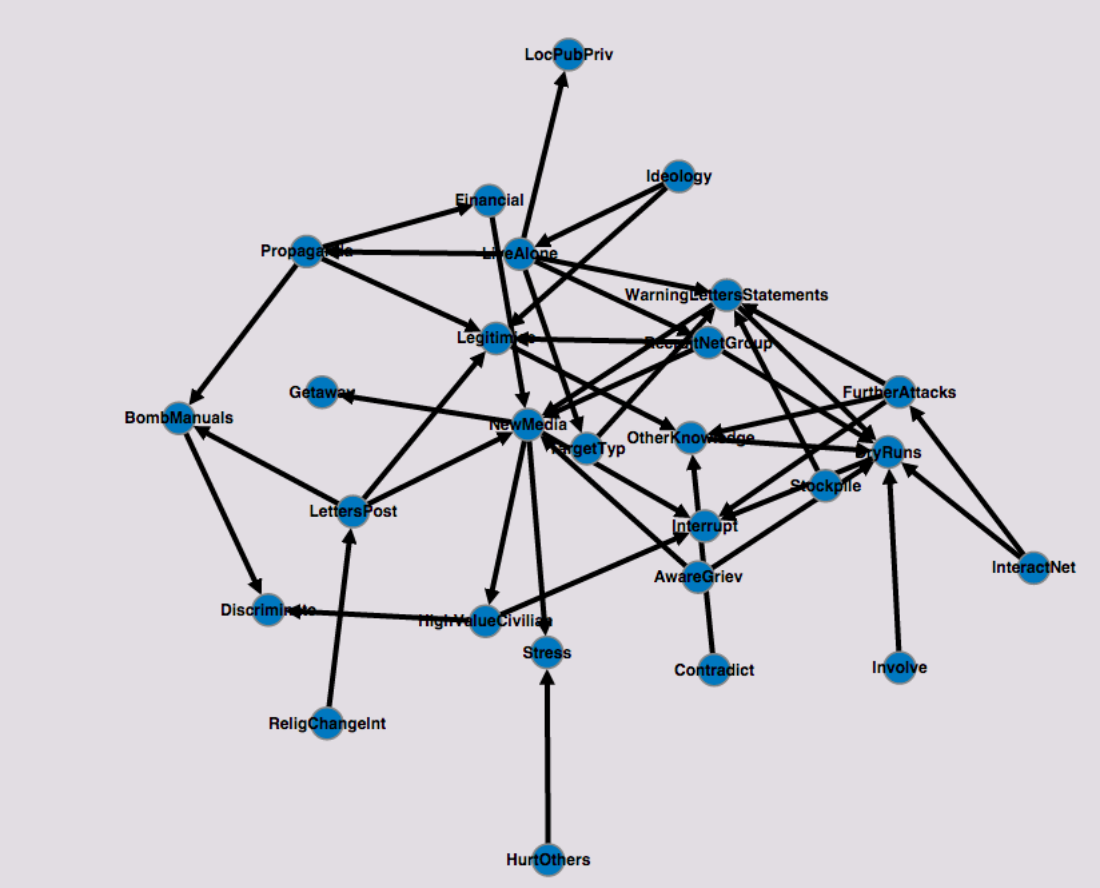


Figure 5: The Bayesian network resulted from the combination of the smaller networks.

2.3 Updating the Bayesian network

An expert in the field can modify the network and thus find an improved model by changing the current one, adding or removing links between features and deleting nodes from the network.

To delete a node, a user must first select the node by clicking on it and then press *delete*. To add an edge between two nodes, a user must drag from the source node to the target node. To remove a node, a user must first select the edge by clicking on it and then press *delete*. Given that the graph underlying the network is not necessary a tree structure and hence edges may overlap, a user can change the graph layout by clicking *ctrl* and then drag nodes. The user can always select to return to the original network that was learnt from the data.

2.4 Information from inference

Using the evidence chosen by the user, inference is run and the distribution of each variable's states is presented back to the user. When inference is completed, the network highlights the nodes of the networks using different colours to show the nodes that represent the evidence, the nodes with the most likely state above a certain threshold and the rest of the nodes with the most likely state that are not above the threshold. The user can change the threshold and the network is updated outlining the nodes that have the predicted state above the new threshold. We display the likelihood distribution of each feature through a pie chart.

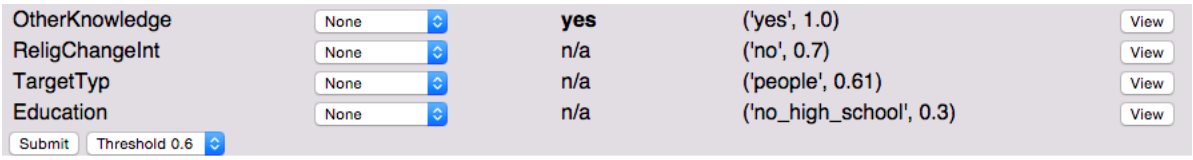
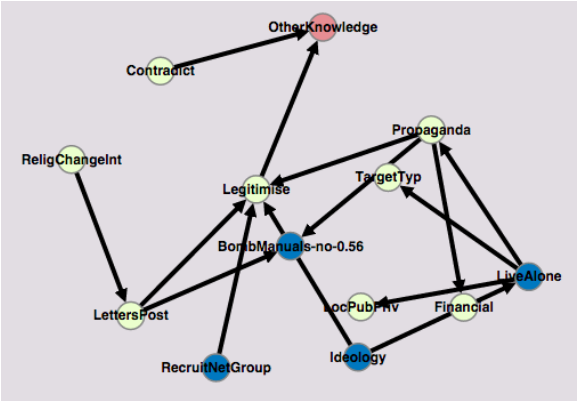
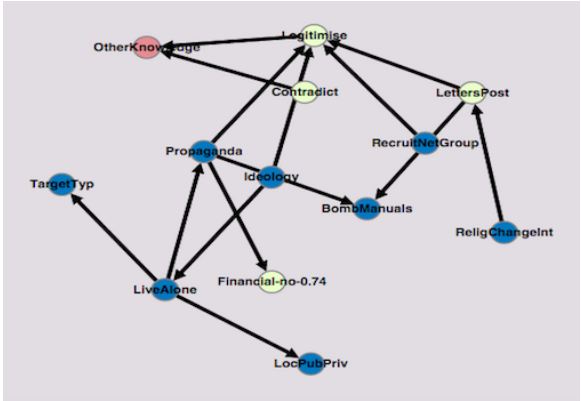


Figure 6: Selecting evidence.

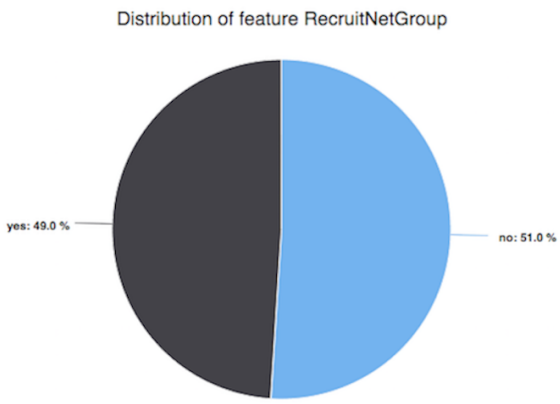


(a) Feature with maximum likelihood above default threshold 0.6

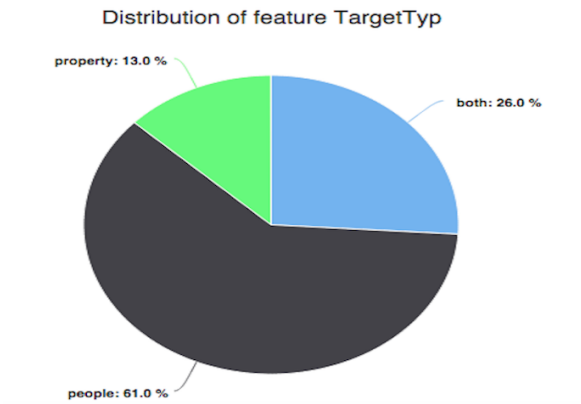


(b) Feature with maximum likelihood above threshold 0.7

Figure 7: Displaying features based on likelihood threshold.



