INVESTIGATION OF COMPUTATIONAL INTELLIGENCE TECHNIQUES FOR INTRUSION DETECTION IN WIRELESS SENSOR NETWORKS

Christopher D. McDermott and Andrei Petrovski

School of Computing Science and Digital Media, Robert Gordon University, Aberdeen, Scotland

ABSTRACT

Wireless Sensor Networks (WSNs) have become a key technology for the IoT and despite obvious benefits, challenges still exist regarding security. As more devices are connected to the internet, new cyber attacks are emerging which join well-known attacks posing significant threats to the confidentiality, integrity and availability of data in WSNs. In this work, we investigated two computational intelligence techniques for WSN intrusion detection. A back propagation neural network was compared with a support vector machine classifier. Using the NSL-KDD dataset, detection rates achieved by the two techniques for six cyber attacks were recorded. The results showed that both techniques offer a high true positive rate and a low false positive rate, making both of them good options for intrusion detection. In addition, we further show the support vector machine classifiers suitability for anomaly detection, by demonstrating its ability to handle low sample sizes, while maintaining an acceptable FPR rate under the required threshold.

KEYWORDS

Wireless Sensor Networks (WSNs), Intrusion Detection System (IDS), Denial of Service (DoS), Artificial Neural Network (ANN), Feed-forward Backpropagation, Support Vector Machine (SVM).

1. INTRODUCTION

The Internet of Things (IoT) is expected to usher in an era of increased connectivity, with an estimated 50 billion devices expected to be connected to the Internet by 2020 [1]. At its core, the aim of the IoT is to connect previously unconnected devices to the Internet [2], thus creating smart devices capable of collecting, storing and sharing data, without human interaction [3]. These newly connected smart devices join previously connected traditional computing devices, to form a hybrid network known as the IoT.

With the rapid growth of the IoT, and the technological development of sensors, Wireless Sensor Networks (WSNs) have become a key technology for the IoT [4]. These networks consist of self-organized sensor nodes, communicating using a wireless medium and are used to perform distributed sensing tasks. The low cost nature and easy implementation has seen them deployed in a wide range of fields such as surveillance, climate change detection, environment monitoring, and numerous healthcare applications [5].

Despite the obvious benefits, challenges still exist with respect to security in WSNs. This is largely due to the inherent nature of being deployed in harsh unattended environments, broadcast in nature, and having limited resources [6] [7]. To address these concerns, extensive research has been conducted into the use of cryptography, authentication, key management and secure routing in WSNs. Whilst the proposed security solutions have been found to reduce cyber attacks, they

have not eliminated them completely [8]. Well-known and new cyber attacks therefore remain a concern in WSNs, and are the rationale for this study.

Computational intelligence provides comparatively low-cost technologies for developing IDSs while taking care of limited resource consumption. In this study, we applied two major computational intelligence techniques: feed-forward backpropagation multi-layer perception, and support vector machine (SVM) in WSN intrusion detection and accuracy recording. More specifically, our study compared the detection rates of Denial of Service (DoS) intrusions achieved by the two techniques. A public dataset is preprocessed, normalised and used as input to each network before detection rates and accuracy are compared. Based on our experimental comparison, we found that both techniques performed well, returning good true positive detection rates, while the SVM offered the better false positive rate. This demonstrates that both techniques could be used for establishing anIDS to reduce cyber attacks in WSNs.

The remainder of this paper is organised as follows. Section 2 presents related work in this field. Section 3 discusses well-known cyber attacks faced by WSNs found in the IoT. Section 4 describes common methods and approaches to detecting cyber attacks. Section 5 discusses artificial neural networks as a method for detecting cyber attacks. Section 6 discusses the use of Support Vector Machines as a method for detecting cyber attacks. Section 7 details the experimental setup used to provide a comparison between the detection rates of the feed-forward neural network and support vector classifier. The achieved results are presented in Section 8. Finally, Section 9 provides some concluding remarks with suggestions for future work.

