A Neural Network for Counter-Terrorism

S.J. Dixon, M.B. Dixon, J. Elliott, E. Guest and D. J. Mullier[1]

Abstract This article presents findings concerned with the use of neural networks in the identification of deceptive behaviour. A game designed by psychologists and criminologists was used for the generation of data used to test the appropriateness of different AI techniques in the quest for counter-terrorism. A feed forward back propagation network was developed and subsequent neural network experiments showed on average a 60% success rate and at best a 68% success rate for correctly identifying deceptive behaviour. These figures indicate that, as part of an investigator support system, a neural network would be a valuable tool in the identification of terrorists prior to an attack.

1 Introduction

DScent was a joint project between five UK universities combining research theories in the disciplines of computational inference, forensic psychology and expert decision-making in the area of counter-terrorism. This article discusses the findings of research and development around the role and the use of neural networks as a tool for identifying deception in the field of counter-terrorism.

For the purpose of data generation and system testing the project team devised a closed world game called "Cutting Corners". This game was used as a test-bed to allow development, application and validation of an artificial intelligence (AI) approach for identifying deceptive behaviour. Within the game participants acted as teams and traversed certain locations using GPS enabled devises to communicate, navigate and purchase items. The game participants either acted as *potentially dishonest*[2] builders who were constructing part of an Olympic stadium, or terrorists masquerading as builders with the aim of planting explosives. The game was divided into rounds with a certain number of dice throws per player and the winner was the first to accomplish their aim.

Each game consisted of four players with between one and three players acting as terrorists. During the game the players could visit three different types of virtual

[1] All Leeds Metropolitan University, s.j.dixon@leedsmet.ac.uk, m.dixon@leedsmet.ac.uk, j.elliott@leedsmet.ac.uk, e.guest@leedsmet.ac.uk, d.mullier@leedsmet.ac.uk

[2] Participants taking on the role of builders during the game trials were encouraged to 'bend the rules' slightly, therefore concealing the deceptive behaviour of the terrorist data within a game.

location: the Builders Yard, selling virtual construction blocks, soil and fertiliser; the Electronics Store, selling virtual wiring and dynamite and the Olympic Site, where virtual items could be unloaded. An initial amount of virtual cash and a virtual van was given to each player at the beginning of a game. During the game van searches and van weight checks were carried out where players displayed two items in their van and were weighed respectively. If the van exceeded the maximum weight allowance the player did not receive a cash reward. On completion of each round the sum of items sold from each shop was calculated.

An investigator support system, known as DScentTrail [1], was developed that presented graphical *scent trails*[3] of a suspect over time. This support system was underpinned by a neural network to help identify and highlight deceptive behaviour. Preliminary work was carried out on a behavioural based AI module which would work separately alongside the neural network, with both identifying deception before integrating their results to update DScentTrail.

2 AI Techniques for Counter-Terrorism

The use of various AI techniques, such as data mining, artificial neural networks, symbolic AI and Case Based Reasoning for counter-terrorism have been advocated by Markman [2] and Marappan [3]. Schneier [4] however, in his article on *Why Data Mining Won't Stop Terror*, writes that data mining works best when you're searching for a well-defined profile, a reasonable number of attacks per year and a low cost of false alarms. Rudmin [5] is also sceptical regarding the use of data mining techniques and disregards them completely as in order to make a Bayesian computation, he estimates that at best in the USA there would be a base-rate of 1 terrorist per 300,000 people and that if a surveillance monitoring system had an accuracy rate of 40% positive identification of real terrorists then according to Bayes' Theorem the misidentification rate would be .01%, or 30,000 innocent people. Rudmin stresses that these numbers are simply examples based on one particular technology. Jonas and Harper [6] in their report on *Effective Counterterrorism and the Limited Role of Predictive Data Mining* agree with Rudmin regarding the unacceptable number of likely false positives, they state that it would be a waste of resources and a threat to civil liberties. In addition to the high number of false positives, they argue against the usefulness of predictive data mining for counterterrorism due to the absence of terrorism patterns, leaving it impossible to develop useful algorithms.

Data mining was not used on the DScent project since it is generally used for extracting information from large quantities of data that is collected for reasons other than for the purpose of mining itself. The DScent data was explicitly designed and collected for identifying suspicious behaviour. DScent would not encounter the problems outlined by Rudmin, Jonas or Harper of having to

[3] A scent trail within the project is a collection of ordered, relevant behavioural information over time for a suspect.

potentially question a huge number of innocent people as the set did not contain the entire population, it was merely a well established sub-set. Ware [7] states that neural networks do not lend themselves easily to real-time updated information, and has concerns regarding the limited availability of historical data. Although Ware's observations may be valid, by identifying the key input factors to the neural network and keeping these to an absolute minimum, the amount of historical data required for training will be far less. Furthermore, if the neural network can identify deception amongst humans from a small amount of inputs then we are getting closer to that *well-defined profile* of which Schneier speaks.