(a) Distribution of feature *RecruitNetGroup*



(b) Distribution of feature *TargetTyp*

Figure 8: Feature distribution.

3 Evaluation of modelling terrorist behaviour

We use the network learnt in inference, that is we propagate the evidence to predict the probable values of each feature and then compare the predictions with the true values of our instances on the new data set. For example, we can predict the values of features such as Mental Illness and Network influences, which are not known unless further investigation is conducted. Correct identification of the true values of the features can extend the knowledge about lone terrorists as well as improving the understanding of how these features influence each other.

In order to test the Bayesian networks we select the nodes that represent the evidence and predict the values of the unobserved nodes. The predicted value is represented by the state of the variable with the highest likelihood. For each example, we compare the predicted

value with the true state of the variable.

We show the performance of the three different networks, relations with members from a wider network, mental illness and ideology, as well as the performance obtained when combining these networks into a single network.

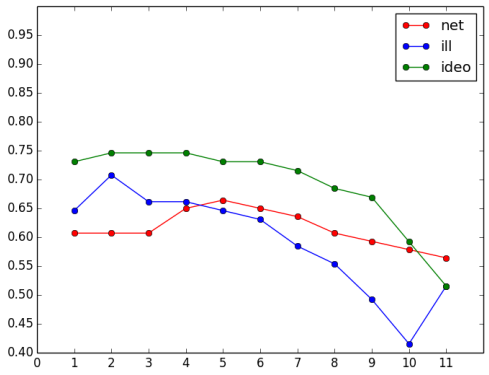
The Bayesian network can be used to test and identify new hypothesis of terrorism psychology. It can also be used to infer characteristics about the offender which are unknown.

3.1 Bayesian networks performance

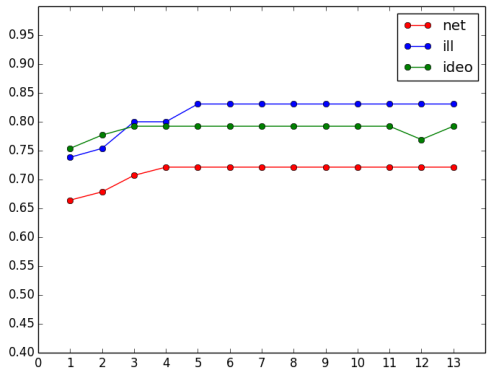
An important aspect of a machine learning algorithm is estimating the performance of the system. Given the limited data set of 111 lone actor terrorists, we want to ensure the model generalizes well on unseen data. The learning accuracy can be unreliable on a limited data set. To overcome this problem, only the attributes obtained after the elimination of low quality features are used in the Bayesian Network.

We select a subset of variables from the nodes that are part of the networks and perform inference using combinations of subsets of these features. We report the results when all binary variables are *yes*, thus when an individual presents all behavioural characteristics, and when all binary variables are *no* respectively. We then report the resuts of these predictions considering the likelihoods that are above 60%.

The following plots show the results of inference on the new data set:

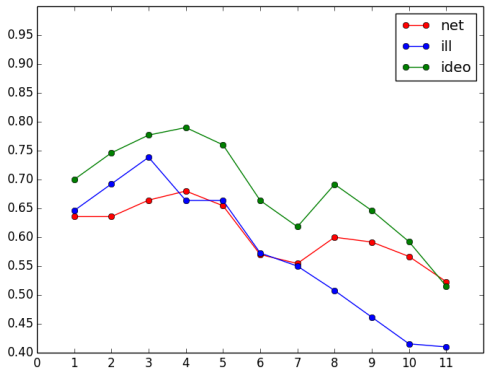


(a) Features from the Bayesian network with evidence of binary features encoded as *yes*

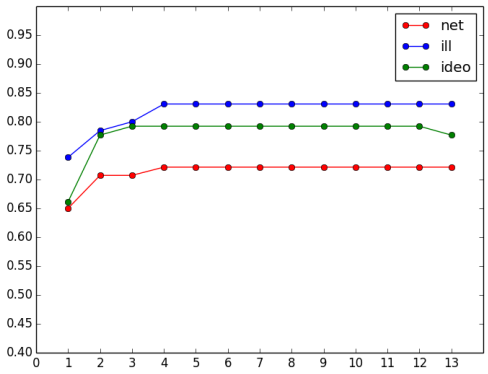


(b) Features from the Bayesian network with evidence of binary features encoded as *no*

Figure 9: Features with most probable state.



(a) Features from the Bayesian network with evidence of binary features encoded as *yes* above threshold 0.6

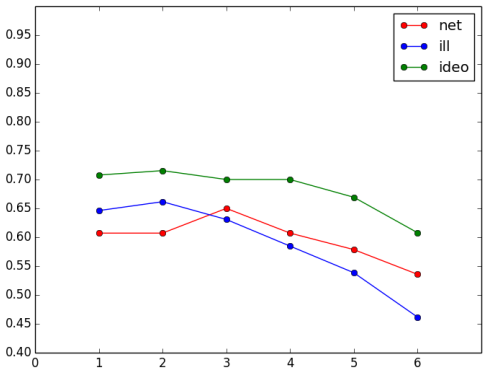


(b) Features from the Bayesian network with evidence of binary features encoded as *no* above threshold 0.6

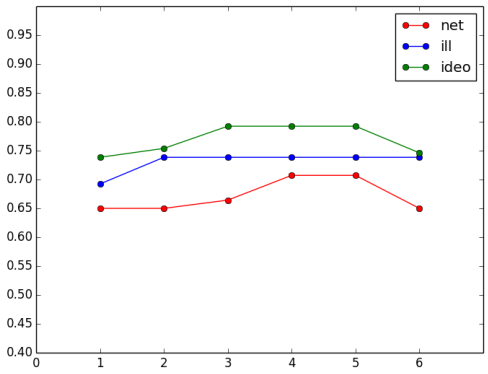
Figure 10: Features with most probable state above threshold 0.6.

The graphs show that the Bayesian networks achieve a better accuracy when features are *no* compared to the case when the variables are selected as *yes*. This may be attributed to weak relationships between the selected variables and the other features that form the network. The ranking is similar in all cases with the network related to ideology performing the best, followed by the network related to mental illness and the one related to network membership. There is a decrease in the performance of all networks, irrespective of the probability threshold, when more than 5 attributes are used as evidence in the case of binary variables encoded as *yes*. In the case of binary variables having value *no*, there is a constant performance for each network starting from 4 nodes used as evidence, independent of whether the likelihoods are considered above the 0.6 threshold. In this case, the performance is greater than 70% for each Bayesian network, meaning that at least 70% features are correctly identifed.

We select the features identified in the key findings from [8] in which the same data set was used and perform inference using combinations of subsets of these features. We analyse the case when all binary variables are *yes* and when all binary variables are *no* respectively.

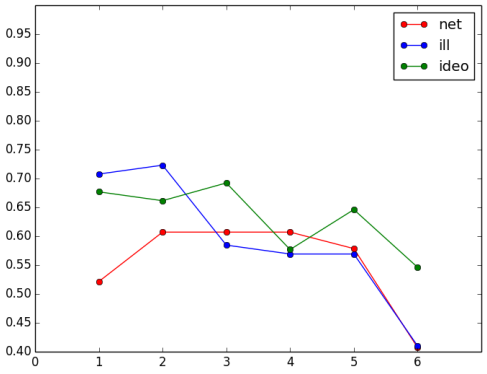


(a) Selected features from the Bayesian network with evidence of binary features encoded as *yes*

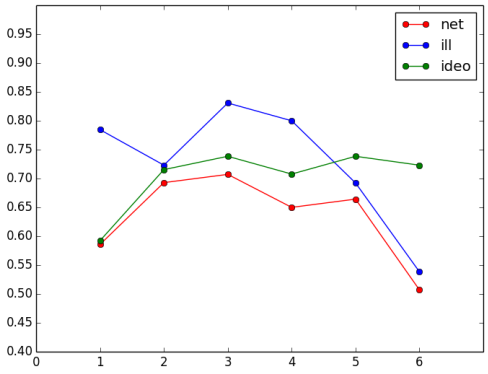


(b) Selected features from the Bayesian network with evidence of binary features encoded as *no*

Figure 11: Selected features with most probable state.