2. RELATED WORK

Wireless sensor networks (WSNs) are by nature distributed, fault tolerant, scalable, and function without a predefined infrastructure [5]. These attributes make it very difficult to design and maintain a WSN which is completely secure and resistant to threats from cyber attacks. Focus has therefore shifted to detecting threats in a timely manner to minimize their impact on the network. Various approaches to intrusion detection in WSN have been proposed including the use of Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs). In [9] a hybrid IDS using support vector machine and is proposed and designed to operate in cluster based WSNs. The SVM is trained using a distributed learning algorithm and returns high detection rates, with a low number of false positives recorded. In [10] the authors use a SVM for intrusion detection and use the Dynamically Growing Self-Organising Tree (DGSOT) clustering algorithm to enhance performance. They demonstrate the approach provides a significant improvement during the training process, outperforms the Rocchio Bundling technique, and offers good detection rates. In [11] new data preprocessing techniques are proposed and tested using various data mining algorithms, including SVM. In their results, the authors found the SVM classifier to offer the best performance (high accuracy) with minimum computational time required, when using datasets with sparse representation.

Research has also been conducted into the use of artificial neural networks for intrusion detection. In [12] the authors propose an artificial neural network based intrusion detection system and test the system with 22 attack types found in the KDD99 data set. Their results show a 75% success rate for most attacks, although attacks with fewer samples return substantially lower rates. In [13] Genetic Algorithm-Levenberg-Marquardt Algorithm (GA-LMBP) is used for the first time in an WSN intrusion detection model. When compared with traditional models based on BP, it offers multilayer cooperative detection and a model of self-leaning with associative memory and fuzzy computing abilities, and offers higher true positive detection rates and lower power consumption. In [14] the authors use an enhanced Fuzzy Adaptive Resonance Theory (ART) neural network to detect threats in WSNs. The original model proposed in [15] was enhanced by training the

network to learn a time-series and detect time-related changes, resulting in higher accuracy rates when compared with the original.

3. CYBER ATTACKS IN WSN

Although cyber attacks facing WSNs are diverse, they can generally be classified as either active or passive attacks. Passive attacks do not characteristically impair a network or change data, but rather monitor *targets for vulnerabilities* or extract information from the network. Active attacks by contrast attempt to change data on or en route to a target, or deprive access to the network using various forms of denial of service (DoS) attacks. This section will discuss known active cyber attacks facing WSNs, used to change data or perform DoS attacks.

- *Hello Flood Attack:* An attacker floods the network with Hello packets so nodes select the compromised node as a forwarder for their packets [16].
- *Sybil Attack:* A compromised node adopts the identity of several other nodes, or creates fake identifies with the intent of disrupting routing paths or data aggregation [17].
- *Wormhole Attack:* A malicious node advertises itself as having the shortest routing path to the base node. All other nodes therefore select it as a forwarder for their packets, resulting in all traffic now passing through the malicious node and being tunneled through a private link to another location.
- *Sink Hole / Black Hole Attack:* A compromised node advertises itself as having the shortest routing path to the base node [17]. All other nodes therefore select it as a forwarder for their packets, resulting in the creation of a sinkhole, and all traffic now passing through the compromised node.
- *Selective Forwarding Attack:* A compromised node exploits inherent trust, whereby a compromised node refuses to forward certain packets and simply drops them.
- Misdirection Attack: A compromised node forwards messages along incorrect paths, in an attempt to divert traffic away from its intended destination.
- *Desynchronisation Attack:* Messages carrying sequence numbers are disrupted, misleading end nodes into thinking some frames have been missed and therefore requesting retransmission [16].
- *SYN Flood Attack:* A network is flooded with malicious SYN request packets creating copious half-open state connections between nodes. In the absence of required ACK reply packets, the nodes resources are exhausted and denial of service is achieved [18].
- *Collision Attack:* A compromised node transmits short noise packets when other nodes are already transmitting, causing a collision with a neighbouring node.
- *Exhaustion Attack:* A compromised node repeatedly sends a RTS message, eliciting a CTS response, which if done continuously would eventually exhaust the resources of both nodes.
- *Unfairness Attack:* Compromised nodes monopolise access to the channel, reducing window time for genuine data transmission. As a result, service although not entirely denied, is significantly degraded.
- *Tampering Attack:* Malicious beacons are continuously sent to sensor nodes, keeping them active and preventing them from switching to sleep mode.
- *Battery Exhaustion Attack:* Node resources are depleted due to prevention of sleep mode.