The choice of a neural network was made as it is the most likely technique that will work with a non-polynomial problem such as behavioural patterns of humans. Jonas and Harper are correct when they state that it is impossible to design algorithms if no differences exist in terrorist and non-terrorist behaviours, though the project team believe that differences may exist. A neural network was chosen at this stage to identify whether these differences did occur. Preliminary work within DScent has paved the way for further research into this area which, providing differences in behaviour can be identified, will include the development of a hybrid AI system including both a neural network and a behavioural based AI module.

3 Development

Feed forward back-propagation neural networks were developed using the JOONE toolset [8] which is an object based neural network framework with a graphical user interface. EasyNN-plus [9] was used to validate the output from Joone. The neural network architecture took the input data from an Excel spreadsheet entering the input layer containing 122 neurons, the data progressed to a hidden layer containing 10 neurons, before it finally reached the output layer which contained a single neuron. The output value was in the range zero to one and was passed into an Excel spreadsheet, all three layers utilised the sigmoid activation function [10]. The Teacher layer trained the network by presenting it with complete examples, including whether the example was a terrorist or not (this is known as supervised learning). The training was then presented graphically via a Root Mean Square Error chart (RMSE) [11] examples of which are presented within the DScent Final Report [1].

The data from the Cutting Corners board game was collated into an Excel spread sheet. The spread sheet contained 144 rows of game data which resulted from playing 36 games. This game data was divided into separate training and test files with a ratio of 4:1 respectively. Three types of training and three types of test files, each containing varying numbers of terrorists, were created for each variation on the input file.

The effectiveness of a neural network is greatly reduced when the number of variables (horizontal), do not have adequate training pattern examples (vertical), as the network does not have the opportunity to explore a large proportion of the

possibilities. It is therefore necessary to prune the input file of unnecessary variables prior to training. It is apparent that by knowing which variables are contributing to the neural network [12] the developer has not only improved the effectiveness of the networks ability to generalise but also gains a better understanding of the problem. Experiments were performed excluding different variables within the import file to enable the ultimate level of accuracy given the number of training patterns available.

Due to the severe lack of training data the results were predictably inaccurate, though much better than anticipated. This did not however present a problem, as the purpose of phase one was to experiment with different tools, architectures, input variables, the ratio of positive and negative patterns presented within the training and test files and to identify the optimal classification threshold within the output. A total of 55 neural network experiments were. A threshold of 0.5 was used as the cut-off point, where a value of 0 indicated 'builder' and a value of 1 indicated 'terrorist', therefore any result greater than or equal to 0.5 was deemed to be a terrorist. The RMSE was plotted for each experiment during training to establish the optimum number of times the neural network was presented with the entire training set, known as an epoch. It is crucial not to over train the network as it has then the potential to memorise the training data and therefore looses the ability to generalise with different data.

The Mann Whitney U test [13] was used to ascertain whether differences between two sets of result data could not have occurred by chance alone. Firstly, the least successful set of neural network results were compared against the most successful set. Secondly, the most successful set were compared against a random set of 28 zeros and ones. An online automated calculation tool [14] was used to perform the final part of the tests, as significance lookup tables do not have U values beyond 30; these results are shown below:

- Test 1: The two samples are not significantly different ($P>=0.05$, two-tailed test).
- Test 2: The difference between the two samples is highly significant ($P < 0.001$, two-tailed test).

These tests prove the value of the neural network even with such small amounts of training data. Altering the threshold to determine whether an output was positive or negative had a direct effect on the success rate of the network. If the initial threshold value of 0.5 was shifted down to a value of 0.13 the number of true positives was increased from 53% to 60%. This had a slightly negative effect on the total number of correct classifications within the test files, taking the percentage down from 64% to 60%. This percentage loss was deemed acceptable, as it was not identifying terrorists from the entire population, but identifying individuals who merited further investigation from a preselected subset who where under suspicion. This was identified as a suitable capability by the stakeholders[4] when consulted regarding functionality for the system.

---

[4] The project was funded by the EPSRC, grant number: EP/F014112/1. Stakeholders included a number of interested personnel from the CPNI and the MoD.

4 Results and Conclusions

The experiments showed on average a 60% success rate (68% peek) for correctly identifying terrorist behaviour. The winning architecture consisted of all three layers; input, hidden and output using the sigmoid activation function. The hidden layer contained 10 neurons which resulted in 11% of the number of variables contained within the input file. The information variables which proved to be of importance were 'locations', 'Stock Items' and 'Stock Take'. Excluded variables were 'Game Number', 'Colour' and 'Van Weight'. The patterns within the training file were presented to the neural network randomly rather than in sequence using over 1500 epochs.