(a) Selected features from the Bayesian network with evidence of binary features encoded as *yes* above threshold 0.6



(b) Selected features from the Bayesian network with evidence of binary features encoded as *no* above threshold 0.6

Figure 12: Selected features with most probable state above threshold 0.6.

The graphs generated using key attributes as identified in the literature show that the Bayesian networks achieve a better accuracy when features are *no* compared to the case when the variables are selected as *yes*. This could be due to weak relationships between the selected variables and the other features that form the network. In the case of measuring the performance using the likelihoods irrespective of their values (i.e. threshold), the network related to ideology achieved more correct predictions than the network related to mental illness and the one related to network membership. In the case of considering likelihoods irrespective of their value, the Bayesian networks with evidence variables encoded as *no* predict the true state of 65%-80% of variables.

We analyse the network represented by the combinations of the three smaller graphs by performing inference using combinations of the previously identified key features. We report the results for the case when all binary variables are *yes* and for the case when all binary variables are *no*.

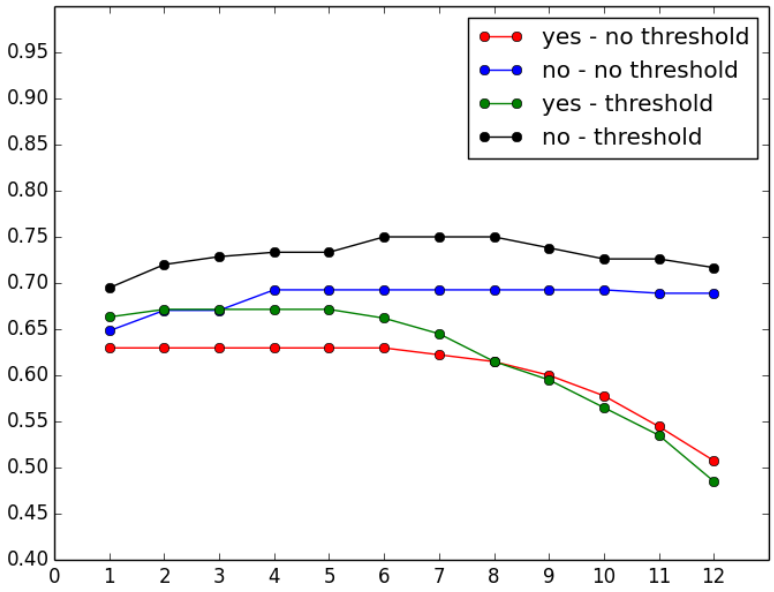


Figure 13: Performance of the combined networks considering all nodes and nodes with likelihood above threshold 0.6 respectively.

Figure 13 shows results consistent with the previous findings from using the Bayesian networks for inference. The Bayesian network obtained as a combination of the three smaller networks related to ideology, mental illness and network connections achieves a better accuracy when features are *no* compared to the case when the variables are selected as *yes*. The combined network achieves a more consistent predictive performance across variables compared to the individual networks. When the variables from the evidence are encoded as *no*, the Bayesian network predicts the true state of 65%-75% of the variables.

3.2 Expert evaluation of Bayesian network graphical interface

We asked Dr. Paul Gill who provided the data set for the project to give feedback on the graphical interface. The feedback was positive and thus represented a confirmation for the need of a tool that models behaviour. The interface was described as *practical* and the results from inference of evidence of variables can outline what the likelihood of the most probable state is. It was stated that being able to see the probability of the states of each feature can allow experts in the field to order the variables based on their likelihood and

thus identify which characteristic or antecedent behavior they should focus on and which question to ask next related to variables.

An improvement would be to incorporate an explanation of the overall framework to provide the users the background of how the relations between variables were identified from the data.

4 Summary of modelling terrorist behaviour

We used the most informative features as identified in the first part where we performed classification in order to generate the Bayesian networks that model how these features interact. We constructed three network structures for each of the three themes, network, ideology and mental illness from the data set. The interaction between variables belonging to different groups can be obtained by combining the smaller networks. We use the networks for probabilistic inference in order to identify the most likely value of each variable based on evidence. The network can be improved by allowing an expert to modify the network. We developed a web-based analysis graphical interface which allows experts to change a network and thus incorporate domain knowledge through an application that provides a simple and efficient visualisation of variables related to lone terrorist behaviour as well as how these variables interact and influence each other.

9

Future work

There is more work to be done in the field of criminology and the models developed in this project can be further developed. The behaviour analysis framework can be applied across a wide range of application domains. Future research can be conducted in the following areas:

- feature selection
- thematic split of features
- lone terrorism vs group terrorism
- Bayesian network

Feature selection

The method presented analyses all pairwise combinations of variables for determining the most informative features. Further experiments are required to determine whether the approach is feasible for large data sets with hundreds of features. Our approach of selecting features is independent of the prediction algorithm as we select the most important features before classification. Another strategy in selecting variables would be to run the feature selection algorithm with classification and select the feature set with the highest accuracy.

Thematic split of feature set

The issue of splitting the feature set into groups is a topic for future work to test in other domains. This method can be applied to applications based on user profiling and behaviour analysis to test whether thematic split of feature results in a better model compared to the case of using a single set of variables. One difficulty would be identifying the relevant categories for the split.

Lone terrorism vs group terrorism

In this project we focused entirely on data related to lone terrorism. The proposed model can be extended to classifications of group terrorism as well as other types of crimes. We can evaluate our feature selection algorithm combined with the ensemble classifier and test the performance of this approach on a different data set.

Bayesian network and graphical interface

Currently, an user can introduce expert knowledge into networks by deleting nodes and adding or removing edges from the graph, that is modyfing relations between variables. An extension would be to allow the expert to update the Bayesian network by changing the conditional probability tables of the network edges. An expert could also update the Bayesian network by adding nodes to the network. In the current model the prior probabilities, that is the a priori knowledge about the model, are generated from the data. One improvement would be to allow the experts to select or change the prior probabilities of the variables. This way the network is compared with previously published results and the relationships among variables in the domain are altered in the case of discrepancies between the graphical model and previous findings.

The current interface requires the networks to be specified explicitly by means of JSON files. The application can be extended to allow the user to upload a data file from which the networks can be learnt and then displayed to the user.

The model developed has two main components: classification and Bayesian networks. Whilst the Bayesian networks were constructed using the features identified to have the best accuracy in classification, the graphical interface can be considered largely independent and can be adapted to be used in any user profiling or behaviour analysis domain. Future work includes incorporating our method into applications based on user profiling and behaviour analysis. Given that a Bayesian network identifies relations between variables, this type of model can be used in understanding behaviour.

10

Conclusion

The problem of identifying the most relevant characteristics and antecedent behaviors of lone actor terrorists and the relationships between these variables is subtle, unpredictable and difficult. We experimented with various combinations of machine learning algorithms and feature sets and provided a way of visualising and updating Bayesian networks.

We presented a feature selection method based on information gain and a model for lone terrorism classification based on three themes: relations with members from a wider network, mental illness and ideology as identified in the literature. Our feature selection algorithm not only refines and limits the factors that do not contain enough information to perform classification but also provides a method for guiding terrorism researchers on what aspects they should focus on. Using cross validation, we achieved a classification accuracy combining the thematic sets of features of 81.3% when trained using the available features and of 74.2% when using feature selection compared to 61.3% and 63.5% respectively on the single set of features.

