A multi-layer perceptron approach for flow-based anomaly detection

Van Efferen, Lennart; Ali-Eldin, Amr M.T.

DOI
10.1109/ISNCC.2017.8072036

Publication date
2017

Document Version
Final published version

Published in
Proceedings of International Symposium on Networks, Computers and Communications, ISNCC 2017

Citation (APA)
Van Efferen, L., & Ali-Eldin, A. M. T. (2017). A multi-layer perceptron approach for flow-based anomaly detection. In *Proceedings of International Symposium on Networks, Computers and Communications, ISNCC 2017* Article 8072036 IEEE. https://doi.org/10.1109/ISNCC.2017.8072036

A MULTI-LAYER PERCEPTRON APPROACH FOR FLOW-BASED ANOMALY DETECTION

Lennart Van Efferen [1,] and Amr M.T. Ali-Eldin [1,2,3]

[1] Leiden Institute of Advanced Computer Science, Leiden University, Leiden, the Netherlands.

[2] Faculty of Technology, Policy and Management, Delft University of Technology, Delft, the Netherlands.

[3] Computer and Control Systems Dept., Faculty of Engineering, Mansoura University, Mansoura, Egypt.

***Abstract*— *The increase in successful cyber-attacks on systems with firewalls and encryption techniques has led to the creation of Intrusion Detection Systems (IDS). Machine learning techniques are often used for these systems to predict malicious behaviour in the vague and unbalanced data. Flow-based IDS monitors only the packet headers of the network traffic and not the attached data to keep up with the growing bandwidth of networks and to maintain the privacy of the users. In this context, a multilayer perceptron approach is analysed on two different datasets and compared to a J48 Decision Tree classifier. Obtained results confirm that flow-based systems seem to be, apart from inevitable, the right way for IDS in the future and that MLP can still be useful in flow-based detection.*

Keywords— Intrusion detection systems (IDS); anomaly detection; Artificial Neural Networks (ANNs); Multi-layer Perceptrons (MLP); J48 decision tree.

I. INTRODUCTION

Cyber-attacks represent a massive threat to the safety and privacy of society. Traditional techniques usually fail to protect systems against cyber-attacks. Intrusion detection systems (IDS) may be useful in detecting and denying intrusions before it occur based on historical (input) data. One of the challenges IDS systems face is the massive growing in size of input data to be scanned in addition to the increase in malicious behaviour.

Monitoring the flows of the network traffic can help reduce the complexity of IDS compared packet payload where the whole packet including the associated data is monitored. By looking at flows instead of the payloads, the data size for investigation drops drastically. Some attacks however can be overlooked by flow-based systems since they reside in the data headers of the packet. In order to reduce these risks, data mining techniques are often used in IDS to predict malicious behaviour through detecting anomalies in network traffic flow. One of the most common approaches are multi-layer perceptron networks or simply, artificial neural networks (ANNs). Machine learning techniques functioning as classification algorithms can work in two different ways: a misuse detection system (MDS) which tries to classify attacks by learning the characteristics of these penetration attempts and anomaly detection system (ADS) which tries to identify abnormality based on a normal model, instead of classifying each attack (binary classification).

The objective of this paper is to analyse the performance of flow-based IDS using MLP compared to a J48 trees approach. This paper is organised as follows: next section introduces related work. Then an overview of the multi-layer perceptron approach is presented in section III followed by a discussion on the datasets used in this work in section IV. Afterwards, the results of the empirical work are shown in section V followed by a discussion in section VI. Finally, the paper is concluded in section VII.

II. RELATED WORK

In the last years, a lot of research has been conducted in the field of cyber security, with special focus on computational algorithms used for both anomaly and misuse detection. A representative dataset that contains a wide variety of instances that can arise in the outside world is still a challenge for IDS research. The DARPA and KDD datasets are still the most publicly available sets, even after the criticism by Mc.Hugh [1], Malhony and Chan [2] and Sabnani et al [3]. Nowadays a lot of researchers generate their own datasets to avoid these unfavourable sets or to overcome their limitations (of incomplete training sets for example). Examples of such datasets are honeypot [4] and winter [5].

Artificial Neural Networks (ANNs) are one of the most successfully employed data processing algorithms. They have the ability to generalise models from incomplete and noisy data. Here, network data is used as the input for the detection of anomalies. Cannady [6] showed good results with an MLP misuse system, even though the time to build the model was relatively high. Other researchers used ANN for malicious detection in different input data, such as Tan [7] in user behaviour data or Gosh [8] in sequences of system calls.