4. INTRUSION DETECTION METHODS

A single method of defense against cyber attacks is neither feasible nor possible. It is therefore advisable to compliment first line security mechanisms such as encryption, authentication and authorization with a second line of defense such as intrusion detection. Here an intrusion can be classified as any set of activities that attempt to compromise the integrity, confidentiality or

availability of a resource. An intrusion detection system (IDS) addresses these directly by providing *confidentiality*; ensuring data is not disclosed to unauthorised individuals or systems, *integrity*; ensuring data is preserved in regard to its meaning, completeness and intended use, *availability*; ensuring the data and system are accessible to authorised individuals.
Intrusion detection systems can be classified by two distinct methods, with a third hybrid approach also available which will be discussed below.

Signature-based Detection. Also known as misuse or rule-based intrusion detection relies on known rules (signatures) of previous attacks and attempts to identify possible intrusions by comparing collected data against these predefined rules [19]. Data streams which match these predefined rules would be detected and identified as potential attacks invoking an alert. High true positive and low false positive rates are possible for known attacks, but conversely signature-based detection offers low detection rates for zero day (new) attacks.

Anomaly-based detection. Anomaly-based detection involves comparing current network traffic with a baseline of previously learnt normal network behavior [19]. The baseline is determined by monitoring the network and hosts during an extended period of activity, and building profiles of normal behaviour for each host or protocol. Any significant deviation from this baseline is detected and classified as an anomaly, raising an alert. In contrast to signature-based methods it offers good detection for new or unknown attacks, although it is often considered to produce a high false positive rate.

Specification-based Detection. Is a hybrid of signature and anomaly-based detection where specifications are developed to describe normal network behavior [20]. Two detection mechanisms are usually combined, one to detect known attacks using signatures, the other to monitor traffic and detect deviations from learnt normal network behaviour. Since specifications of normal behaviour are developed manually Low false positive rates can be achieved compared with anomaly-based detection methods.

5. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks (ANNs) are a method of machine learning commonly used for anomaly-based detection [21]. Interconnected neurons exchange information to estimate or predict outputs from supplied patterns of inputs. They are typically organized into layers, each consisting of a specified number of interconnected neurons. Patterns of data are supplied at the input layer, processed at the hidden layer using a system of weighted connections, before supplying an answer at the output layer as shown in Figure 1.

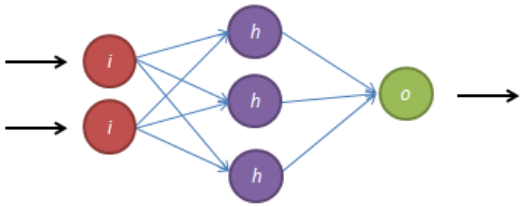

Figure 1. Neuron connections and information flow.

The arrows represent connections between neurons and also indicate the direction or flow of information. Each connection has a weight (integer number) that controls the signal between the two neurons. If the output from the network does not meet the desired output, performance can be improved by iteratively updating the weights until it reaches an acceptable accuracy or until no

further improvement to the learning can be made. Figure 2 shows two ANN architectures which exist *Feed-forward* and *Feedback* networks.

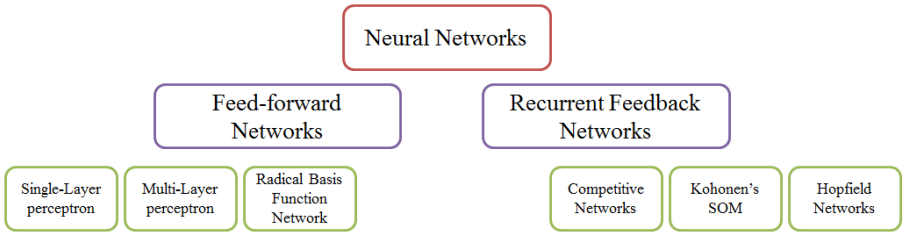

Figure 2. Feed-forward and Feed-back network architectures [22].

In feed-forward networks information flow is unidirectional from input to output as shown in Figure 3. Since feedback loops do not exist the output of any layer does not affect that same layer. Feed-forward networks are commonly used in pattern generation, recognition and classification.