We constructed three network structures for each of the three themes, network membership, ideology and mental illness from the data set and combined these networks to create a single graph. We used the networks for probabilistic inference in order to identify the most likely value of each variable based on evidence. We developed a web-based analysis graphical interface which allows experts to change a network and thus incorporate domain knowledge through an application that provides a simple and efficient visualisation of variables related to lone terrorist behaviour as well as how these variables interact and influence each other. The Bayesian network framework can help researchers to test theories as well as to refine knowledge.

Bibliography

[1] Clark McCauley, Sophia Moskalenko, and Benjamin Van Son. "Characteristics of Lone-Wolf Violent Offenders: a Comparison of Assassins and School Attackers". In: *Perspectives on Terrorism* 7.1 (2013). ISSN: 2334-3745.

[2] Clark McCauley and Sophia Moskalenko. "Toward a Profile of Lone Wolf Terrorists: What Moves an Individual from Radical Opinion to Radical Action". In: *Terrorism and Political Violence* 26 (2014), pp. 69–85.

[3] P. Gill and E. Corner. "Disaggregating Terrorist Offenders: Implications for Research and Practice". In: *Criminology & Public Policy* 12.1 (July 2013), pp. 93–101. DOI: 10.1111/1745-9133.12015.

[4] *Looking Back, Looking Forward: Perspectives on Terrorism and Responses to It Strategic Multi-layer Assessment Occasional White Paper*. Sept. 2013.

[5] Jason Spitaletta. *Psychological Risk Factors of Terrorism*.

[6] Clark McCauley and Sophia Moskalenko. *Two Possible Profiles of Lone-actor Terrorists*.

[7] Jason Spitaletta. *Countering Terrorists: Psychological Risk Factors of Radicalization*.

[8] P. Gill, J. Horgan, and P. Deckert. "Bombing Alone: Tracing the Motivations and Antecedent Behaviors of Lone-Actor Terrorists". In: *Journal of Forensic Sciences* 59.2 (2014), 425435. DOI: 10.1111/1556-4029.12312.

[9] Paul Gill. "Terrorist violence and the contextual, facilitative and causal qualities of group-based behaviors". In: *Aggression and Violent Behavior* 17.6 (2012), pp. 565 –574. ISSN: 1359-1789. DOI: 10.1016/j.avb.2012.08.002. URL: http://www.sciencedirect.com/science/article/pii/S1359178912000894.

[10] Anna L. Buczak and Christopher M. Gifford. "Fuzzy Association Rule Mining for Community Crime Pattern Discovery". In: *ACM SIGKDD Workshop on Intelligence and Security Informatics*. ISI-KDD '10. Washington, D.C.: ACM, 2010, 2:1–2:10. ISBN: 978-1-4503-0223-4. DOI: 10.1145/1938606.1938608. URL: http://doi.acm.org/10.1145/1938606.1938608.

[11] Renjie Liao et al. "A novel serial crime prediction model based on Bayesian learning theory". In: *International Conference on Machine Learning and Cybernetics, ICMLC 2010, Qingdao, China, July 11-14, 2010, Proceedings*. 2010, pp. 1757–1762. DOI: 10.1109/ICMLC.2010.5580971. URL: http://dx.doi.org/10.1109/ICMLC.2010.5580971.

[12] Cynthia Rudin, Daniel Wagner Tong Wang, and Rich Sevieri. "Finding Patterns with a Rotten Core: Data Mining for Crime Series with Core Sets (KDD Workshop Version)". In: (2014). URL: http://dssg.uchicago.edu/kddworkshop/papers/rudin.pdf.

[13] N. Bouhana, S. D. Johnson, and M. Porter. "Consistency and specificity in burglars who commit prolific residential burglary: Testing the core assumptions underpinning behavioural crime linkage. Legal and Criminological Psychology". In: (2014). DOI: 10.1111/lcrp.12050.

[14] Barry Charles Ezell et al. "Probabilistic Risk Analysis and Terrorism Risk". In: *Risk Analysis* 30.4 (2010). DOI: 10.1111/j.1539-6924.2010.01401.x.

[15] Richard Berk. "Algorithmic Criminology". In: *Security Informatics* 2.5 (Jan. 2013). ISSN: 2190-8532. DOI: 10.1186/2190-8532-2-5.

[16] R. A. Berk and J. Bleich. "Statistical Procedures for Forecasting Criminal Behavior". In: *Criminology & Public Policy* 12 (2013), 513544. DOI: 10.1111/1745-9133.12047.

[17] Krishna Pattipati et al. *Hidden Markov Models and Bayesian Networks for Counter-Terrorism*. Wiley-IEEE Press, 2006, pp. 27–50. ISBN: 9780470874103.

[18] Manoj K. Jha. "Dynamic Bayesian Network for Predicting the Likelihood of a Terrorist Attack at Critical Transportation Infrastructure Facilities". In: *Journal of Infrastructure Systems* 15.1 (Mar. 2009), pp. 31–39. DOI: 10.1061/(ASCE)1076-0342(2009)15:1(31).

[19] [Online; accessed 1-Jun-2015]. URL: http://en.wikipedia.org/wiki/Definitions_of_terrorism.

[20] Raffaello Pantucci. "A Typology of Lone Wolves: Preliminary Analysis of Lone Islamist Terrorists". In: *Developments in Radicalisation and Political Violence* (2011).

[21] Virginia Braun and Victoria Clarke. "Using thematic analysis in psychology". In: *Qualitative Research in Psychology* 3.2 (2006), pp. 77–101. DOI: 10.1191/1478088706qp063oa. eprint: http://www.tandfonline.com/doi/pdf/10.1191/1478088706qp063oa. URL: http://www.tandfonline.com/doi/abs/10.1191/1478088706qp063oa.

[22] [Online; accessed 30-January-2015]. URL: http://robotics.stanford.edu/~ronnyk/glossary.html.

[23] Wikipedia. *Feature (machine learning) — Wikipedia, The Free Encyclopedia*. [Online; accessed 30-January-2015]. 2015. URL: http://en.wikipedia.org/w/index.php?title=Feature_(machine_learning)&oldid=644492741.

[24] Wikipedia. *Dimensionality reduction — Wikipedia, The Free Encyclopedia*. [Online; accessed 30-January-2015]. 2014. URL: http://en.wikipedia.org/w/index.php?title=Dimensionality_reduction&oldid=620805391.

[25] Wikipedia. *Feature selection — Wikipedia, The Free Encyclopedia*. [Online; accessed 30-January-2015]. 2014. URL: http://en.wikipedia.org/w/index.php?title=Feature_selection&oldid=639816951.

[26] Wikipedia. *Feature extraction — Wikipedia, The Free Encyclopedia*. [Online; accessed 30-January-2015]. 2014. URL: http://en.wikipedia.org/w/index.php?title=Feature_extraction&oldid=640390842.

[27] Ann E. Nicholson Kevin B. Korb. *Bayesian Artificial Intelligence, Second Edition*. CRC Press, Jan. 2010, pp. 29–33. ISBN: 9781439815915.

[28] [Online; accessed 30-January-2015]. URL: http://en.wikipedia.org/wiki/Bayesian_network.

[29] Andrew Gelman et al. *Bayesian Data Analysis, Third Edition*. Chapman and Hall/CRC, Nov. 2013, p. 1. ISBN: 9781439840955.

[30] Thomas M. Mitchell. *Machine Learning*. 1st ed. New York, NY, USA: McGraw-Hill, Inc., Oct. 1997. ISBN: 0070428077, 9780070428072.

[31] Thomas M. Mitchell. *Machine Learning*. 1st ed. New York, NY, USA: McGraw-Hill, Inc., Oct. 1997, p. 2. ISBN: 0070428077, 9780070428072.

[32] Nello Cristianini and Josh Shawe-Taylor. *An introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, Mar. 2000. ISBN: 0 521 78019 5.

[33] [Online; accessed 1-Jun-2015]. URL: http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#/media/File:KnnClassification.svg.