Certain rows within the input file were consistently classified either correctly or incorrectly, obtaining either a minimum of a 90 percent success rate or a maximum of 10 percent success rate throughout all 50 neural network experiments. The proportion of these successful and unsuccessful rows that were terrorist patterns of behaviour was 14% and 71% respectively. After analysing these rows it was apparent that the neural network had generalised much better for the builders, this was as expected given there were more builder examples in the training files. From the correctly identified terrorist rows, the neural network performed far better for those who used dynamite to carry out the tasks rather than those using fertiliser, again due to more terrorists using dynamite. Not all games were played in full; they ended when a player won, which is another reason for the neural network incorrectly classifying records. The next stage of development would have been to introduce the concept of pattern completeness; this would be to train and refine the neural network on patterns with varying degrees of completeness and identify chunks of behaviour which were deceptive in isolation. This type of discrete deception identification would be far more valuable in reality.

Problem domains such as counter-terrorism intrinsically contain many information variables. Each time a variable is added, the number of possible pattern combinations increases exponentially. Therefore, with 100 variables within the input file, a vast number of rows would be required to cover just a small number of possible combinations of data. Take for example the winning neural network where only location information, stock items and stock take information was used (92 variables), each variable had an average of four possible values, i.e. 4^{92}, resulting in 2.45×10^{55} rows of training data required to cover every possible combination. This poses a problem, as large numbers of historical patterns of terrorist behaviour are not available.

Overall the neural network showed extremely promising results taking into account the sparse amount of training data. Future work is underway to develop a method for generating behavioural data, building on the rules of the board game. This is planned to be done by combining intelligent agents [15] with gene expression programming [16] and the use of an Emdros database [17].

A neural network has great potential in the quest to aid counter-terrorism, though certain pre-requisites must be met. These include providing an adequate set

of training data; identification of an optimal results classification threshold; and performing pre-processing to undertake tasks with which neural networks have difficulty, such as cross referencing rows against column data.

References

1. Dixon, S., Guest, E., Dixon, M., Elliott, J., Mullier, D.: DScent Final Report (2011). Available from: http://www.leedsmet.ac.uk/aet/computing/Computing_DScentFinalReport v2_0_2011.pdf Accessed 8 March 2011.
2. Markman, A., Rachkovskij, D., Misuno, I., Revunova, E.: Analogical Reasoning Techniques in Intelligent Counterterrorism Systems. International Journal "Information Theories & Applications" Volume 10 (2) (2003).
3. Marappan, K., Nallaperumal, K., Kannan, S., Bensujin, B.: A Soft Computing Model to Counter Terrorism. IJCSNS International Journal of Computer Science and Network Security, Volume 8 May 2008, p.141. (2008).
4. Schneier, B.: Why Data Mining Won't Stop Terror. Wired (2006). Available from: http://www.wired.com/politics/security/commentary/securitymatters/2006/03/70357 Accessed 28 October 2010.
5. Rudmin, F.: Why Does the NSA Engage in Mass Surveillance of Americans When It's Statistically Impossible for Such Spying to Detect Terrorists? Counterpunch (2006). Available from:http://www.counterpunch.org/rudmin05242006.html Accessed 1 Aug 2011.
6. Jonas, J., Harper**,** J.: Effective counterterrorism and the limited role of predictive data mining**.** CATO Institute (2006) Available from: http://www.cato.org/pubs/pas/pa584.pdf Accessed 27 October 2010.
7. Ware, B.S., Beverina, A., Gong, L., Colder, B.: A Risk-Based Decision Support System for Antiterrorism (2002). Available from: http://www.dsbox.com/Documents/MSS_A_Risk-Based_Decision_Support_System_for_Antiterrorism.pdf Accessed 30 Jan 2009
8. Marrone, P.: An Object Oriented Neural Engine. SourceForge (2010). Available from: http://www.jooneworld.com Accessed 26 October 2010.
9. Neural Planner Software: EasyNN-plus - Neural Network Software. (2010). Available from: http://www.easynn.com Accessed 26 October 2010.
10. Mitchell, T.M.: Machine Learning, WCB-McGraw-Hill (1997).
11. Levinson, N.: The Wiener RMS (ROOT MEAN SQUARE) ERROR Criterion in Filter Design and Prediction. Journal of Mathematics and Physics (1946).
12. Sexton, R.S., Sikander, N.A.: Data mining using a genetic algorithm-trained neural network. Intelligent Systems in Accounting, Finance and Management 10(4):201-210. doi:10.1002/isaf.205 (2002).
13. Mann, H.B., Whitney, D.R.:On a test of whether one of two random variables is stochastically larger than the other. Annals of Mathematical Statistics Volume 18 March 1947, p.50–60 (1947)
14. Avery, L.: Mann-Whitney U Test (2007). Available from: http://elegans.swmed.edu/~leon/stats/utest.html Accessed 27 October 2010.
15. Evertsz, R.: Populating VBS2 with Realistic Virtual Actors. Proceedings of the 18th Conference on Behavior Representation in Modeling and Simulation, p.1–8. (2009).
16. Ferreira, C.: Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence (Studies in Computational Intelligence). 2nd ed. Springer (2006).
17. Petersen, U.: Emdros: a text database engine for analyzed or annotated text. Proceedings of the 20th international conference on Computational Linguistics (2004). Available from: http://emdros.org/petersen-emdros-COLING-2004.pdf Accessed 24 January 2011.