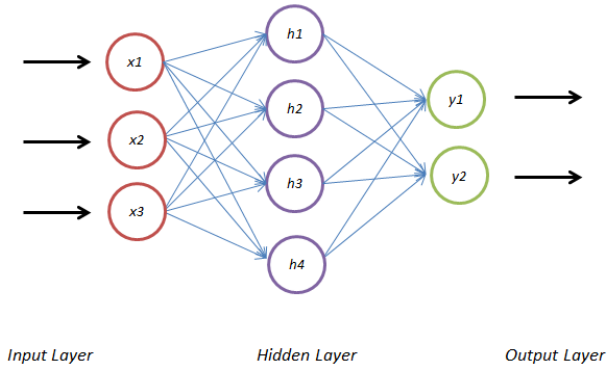

Figure 3. Feed-forward Neural Network Architecture.

Feedback networks differ since feedback paths now exist allowing information to travel in both directions. Inputs to each neuron can now be modified meaning the state of the network is continuously changing and evolving, as shown in Figure 4. Feedback networks are commonly used for image captioning, speech recognition and motion detection.

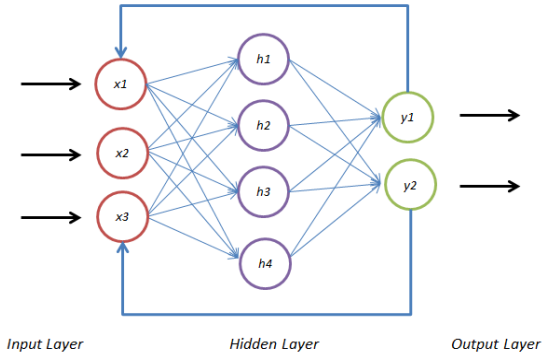

Figure 4. Feedback Neural Network Architecture

Both types of NN architectures share an ability to learn, but do so differently. Feed-forward networks require supervised learning, where the network is provided with the desired answer (output) to each given input and used to train the network to provide accurate future outputs. Feedback network instead utilise unsupervised learning where the desired output is not provided and instead the data is clustered based on relationships among the variables in the data.

6. SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is another a method of supervised machine learning which can be used an alternative approach to anomaly-based detection [10]. The SVM supervised machine learning algorithm is commonly used to solve complex classification problems, but supports both regression and classification tasks and can handle multiple continuous and categorical variables.
In this method of machine learning the algorithm performs classification tasks by constructing hyperplanes in a *n-dimensional* space (where n is number of features you have) that separate cases of different class labels [23]. Classification is then performed by finding the hyper-plane that best differentiates the two classes.

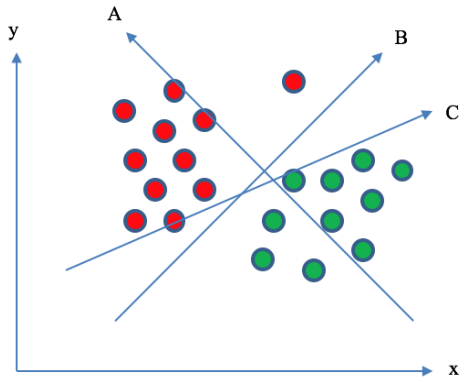

Figure 5. Feedback Neural Network Architecture

In Figure 5, three hyper-planes exist (A, B and C). We see that hyper-plane A misclassifies the two classes, hyper-plane C offers a better accuracy of classification, and hyper-plane B offers the best classification of the two classes.

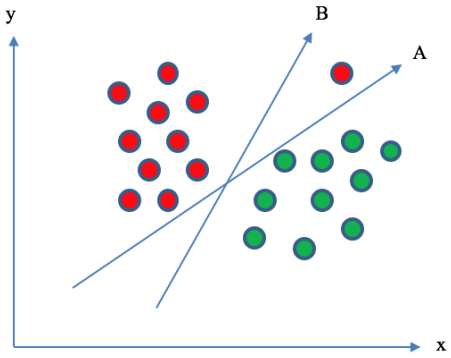

Figure 6. Feedback Neural Network Architecture

In Figure 6, two hyper-planes exist (A and B). When both hyper-planes appear to offer good classification, SVM will consider the distance (*margin)* between the nearest data point (for both classes) and the hyper-plane to decided which plane offers the best classification [23]. SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. The hyper-plane with the highest margin will then be selected. In Figure 6, hyper-plane B has the highest margin, however it has a classification error, therefore hyper-plane A will be selected.