[34] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. Pittsburgh, Pennsylvania, USA: ACM, 1992, pp. 144–152. ISBN: 0-89791-497-X. DOI: 10.1145/130385.130401. URL: http://doi.acm.org/10.1145/130385.130401.

[35] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, Jan. 2001, p. 15. ISBN: 0262194759.

[36] [Online; accessed 30-January-2015]. URL: http://docs.opencv.org/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html.

[37] [Online; accessed 30-January-2015]. URL: http://www.neural-forecasting.com/support_vector_machines-Dateien/image003.gif.

[38] [Online; accessed 30-May-2015]. URL: http://www.svms.org/parameters/.

[39] Maja Pantic. *Computer Based Coursework Manual Machine Learning (Course 395), Imperial College*. 2013.

[40] Jeff Gruenewald, Steven Chermak, and Joshua D. Freilich. "Distinguishing "loner" attacks from other domestic extremist violence: A comparison of far-right homicide incident and offender characteristics". In: *Criminology & Public Policy* 12.1 (July 2013), pp. 65–91.

[41] Natalia Jaworska and Angelina ChupetlovskaAnastasova. "A Review of Multidimensional Scaling (MDS) and its Utility in Various Psychological Domains". In: *Tutorials in Quantitative Methods for Psychology* 5.1 (2009), pp. 1–10.

[42] A. Mokros and L. J. Alison. "Is offender profiling possible? Testing the predicted homology of crime scene actions and background characteristics in a sample of rapists". In: *Legal and Criminological Psychology* 7.1 (Feb. 2002), pp. 25–43. DOI: 10.1348/135532502168360.

[43] D. Canter. "Offender profiling and criminal differentiation". In: *Legal and Criminological Psychology* 5.1 (Feb. 2000), pp. 23–46. DOI: 10.1348/135532500167958.

[44] Laurence Alison, Craig Bennell, and David Ormerod. "THE PERSONALITY PARADOX IN OFFENDER PROFILING A Theoretical Review of the Processes Involved in Deriving Background Characteristics From Crime Scene Actions". In: *Psychology, Public Policy, and Law* 8.1 (2002), pp. 115–135. DOI: 10.1037//1076-8971.8.1.115.

[45] L. Lefakis and F. Fleuret. "Jointly Informative Feature Selection". In: *Proceedings of the international conference on Artificial Intelligence and Statistics (AISTATS)*. 2014, pp. 567–575. URL: http://fleuret.org/papers/lefakis-fleuret-aistats2014.pdf.

[46] Mark A. Hall. *Correlation-based feature selection for machine learning*. Tech. rep. 1998.

[47] C. Krier et al. "Feature Scoring by Mutual Information for Classification of Mass Spectra". In: *FLINS 2006, 7th International FLINS Conference on Applied Artificial Intelligence*. Genova (Italy), 2006, pp. 557–564.

[48] Elzbieta Pekalska et al. "Pairwise Selection of Features and Prototypes." In: *CORES*. Ed. by Marek Kurzynski et al. Vol. 30. Advances in Soft Computing. Springer, Oct. 2, 2008, pp. 271–278. ISBN: 978-3-540-25054-8. URL: http://dblp.uni-trier.de/db/conf/cores/cores2005.html#PekalskaHLD05.

[49] Hanchuan Peng, Fuhui Long, and Chris Ding. "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2005), pp. 1226–1238.

[50] B. Delaneyy et al. *The Application of Statistical Relational Learning to a Database of Criminal and Terrorist Activity*. 2010, pp. 409–417. ISBN: 978-0-89871-703-7.

[51] Jeffrey Allanach et al. "Detecting, Tracking and Counteracting Terrorist Networks via Hidden Markov Models". In: *Aerospace Conference, 2004. Proceedings* (2004). ISSN: 1095-323X. DOI: 10.1109/AERO.2004.1368130.

[52] K. Baumgartner, S. Ferrari, and G. Palermo. "Constructing Bayesian Networks for Criminal Profiling from Limited Data". In: *Know.-Based Syst.* 21.7 (Oct. 2008), pp. 563–572. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2008.03.019. URL: http://dx.doi.org/10.1016/j.knosys.2008.03.019.

[53] Mohammed M. Olama et al. "A Bayesian Belief Network of Threat Anticipation and Terrorist Motivations". In: *Proc. of SPIE* 7666 (2010). DOI: 10.1117/12.849464.

[54] Kamal Dahbur and Thomas Muscarello. "Classification System for Serial Criminal Patterns". In: *Artif. Intell. Law* 11.4 (Jan. 2003), pp. 251–269. ISSN: 0924-8463. DOI: 10.1023/B:ARTI.0000045994.96685.21. URL: http://dx.doi.org/10.1023/B:ARTI.0000045994.96685.21.

[55] Aixin Sun et al. "Using Support Vector Machines for Terrorism Information Extraction". English. In: *Intelligence and Security Informatics*. Ed. by Hsinchun Chen et al. Vol. 2665. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, pp. 1–12. ISBN: 978-3-540-40189-6. DOI: 10.1007/3-540-44853-5_1. URL: http://dx.doi.org/10.1007/3-540-44853-5_1.

[56] Manoj K. Jha and Ronald A. Keele. "Using Dynamic Bayesian Networks for Investigating the Impacts of Extreme Events". In: (2012). DOI: 10.5772/38568.

[57] Somayeh Shojaee et al. "A study on classification learning algorithms to predict crime status". In: *International Journal of Digital Content Technology and its Applications(JDCTA)* 7.9 (May 2013), pp. 361–369. ISSN: 1975-9339.

[58] [Online; accessed 27-April-2015]. URL: http://www.princeton.edu/politics/about/file-repository/public/Wright_on_Terrorism.pdf.

[59] Faryal Gohar, Wasi Haider Butt, and Usman Qamar. "Terrorist Group Prediction Using Data Classification". In: *Proceedings of the International Conference on Artificial Intelligence and Pattern Recognition, Kuala Lumpur, Malaysia, 2014*. 2014, pp. 199–208. ISBN: 978-1-941968-02-4. URL: http://www.academia.edu/9346283/Terrorist_Group_Prediction_Using_Data_Classification.

[60] [Online; accessed 30-January-2015]. URL: http://www.dtic.mil/doctrine/new_pubs/jp1_02.pdf.

[61] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press Cambridge, Massachusetts London, England: Massachusetts Institute of Technology, 2009. ISBN: 978-0-262-01319-2.

[62] Max Bramer. *Undergraduate Topics in Computer Science*. Springer Science+Business Media, 2007, pp. 72–73. ISBN: 1-84628-765-0. URL: http://lib.mdp.ac.id/ebook/Karya%20Umum/Data-Mining-Undergraduate-Topics.pdf.

[63] P. J. Taylor et al. "Jaccard's heel: Radex models of criminal behaviour are rarely falsifiable when derived using Jaccard coefficient". In: *Legal and Criminological Psychology* 17 (2012), 4158. DOI: 10.1348/135532510X518371.

[64] [Online; accessed 18-April-2015]. URL: http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html.

[65] Lei Xu, Adam Krzyzak, and Ching Y. Suen. "Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition". In: *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS* 22.3 (1992).

[66] [Online; accessed 31-May-2015]. URL: http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html.

[67] [Online; accessed 09-May-2015]. URL: http://scikit-learn.org/stable/modules/linear_model.html.

[68] S. B. Kotsiantis. *Supervised Machine Learning: A Review of Classification Techniques. Informatica 31:249268.* 2007.

[69] O. Pourret, P. Naïm, and B. Marcot. *Bayesian Networks: A Practical Guide to Applications.* Statistics in Practice. Wiley, 2008. ISBN: 9780470994542. URL: http://books.google.co.uk/books?id=KDrRMasDY4oC.

[70] [Online; accessed 30-May-2015]. URL: http://en.wikipedia.org/wiki/Pearson%27s_chi-squared_test.