7. IMPLEMENTATION AND CONFIGURATION OF NETWORKS

In this paper anomaly-based detection was chosen as a method of intrusion detection in wireless sensor networks. To test and compare detection rates, a *Multi-Layer Perceptron Backpropagation Neural Network (BPN)* was chosen from the architectures shown in Figure 2 and will be compared against a *Support Vector Machine (SVM)* Classifier.

7.1. DATASET PRE-PROCESSING AND NORMALISATION

In the absence of many credible public datasets with which to study network based anomaly intrusion detection many in the research community revert back to the popular KDDCUP'99 dataset. The dataset however suffers from a number of problems highlighted by McHugh [24] and was therefore discredited for use in this paper. An amended version (NSL-KDD) [25] was suggested to address some of these inherent problems and was subsequently chosen instead.
Before the dataset could be offered to either network, data preprocessing and normalization was required to convert the raw input data into an appropriate format which the machine learning algorithms could use for subsequent analysis. The creators of the NSL-KDD dataset had preprocessed the data in part by filling in missing values and removing redundant or duplicate records, therefore the final preprocessing step was to select the most relevant features. The workflow proposed by Can et al. [12] was followed and 41 features reduced to 22 as follows:
`

(1,2,34,5,6,10,12,14,17,22,23,24,27,29,30,32,33,35,36,37,41)

Three of the chosen features contained strings: Protocol (2) Service (3) and Flag (4) and therefore required to be converted from non-numeric qualitative data to numeric quantitative data to bring all variables into proportion with one another [26].

Finally, normalisation was required to scale all attributes into range [0,1] to achieve unity- based normalisation. This could be achieved with equation [13]:

$$x_{i,0\ to\ 1} = \frac{x_i - x_{min}}{x_{max} - x_{min}}.$$

Using this method of min-max normalisation a value of zero can cause problems for some models since the highest and lowest values could remain the same [0,1] therefore Can et al used an arbitrary range between 0.1 to 0.9 and amended the equation as follows [12]:

$$x_{i,} = 0.8x \frac{x_i - x_{min}}{x_{max} - x_{min}} + 0.1$$

where x_i is each data point, $x_{min}$is the minimum data point value, $x_{max}$is the maximum data point value, and $x_{i,0\ to\ 1}$is the data normalized between 0 and 1. It is also possible to use the following equation to achieve a more centralised set of normalised data, with zero being the central point.

$$x_{i,-1\ to\ 1} = \frac{x_{i-\left(\frac{x_{max}+\ x_{min}}{2}\right)}}{\left(\frac{x_{max}\ -\ x_{min}}{2}\right)}$$

7.2. NETWORK CONFIGURATION AND SIMULATIONS

In this paper, NSL-KDDTrain+_20Percent dataset is selected to train and test the BPN. The MATLAB nprtool was used with the parameters shown in Table 1. The dataset contains four attack types (DOS, R2L, Probe, U2R) but for the purpose of this study, this was reduced to focus on only DOS attacks, of which six attack types were found (Smurf, Neptune, Back, Teardrop, Pod, Land).

The NSL-KDDTrain+_20Percent dataset was also offered to the SVM classifier using MATLAB with the parameters shown in Table 2. The same six attack types used in the previous setup were again used to test intrusion detection rates (Smurf, Neptune, Back, Teardrop, Pod, Land).

Table 1. Feed-forward Backpropagation Neural Network Configuration Parameters.

Feed-forward Back Propagation Neural Network	
Network Simulator for Dataset	MATLab *nprtool*
Implemented Attacks	Smurf, Neptune, Back, Teardrop, Pod, Land
Dataset Samples	25192
Number of Input and Output Layers	22/1
Network Type	Feed-Forward Backpropagation
Training Function	TRAINLM
Adaption Learning Function	LEARNGDM
Performance Function	MSE
Number of Layers	2
Number of Neurons	20
Transfer Function	LOGSIG
Epochs	100
Min_Grad	1e-010

Table 2. Support Vector Machine Configuration Parameters.

Support Vector Machine Classifier	
Network Simulator for Dataset	MATLab *Classification*
Implemented Attacks	Smurf, Neptune, Back, Teardrop, Pod, Land
Dataset Samples	25192
Network Type	Support Vector Machine
Preset	Fine Gaussian SVM
Kernel Function	Gaussian
Kernel Scale	1.2
Box Constraint Level	1
Multiclass Method	Ove-vs-Ove
Standardise Data	True

8. RESULTS AND DISCUSSION

To evaluate the two techniques compared in this study, we collected statistics on the True Positive (TPR) and False Positive rates (FPR) for all included attacks. The TPR can be calculated using $TPR = \frac{\text{TP}}{\text{TP+FN}}$ where True Positive (TP) denotes the number of attacks identified and False Negative (FN) denotes the number of attacks which were not detected.

The FPR can be calculated using $FPR = \frac{\text{FP}}{\text{FP+TN}}$ where False Positive (FP) denotes the number of attacks detected that were actually normal traffic and True Negative (TN) denotes when attacks do not exist and are therefore not detected.

Test results for the BPN are shown in Table 3 and the confusion matrix in Figure 7. It can be clearly seen that the network returned good TPR for most attacks, all of which were above the desired 90% TPR (highlighted green). Neptune attacks for example had a TPR of 99.3%, since 8240 samples were correctly detected, whilst 55 samples were incorrectly classified. Interestingly as the number of attacks samples decreased, it resulted in not only a decrease in the TPR but also an increase in the FPR. This is evident for the back and teardrop attacks which both had relatively small samples and returned FPRs of 7.8% and 2.8% respectively. Attacks with very few samples (pod and land) proved difficult to detect, with this method of anomaly-based intrusion detection. The results obtained were found to be comparable with those found in related research as described in [12] which found similar detection rates and relationship between low sample attacks and detection rates.

Table 3. Back Propagation Neural Network Results.

Back Propagation Neural Network (BPN)							
Attack	**Sample**	**True Positive Rate Detection Rate**			**False Positive Rate**		
		TP	**FN**	**TPR**	**FP**	**TN**	**FPR**
Normal	13449	13374	75	98.8%	74	13374	1.2%
Neptune	8282	8240	42	99.3%	55	42	0.7%
Smurf	529	519	10	100%	0	10	0.0%
Back	196	177	19	92.2%	15	19	7.8%
Teardrop	188	139	49	97.2%	4	49	2.8%
Pod	38	0	38	0.0%	0	38	0.0%
Land	1	0	1	0.0%	0	1	0.0%

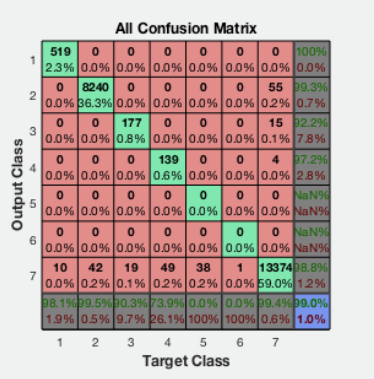

Figure 7. Feedback Neural Network confusion matrix.

Test results for the support vector machine are shown in Table 4 and the confusion matrix in Figure 8. It can be clearly seen that overall the network performed comparatively with the BPN with good TPR for most attacks (highlighted green). Neptune attacks by comparison had a TPR of 99.2%, since 8223 samples were correctly detected, whilst 38 samples were incorrectly classified. Once again, though the data showed that the impact of having fewer attack samples was to decrease the TPR, it was however able to detect very low samples such as pod and land with TPR of 44.7% and 100% respectively. Interestingly it was also noted that overall the FPR was much lower than that of BPN and all remained below the desired 1% for attacks, suggesting less samples were misclassified. The results obtained were found to be comparable with those found in related research as described in [11] which found an SVM classifier to offer the best performance (high accuracy) in comparison to other data mining algorithms, when using the same NSL-KDD dataset.

Table 4. Support Vector Machine results.