[71] [Online; accessed 30-May-2015]. URL: http://en.wikipedia.org/wiki/G-test.

[72] Kiran R Karkera. *Building Probabilistic Graphical Models with Python.* Livery Place 35 Livery Street Birmingham B3 2PB, UK: Packt Publishing Ltd., 2014. ISBN: 978-1-78328-900-4.

11

Appendix

A Feature split

Categories	Features
Network	RelatStat, Children, ParRelatStat, Education, OccCat, CrimCon, WarningLettersStatements, LifeAspectChange, Funds, LiveAlone, Virtual, DryRuns, BombManuals, NewMedia,,Interrupt, NotCareInjustice, PersRelat, Financial, Violence, TargetTyp, Stockpile, Regret, Implement, MultiAttackMeth, Discriminate, LettersPost, Getaway, MultiEventTarget, Involve, InteractNet, OtherInv, OtherKnowledge, RecruitNetGroup, FurtherAttacks, ClaimResp
Mental illness	RelatStat, Children, ParRelatStat, Education, OccCat, CrimCon, WarningLettersStatements, AwareGriev, LifeAspectChange, SubAbuse, MentalIll, Isolated, LiveAlone, Virtual, DryRuns, BombManuals, NewMedia, Tipping,,Interrupt, NotCareInjustice, HarmVictimHelpless, Financial, HurtOthers, Stress, SubstanceUse, Violence, TargetTyp, Stockpile, Obsess, Regret, Insanity, Implement, MultiAttackMeth, Discriminate, LettersPost, Getaway, MultiEventTarget, OtherKnowledge, FurtherAttacks, ClaimResp
Ideology	RelatStat, Children, ParRelatStat, Education, OccCat, MilExp, CrimCon, Ideology, Religion, WarningLettersStatements, AwareIdeo, IdeoChangeInt, ReligChangeInt, LifeAspectChange, Legitimise, Denounce, LiveAlone, Adoption, Training, Virtual, DryRuns, BombManuals, WideGroup, NewMedia,Interrupt, NotCareInjustice, Financial, Violence, TargetTyp, LocationNature, LocPubPriv, Stockpile, Contradict, Regret, BeliefChange, Implement, MultiAttackMeth, Discriminate, LettersPost, Getaway, MultiEventTarget, OtherKnowledge, RecruitNetGroup, Propaganda, OwnProp, FurtherAttacks, ClaimResp, PossessStories

Table 1: Features belonging to each of the three themes as well as to the single set of features

B Feature selection

Listing 11.1: Closest features for high value and civilian target respectively as obtained using SPSS analysis for all features

```
python base/single_input_features.py 'all'
Use features from SPSS

########## CIVIL ##########
####### FEATURES ####### 44
['AwareGriev', 'WarningLettersStatements', 'HurtOthers', 'AwareIdeo', 'Religion', '
    Discriminate', 'Propaganda', 'ReligChangeInt', 'TargetTyp', 'Implement', '
    LifeAspectChange', 'MultiEventTarget', 'MilExp', 'RelatStat', 'Funds', '
    FurtherAttacks', 'Stockpile', 'InteractNet', 'Isolated', 'LiveAlone', 'OccCat', '
    MentalIll', 'RecruitNetGroup',  'Violence', 'Tipping', 'Virtual', 'Ideology', '
    Insanity', 'ClaimResp', 'LettersPost', 'WideGroup', 'DryRuns', 'LocationNature',
    'Stress', 'Education', 'PersRelat', 'Involve', 'ParRelatStat', 'Training', '
    IdeoChangeInt', 'SubAbuse', 'OwnProp', 'Adoption', 'BombManuals']
########## HIGHVAL ##########
####### FEATURES ####### 44
['WarningLettersStatements', 'LocationNature', 'Tipping', 'AwareGriev', '
    ReligChangeInt', 'Funds', 'AwareIdeo', 'FurtherAttacks', 'Discriminate', '
    MultiEventTarget', 'HurtOthers', 'Propaganda', 'Implement', 'Virtual', '
    OtherKnowledge', 'IdeoChangeInt', 'Stockpile', 'CrimCon', 'BombManuals', '
    MentalIll', 'InteractNet', 'MilExp', 'RelatStat', 'Adoption', 'ClaimResp', '
    Violence', 'NotCareInjustice', 'TargetTyp', 'RecruitNetGroup', 'LifeAspectChange'
    , 'DryRuns', 'Stress', 'Religion', 'Obsess', 'Ideology', 'Training', 'OtherInv',
    'PersRelat', 'Involve', 'LettersPost', 'OccCat', 'Isolated', 'Insanity', '
    LocPubPriv']
```

Listing 11.2: Closest features for high value and civilian target respectively as obtained using SPSS analysis for thematic features

```
python base/input_features.py
Use features from SPSS

########## HIGHVAL ##########
####### net FEATURES ####### 24
['WarningLettersStatements', 'Funds', 'Discriminate', 'FurtherAttacks', 'Implement',
    'MultiEventTarget', 'Virtual', 'Stockpile', 'OtherKnowledge', 'InteractNet', '
    TargetTyp', 'BombManuals', 'CrimCon', 'RelatStat', 'Violence', 'RecruitNetGroup',
     'ClaimResp', 'LifeAspectChange', 'DryRuns','NotCareInjustice', 'OccCat', '
    OtherInv', 'LettersPost', 'Involve']
####### ill FEATURES ####### 28
['WarningLettersStatements', 'Tipping', 'AwareGriev', 'Discriminate', '
    MultiEventTarget', 'FurtherAttacks', 'Implement', 'HurtOthers', 'OtherKnowledge',
     'Stockpile', 'MentalIll', 'CrimCon', 'BombManuals', 'ClaimResp', 'TargetTyp', '
    Violence', 'NotCareInjustice', 'LiveAlone', 'Stress', 'Insanity', 'Isolated', '
    DryRuns', 'Obsess', 'RelatStat', 'LifeAspectChange', 'OccCat', 'LettersPost', '
    Children']
####### ideo FEATURES ####### 33
['WarningLettersStatements', 'LocationNature', 'Discriminate', 'AwareIdeo', '
    MultiEventTarget', 'ReligChangeInt','FurtherAttacks', 'Implement', 'Propaganda',
    'Virtual', 'OtherKnowledge', 'IdeoChangeInt', 'CrimCon', 'Stockpile', 'MilExp', '
    BombManuals', 'Adoption', 'RelatStat', 'RecruitNetGroup', 'Violence', 'TargetTyp'
    , 'NotCareInjustice', 'LifeAspectChange', 'ClaimResp', 'DryRuns', 'Religion', '
    Children', 'OccCat', 'Training', 'Ideology', 'LettersPost', 'ParRelatStat', '
    WideGroup']
########## CIVIL ##########
####### net FEATURES ####### 24
['WarningLettersStatements', 'MultiEventTarget', 'RelatStat', 'Discriminate', '
    Implement', 'TargetTyp', 'Funds', 'LifeAspectChange', 'Stockpile', '
    FurtherAttacks', 'InteractNet', 'LiveAlone', 'RecruitNetGroup', 'OccCat', '
    Virtual', 'Violence', 'LettersPost', 'ClaimResp', 'ParRelatStat', 'PersRelat', '
    Involve', 'OtherKnowledge', 'Education', 'BombManuals']
####### ill FEATURES ####### 28
['Implement', 'Tipping', 'AwareGriev', 'WarningLettersStatements', 'Isolated', '
    TargetTyp', 'RelatStat', 'HurtOthers', 'Discriminate', 'FurtherAttacks', '
    MultiEventTarget', 'ClaimResp', 'OccCat', 'Stockpile', 'LifeAspectChange', '
    MentalIll', 'CrimCon', 'LiveAlone', 'Insanity', 'Education', 'Virtual', 'DryRuns'
    , 'LettersPost', 'ParRelatStat', 'Violence', 'NewMedia', 'Obsess', 'SubAbuse']
####### ideo FEATURES ####### 33
['WarningLettersStatements', 'AwareIdeo', 'Implement', 'Propaganda', '
    MultiEventTarget', 'MilExp', 'TargetTyp', 'RelatStat', 'Stockpile', 'Discriminate
    ', 'Religion', 'FurtherAttacks', 'RecruitNetGroup', 'LocationNature', 'LiveAlone'
    , 'ReligChangeInt', 'CrimCon', 'WideGroup', 'Virtual', 'IdeoChangeInt', 'OccCat',
     'LifeAspectChange', 'ParRelatStat', 'ClaimResp', 'Financial', 'LettersPost', '
    Violence', 'Ideology', 'NewMedia', 'BombManuals', 'DryRuns', 'Education', '
    Interrupt']
```