Support Vector Machine (SVM) Results							
Attack	**Sample**	**True Positive Rate Detection Rate**			**False Positive Rate**		
		TP	**FN**	**TPR**	**FP**	**TN**	**FPR**
Normal	13449	13376	73	99.4%	104	13376	0.99%
Neptune	8282	8223	59	99.2%	38	45	0.84%
Smurf	529	519	10	98.1%	2	5	0.4%
Back	196	165	31	84.2%	0	31	0%
Teardrop	188	181	7	96.3%	0	7	0%
Pod	38	17	21	44.7%	3	16	0.19%
Land	1	1	0	100%	29	0	0%

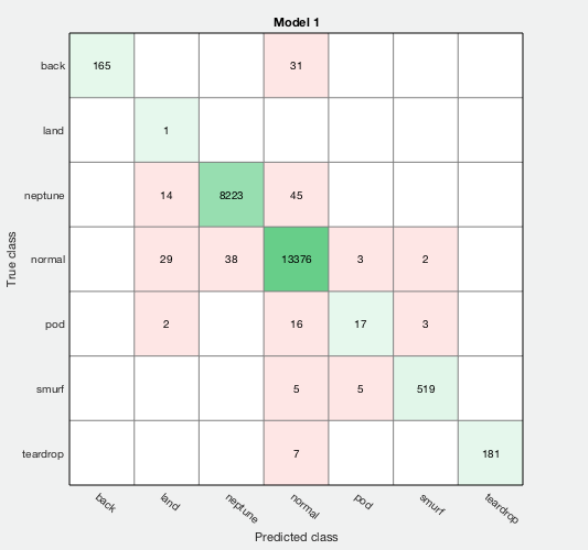

Figure 8. Support Vector Machine confusion matrix.

9. CONCLUSIONS

In this study, we analysed computational intelligence techniques for intrusion detection in WSNs. We reviewed major DoS attacks faced by WSNs and methods of intrusion detection. Finally, we carried out MatLab simulations to observe and evaluate the performance of an artificial neural network and support vector machine in detecting WSN intrusions. Our experimental studies demonstrated the promise of both computational intelligence techniques in effectively detecting intrusions. Both techniques returned good and comparable FPR results, however the support vector machine also further demonstrated its suitability for anomaly detection by handling low sample sizes better, while still maintaining an FPR rate under the 1% threshold. This would suggest for the dataset used (NSL-KDD) that the support vector machine proved to be the better technique for anomaly detection.

The next step in our research will be to investigate threats faced by IP-based WSNs and develop a novel computational intelligence technique capable of detecting and predicting botnet DDOS activity in a physical WSN deployment.

REFERENCES

[1] D. Evans, "The Internet of Things - How the Next Evolution of the Internet is Changing Everything," CISCO white Pap., no. April, pp. 1–11, 2011.

[2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Comput. Networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] J. Radak, B. Ducourthial, V. Cherfaoui, and S. Bonnet, "Ad-hoc Networks and Wireless," Ad-hoc Networks Wirel. 2014 - Int. Work. Wirel. Sensor, Actuator Robot Networks (WiSARN 2014), vol. 8629, pp. 27–34, 2015.

[4] S. Yinbiao and K. Lee, "Internet of Things : Wireless Sensor Networks Executive summary," 2014.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Commun. Mag., vol. 40, no. 8, pp. 102–105, 2002.

[6] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: a survey," IEEE Commun. Surv. Tutorials, vol. 11, no. 2, pp. 52–73, 2009.

[7] A.-S. K. Pathan, H.-W. Lee, and C. S. Hong, "Security in wireless sensor networks: issues and challenges," 2006 8th Int. Conf. Adv. Commun. Technol., vol. 2, p. 6 pp.-pp.1048, 2006.

[8] P. Yi, Y. Jiang, Y. Zhong, and S. Zhang, "Distributed Intrusion Detection for Mobile Ad Hoc Networks," 2005 Symp. Appl. Internet Work. (SAINT 2005 Work., pp. 94–97, 2005.

[9] H. Sedjelmaci and M. Feham, "Novel Hybrid Intrusion Detection System for Clustered Wireless Sensor Network," Int. J. Netw. Secur. Its Appl. (IJNSA), Vol.3, No.4, July 2011, vol. 3, no. 4, pp. 1–14, 2011.

[10] L. Khan, M. Awad, and B. Thuraisingham, "A new intrusion detection system using support vector machines and hierarchical clustering," VLDB J., vol. 16, no. 4, pp. 507–521, 2007.

[11] S. K. Sahu, S. Sarangi, and S. K. Jena, "A detail analysis on intrusion detection datasets," Souvenir 2014 IEEE Int. Adv. Comput. Conf. IACC 2014, pp. 1348–1353, 2014.

[12] O. Can, C. Turguner, and O. K. Sahingoz, "A Neural Network Based Intrusion Detection System For Wireless Sensor Networks," Signal Process. Commun. Appl. Conf. (SIU), 2015 23th, pp. 2302–2305, 2015.

[13] F. Lu and L. Wang, "Intrusion Detection System Based on Integration of Neural Network for Wireless Sensor Network," J. Softw. Eng. 2014.

[14] Y. Y. Li and L. E. Parker, "Intruder detection using a wireless sensor network with an intelligent mobile robot response," Southeastcon, 2008. IEEE, pp. 37–42, 2008.

[15] A. Kulakov and D. Davcev, "Tracking of unusual events in wireless sensor networks based on artificial neural-networks algorithms," Inf. Technol. Coding Comput. 2005. ITCC 2005. Int. Conf., pp. 534–539, 2005.

[16] M. Panda, "Security Threats at Each Layer of Wireless Sensor Networks," Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 3, no. 11, pp. 61–67, 2013.

[17] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," Proc. First IEEE Int. Work. Sens. Netw. Protoc. Appl. 2003., pp. 113–127, 2003.

[18] S. H. . Harris, G. M. W. Al-Saadoon, R. . Ahmad, and M. A. H. . Ghani, "Anomaly Detection of IP Header Threats," Int. J. Comput. Sci. Secur., vol. 4, no. 6, pp. 497–504, 2011.
[19] N. A. Alrajeh, S. Khan, and B. Shams, "A Survey of Intrusion Detection Systems in Wireless Sensor Networks," IEEE Commun. Surv. Tutorials, vol. Early Acce, 2013.
[20] I. Butun, S. D. Morgera, and R. Sankar, "A Survey of Intrusion Detection Systems in Wireless Sensor Networks," IEEE Commun. Surv. Tutorials, vol. 16, no. 1, pp. 266–282, 2014.
[21] A. Petrovski, P. Rattadilok, and S. Petrovski, "Designing a Context-aware Cyber Physical System for Detecting Security Threats in Motor Vehicles," Proc. 8th Int. Conf. Secur. Inf. Networks, pp. 267–270, 2015.
[22] M. T. Taro Ishitaki, Tetsuya Oda, Keita Matsuo, Leonard Barolli, "Performance Evaluation of a Neural Network Based Intrusion Detection System for Tor Networks Considering different Hidden Units," Network-Based Inf. Syst. (NBiS), 2015 18th Int. Conf., pp. 620–627, 2015.
[23] D. S. Kim and J. S. Park, "Network-Based Intrusion Detection with Support Vector Machines," Inf. Netw. Int. Conf. ICOIN 2003, Cheju Island, Korea, Febr. 12-14, 2003.Revised Sel. Pap., pp. 747–756, 2003.
[24] J. McHugh, "Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," ACM Trans. Inf. Syst. Secur., vol. 3, no. 4, pp. 262–294, 2000.
[25] University of North Brunswick, "NSL-KDD Dataset," 2016. [Online]. Available: https://web.archive.org/web/20150205070216/http://nsl.cs.unb.ca/NSL-KDD/. [Accessed: 03-Mar-2016].
[26] B. Etzkorn, "Data Normalization and Standardization," 2011. [Online]. Available: http://www.benetzkorn.com/2011/11/data-normalization-and-standardization/. [Accessed: 31-Mar-2016].

AUTHORS

Christopher D. McDermott is as a Lecturer in the School of Computing Science and Digital Media at Robert Gordon University, Scotland. He completed his BSc (Hons) Business Information Technology at The University of Northumbria, England. Following a successful period in industry he is now currently pursuing his PhD investigating security threats facing the Internet of Things (IoT). He is a member of the Security and Privacy and Machine Learning research groups and his current research interests include, but are not limited to Network Security, Digital Forensic Science, Steganography, Information Security, Internet Security.

Andrei Petrovski is a Reader in Computational Systems in the School of Computing Science and Digital Media at Robert Gordon University, Scotland. He completed his MSc in Electrical Engineering at Samara State University in Russia, and his PhD in computing at Robert Gordon University. He is a member of the Security and Privacy and Machine Learning research groups and his research expertise and interests include computational modelling, optimisation and decision support, practical applications of computer-assisted measurements, fault diagnosis and predictive control. He has successfully applied this expertise for improving accuracy and versatility of instrumentation systems, as well as for adding intelligent features to such systems.