C Classification results using cross validation

C.1 Classification on the entire feature set using cross validation

Measure	LR	DT	KNN	SVM
Accuracy	0.576	0.637	0.582	0.644
Highvalue precision	0.560	0.632	0.553	0.648
Highvalue recall	0.498	0.606	0.553	0.595
Highvalue F1	0.509	0.603	0.539	0.601
Civil precision	0.600	0.663	0.622	0.664
Civil recall	0.647	0.667	0.608	0.688
Civil F1	0.611	0.653	0.601	0.664

Table 2: Classification measures for single set of features

Algorithm	Accuracy Majority	Accuracy Weighted Majority
LR	0.513	0.584
DT	0.623	0.704
KNN	0.625	0.700
SVM	0.613	0.648

Table 3: Classification accuracy for thematic sets of features

Measure	LR	DT	KNN	SVM
Highvalue precision	0.472	0.613	0.616	0.577
Highvalue recall	0.449	0.581	0.611	0.826
Highvalue F1	0.449	0.584	0.595	0.631
Civil precision	0.539	0.642	0.662	0.814
Civil recall	0.570	0.657	0.638	0.430
Civil F1	0.544	0.641	0.635	0.500

Table 4: Classification measures for combining the predictions of thematic sets of features with majority voting

Measure	LR	DT	KNN	SVM
Highvalue precision	0.573	0.696	0.684	0.624
Highvalue recall	0.497	0.681	0.695	0.847
Highvalue F1	0.514	0.675	0.675	0.673
Civil precision	0.607	0.735	0.742	0.853
Civil recall	0.660	0.724	0.703	0.472
Civil F1	0.623	0.717	0.708	0.544

Table 5: Classification measures for combining the predictions of thematic sets of features with weighted majority voting

C.2 Classification on subset of features using cross validation

C.2.1 Single set of features

Algorithm	Accuracy 90%	Accuracy 70%	Accuracy 50%
LR	0.537	0.550	0.539
DT	0.627	0.627	0.610
KNN	0.642	0.635	0.624
SVM	0.659	0.615	0.628

Table 6: Classification accuracy for subset of single set of features

Measure	LR	DT	KNN	SVM
Highvalue precision	0.514	0.606	0.623	0.801
Highvalue recall	0.517	0.591	0.648	0.384
Highvalue F1	0.503	0.586	0.626	0.485
Civil precision	0.564	0.663	0.676	0.634
Civil recall	0.555	0.659	0.638	0.900
Civil F1	0.551	0.648	0.647	0.737

Table 7: Classification measures for single set of features when selecting features that appear in at least 90% folds

Measure	LR	DT	KNN	SVM
Highvalue precision	0.511	0.610	0.646	0.584
Highvalue recall	0.530	0.586	0.529	0.805
Highvalue F1	0.507	0.585	0.562	0.630
Civil precision	0.593	0.655	0.652	0.764
Civil recall	0.567	0.661	0.728	0.447
Civil F1	0.568	0.648	0.675	0.500

Table 8: Classification measures for single set of features when selecting features that appear in at least 70% folds

Measure	LR	DT	KNN	SVM
Highvalue precision	0.503	0.592	0.617	0.726
Highvalue recall	0.531	0.569	0.541	0.301
Highvalue F1	0.505	0.566	0.560	0.385
Civil precision	0.579	0.641	0.644	0.614
Civil recall	0.547	0.645	0.696	0.914
Civil F1	0.550	0.628	0.657	0.717

Table 9: Classification measures for single set of features when selecting features that appear in at least 50% folds

C.2.2 Thematic sets of features

Algorithm	Accuracy Majority	Accuracy Weighted Majority
LR	0.510	0.581
DT	0.610	0.704
KNN	0.682	0.751
SVM	0.653	0.722

Table 10: Classification accuracy for thematic sets of features when selecting feaures that appear in at least 90% folds

Measure	LR	DT	KNN	SVM
Highvalue precision	0.480	0.586	0.689	0.733
Highvalue recall	0.505	0.572	0.637	0.478
Highvalue F1	0.479	0.568	0.647	0.541
Civil precision	0.546	0.640	0.698	0.650
Civil recall	0.515	0.644	0.721	0.807
Civil F1	0.516	0.632	0.701	0.708

Table 11: Classification measures for combining the predictions of thematic sets of features with majority voting when selecting features that appear in at least 90% folds

Measure	LR	DT	KNN	SVM
Highvalue precision	0.560	0.711	0.751	0.775
Highvalue recall	0.549	0.666	0.725	0.601
Highvalue F1	0.542	0.672	0.721	0.641
Civil precision	0.612	0.728	0.783	0.731
Civil recall	0.610	0.737	0.774	0.829
Civil F1	0.600	0.721	0.767	0.761

Table 12: Classification measures for combining the predictions of thematic sets of features with weighted majority voting when selecting features that appear in at least 90% folds

Algorithm	Accuracy Majority	Accuracy Weighted Majority
LR	0.531	0.610
DT	0.619	0.688
KNN	0.691	0.750
SVM	0.637	0.700

Table 13: Classification accuracy for thematic sets of features when selecting feaures that appear in at least 70% folds

Measure	LR	DT	KNN	SVM
Highvalue precision	0.494	0.608	0.685	0.736
Highvalue recall	0.502	0.575	0.661	0.334
Highvalue F1	0.483	0.577	0.659	0.427
Civil precision	0.569	0.646	0.722	0.618
Civil recall	0.556	0.657	0.719	0.902
Civil F1	0.551	0.640	0.710	0.726

Table 14: Classification measures for combining the predictions of thematic sets of features with majority voting when selecting features that appear in at least 70% folds

Measure	LR	DT	KNN	SVM
Highvalue precision	0.587	0.692	0.759	0.807
Highvalue recall	0.593	0.651	0.715	0.482
Highvalue F1	0.578	0.655	0.723	0.571
Civil precision	0.645	0.710	0.770	0.677
Civil recall	0.627	0.719	0.781	0.893
Civil F1	0.626	0.704	0.766	0.762

Table 15: Classification measures for combining the predictions of thematic sets of features with weighted majority voting when selecting features that appear in at least 70% folds

Algorithm	Accuracy Majority	Accuracy Weighted Majority
LR	0.543	0.622
DT	0.616	0.684
KNN	0.670	0.729
SVM	0.613	0.693

Table 16: Classification accuracy for thematic sets of features when selecting feaures that appear in at least 50% folds

Measure	LR	DT	KNN	SVM
Highvalue precision	0.522	0.601	0.660	0.673
Highvalue recall	0.513	0.595	0.655	0.256
Highvalue F1	0.499	0.583	0.645	0.340
Civil precision	0.575	0.648	0.699	0.599
Civil recall	0.572	0.635	0.683	0.926
Civil F1	0.560	0.628	0.681	0.717

Table 17: Classification measures for combining the predictions of thematic sets of features with majority voting when selecting features that appear in at least 50% folds

Measure	LR	DT	KNN	SVM
Highvalue precision	0.613	0.685	0.733	0.769
Highvalue recall	0.599	0.656	0.707	0.443
Highvalue F1	0.587	0.654	0.705	0.530
Civil precision	0.657	0.711	0.755	0.670
Civil recall	0.645	0.711	0.749	0.912
Civil F1	0.637	0.698	0.740	0.764

Table 18: Classification measures for combining the predictions of thematic sets of features with weighted majority voting when selecting features that appear in at least 50% folds

D Classification on new data set

D.1 Target choice

	LR	DT	KNN	SVM
Accuracy	0.800	0.700	0.700	0.900
High value Precision	0.500	0.400	0.333	1.000
High value Recall	0.500	1.000	0.500	0.500
High value F1	0.500	0.571	0.400	0.666
Civil Precision	0.875	1.000	0.857	0.888
Civil Recall	0.875	0.625	0.750	1.000
Civil F1	0.875	0.769	0.800	0.941

Table 19: Classification measures for single set of features

	LR	DT	KNN	SVM
Accuracy	0.600	0.900	0.700	0.800
High value Precision	0.333	0.666	0.333	0.000
High value Recall	1.000	1.000	0.500	0.000
High value F1	0.500	0.800	0.400	0.000
Civil Precision	1.000	1.000	0.857	0.800
Civil Recall	0.500	0.875	0.750	1.000
Civil F1	0.666	0.933	0.800	0.888

Table 20: Classification measures using a single set of features that appear in at least 90% folds

	LR	DT	KNN	SVM
Accuracy	0.600	0.800	0.600	0.800
High value Precision	0.250	0.500	0.250	0.000
High value Recall	0.500	1.000	0.500	0.000
High value F1	0.333	0.666	0.333	0.000
Civil Precision	0.833	1.000	0.833	0.800
Civil Recall	0.625	0.750	0.625	1.000
Civil F1	0.714	0.857	0.714	0.888

Table 21: Classification measures for combining the predictions of thematic sets of features with weighted majority voting using all features

	LR	DT	KNN	SVM
Accuracy	0.600	0.700	0.500	0.400
High value Precision	0.250	0.333	0.200	0.166
High value Recall	0.500	0.500	0.500	0.500
High value F1	0.333	0.400	0.285	0.250
Civil Precision	0.833	0.857	0.800	0.750
Civil Recall	0.625	0.750	0.500	0.375
Civil F1	0.714	0.800	0.615	0.500

Table 22: Classification measures for combining the predictions of thematic sets of features with weighted majority voting when selecting features that appear in at least 90% folds

D.2 Discriminate

	LR	DT	KNN	SVM
Accuracy	0.600	0.700	0.500	0.600
Negative Precision	0.750	1.000	0.666	0.750
Negative Recall	0.500	0.500	0.333	0.500
Negative F1	0.600	0.666	0.444	0.600
Positive Precision	0.500	0.571	0.428	0.500
Positive Recall	0.750	1.000	0.750	0.750
Positive F1	0.600	0.727	0.545	0.600

Table 23: Classification measures for single set of features

	LR	DT	KNN	SVM
Accuracy	0.600	0.900	0.500	0.700
Negative Precision	0.666	1.000	0.666	0.800
Negative Recall	0.666	0.833	0.333	0.666
Negative F1	0.666	0.909	0.444	0.727
Positive Precision	0.500	0.800	0.428	0.600
Positive Recall	0.500	1.000	0.750	0.750
Positive F1	0.500	0.888	0.545	0.666

Table 24: Classification measures using a single set of features that appear in at least 90% folds

	LR	DT	KNN	SVM
Accuracy	0.700	0.600	0.600	0.800
Negative Precision	0.666	0.750	0.750	0.750
Negative Recall	1.000	0.500	0.500	1.000
Negative F1	0.800	0.600	0.600	0.857
Positive Precision	1.000	0.500	0.500	1.000
Positive Recall	0.250	0.750	0.750	0.500
Positive F1	0.400	0.600	0.600	0.666

Table 25: Classification measures for combining the predictions of thematic sets of features with weighted majority voting using all features

	LR	DT	KNN	SVM
Accuracy	0.600	0.800	0.400	0.700
Negative Precision	0.666	0.833	0.500	0.666
Negative Recall	0.666	0.833	0.333	1.000
Negative F1	0.727	0.833	0.400	0.800
Positive Precision	0.500	0.750	0.333	1.000
Positive Recall	0.500	0.750	0.500	0.250
Positive F1	0.500	0.750	0.400	0.400

Table 26: Classification measures for combining the predictions of thematic sets of features with weighted majority voting when selecting features that appear in at least 90% folds

D.3 Violence

	LR	DT	KNN	SVM
Accuracy	0.700	0.800	0.500	0.500
Negative Precision	0.500	0.666	0.250	0.250
Negative Recall	0.666	0.666	0.333	0.333
Negative F1	0.571	0.666	0.285	0.285
Positive Precision	0.833	0.857	0.666	0.666
Positive Recall	0.714	0.857	0.571	0.571
Positive F1	0.769	0.857	0.615	0.615

Table 27: Classification measures for single set of features

	LR	DT	KNN	SVM
Accuracy	0.300	0.700	0.500	0.600
Negative Precision	0.250	0.500	0.250	0.333
Negative Recall	0.666	0.666	0.333	0.333
Negative F1	0.363	0.571	0.285	0.333
Positive Precision	0.500	0.833	0.666	0.714
Positive Recall	0.142	0.714	0.571	0.714
Positive F1	0.222	0.769	0.615	0.714

Table 28: Classification measures using a single set of features that appear in at least 90% folds

	LR	DT	KNN	SVM
Accuracy	0.600	0.700	0.600	0.700
Negative Precision	0.333	0.500	0.000	0.500
Negative Recall	0.333	0.666	0.000	0.666
Negative F1	0.333	0.571	0.000	0.571
Positive Precision	0.714	0.833	0.666	0.833
Positive Recall	0.714	0.714	0.857	0.714
Positive F1	0.714	0.769	0.750	0.769

Table 29: Classification measures for combining the predictions of thematic sets of features with weighted majority voting using all features

	LR	DT	KNN	SVM
Accuracy	0.600	0.400	0.600	0.500
Negative Precision	0.333	0.285	0.000	0.250
Negative Recall	0.333	0.666	0.000	0.333
Negative F1	0.333	0.400	0.000	0.285
Positive Precision	0.714	0.666	0.666	0.666
Positive Recall	0.714	0.285	0.857	0.571
Positive F1	0.714	0.400	0.750	0.615

Table 30: Classification measures for combining the predictions of thematic sets of features with weighted majority voting when selecting features that appear in at least 90% folds

E Classification on new data set using features from Proxscal

E.1 Single feature set

	LR	DT	KNN	SVM
Civil Precision	0.551	0.663	0.624	0.592
Civil Recall	0.581	0.644	0.672	0.920
Civil F1	0.555	0.638	0.636	0.712
High value Precision	0.514	0.552	0.585	0.654
High value Recall	0.459	0.519	0.526	0.276
High value F1	0.468	0.518	0.539	0.364

Table 31: Results using the entire feature set as resulted from Multidimensional Scaling

E.2 Thematic feature set

Majority	LR	DT	KNN	SVM
High value precision	0.494	0.569	0.597	0.716
High value recall	0.438	0.550	0.599	0.398
High value F1	0.444	0.544	0.586	0.485
Civilian precision	0.527	0.634	0.611	0.596
Civilian recall	0.583	0.647	0.665	0.794
Civilian F1	0.544	0.629	0.626	0.670

Table 32: Results using the thematic feature sets as resulted from Multidimensional Scaling. The outputs of the three themes are combined using majority voting

Weighted Majority	LR	DT	KNN	SVM
High value precision	0.600	0.662	0.679	0.771
High value recall	0.534	0.633	0.671	0.506
High value F1	0.546	0.632	0.658	0.577
Civilian precision	0.605	0.693	0.709	0.641
Civilian recall	0.654	0.711	0.721	0.841
Civilian F1	0.620	0.692	0.702	0.713

Table 33: Results using the thematic feature sets as resulted from Multidimensional Scaling. The outputs of the three themes are combined using weighted majority voting