For the classification of network traffic, payload and flow based systems have been investigated thoroughly. Gogoi et al. [9] and Alaidaros et al. [10] gave a great overview on how the performance and accuracy are compared and where the advantages and disadvantages of these systems lie. Wang et al. [11] developed a payload based anomaly detector called PAYL with almost 100% accuracy on traffic over port 80. In [12], three different techniques were compared and evaluated; Bayesian networks, decision trees and MLP. The decision trees performed superior compared to the Bayesian networks in classification accuracy, but required a longer training time.

978-1-5090-4260-9/17/$31.00 ©2017 IEEE

MLP showed less accurate values compared to the decision trees and had a longer training time than the Bayesian networks. Jadidi et al. [13] used a neural network optimized with a Gravitational Search Algorithm (GSA) on Winters dataset for a flow-based system. The system resulted in 99.43% accuracy on classifying benign and malicious flows. Abuadlla et al. [14] used three different training algorithms for a two stage neural network as a flow-based system; Resilient Backpropagation, Radial Basis Function Net and Levenberg Marquardt. The first stage was the anomaly detection stage and the second stage was the detection and classification stage. The analysis firstly showed an improvement of prediction accuracy in the second stage compared to the first (anomaly detection) stage. Secondly, a multilayer perceptron with Levenberg Marquardt had low memory consumption and a low false alarm rate compared to the Radial Basis Function and was faster compared to Backpropagation.

The data IDSs retrieve as input must be relevant for the detection of cyber-attacks, therefore it could be network traffic (data packets), command sequences from user inputs, or low level system information (e.g. log files or CPU usage) of the system or network. However, the volumes for this sort of data can be huge, especially when the inspected network is of some size. Furthermore, the data distribution is highly imbalanced and there is not a realizable boundary between normal and abnormal behaviour. To make it even more difficult, people come up with new penetration techniques every day so there is a need for continuous adaption for this changing environment.

Most IDS systems use MDS approach because it is effective when the intrusion signature is created in a good way. Moreover, since network traffic tends to be vague and rarely is normal, ADS that attempts to build a model for this normal behaviour frequently fails [15]. The false alarm rate is therefore high compared to MDS. However, MDS only detect attacks when prior knowledge of the attack is available. For responsive behaviour in a detection system without prior knowledge, an ADS is preferable as MDS lack of self-learning abilities and therefore fails to defend against new attack types if the signatures are not updated [16].

Machine learning can be either supervised or unsupervised learning. Unsupervised learning creates a model without the need for a labelled dataset by modelling the universal properties of the data. Supervised learning trains a classifier on a labelled dataset in order to determine to which class instances it fits. The problem with supervised learning in IDS is the lack of available labelled datasets. Two commonly used data sets consisting of network traffic and audit logs are available online and are known as DARPA and KD99. These sets have been used extensively by researchers to learn and develop IDS. Yet, as numerous research showed, these data sets have problems that need to be overcome in order to create a more useful detection system [1-3] .

There are different machine learning techniques for these cybersecurity systems, such as artificial neural networks, evolutionary computation, artificial immune systems, fuzzy systems, swarm intelligence or soft computing. These techniques are used because they can adapt to a changing environment, they are resilient against noisy information and can exhibit fault tolerance. Artificial neural networks (ANNs) seem promising for the classification of network traffic [6]. They can generalize from limited noisy and incomplete data, which are often characteristics of network data. Furthermore, if the network structure is well developed, high computational speed can be achieved and more attacks can be prevented [17, 18]. Normal multilayer feed forward (MLFF) require long training time, and hence Radial Basis Function (RBF) neural networks are used instead [19]. Another way to improve the detection rate is good feature selection, which will be done for this research. Additionally a comparison is conducted with a J48 decision tree mechanism due to its efficiency in the classification of network traffic [20].

III. THE MULTI-LAYER PERCEPTRON APPROACH

Artificial neural networks emerged from the way biological neurons in the brain work. A biological neuron will fire a potential action if the cumulative input of the signals arriving exceeds a certain threshold, represented as θ. This threshold however varies around an average value and is not the same for every neuron so that it is uncertain if a neuron is doing what is expected. The firing thresholds are being updated continuously, which is the key factor for the adaptive learning abilities of the network. The same is true for a single layer artificial perceptron which can only solve linear problems:

$$y = \sum_{i=1}^{n} W_i \times X_i \qquad (1)$$

Where n represents the number of inputs corresponding to the features. The weights per input are denoted to Wi and Xi is the input data. The perceptron then translates the inputs to an output signal, with respect to the threshold, using a transfer function. For example the output will be 1 (firing state) when $\sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{W_i} \times \mathrm{X_i} \quad > \theta$. Commonly used examples of these functions are Unit Step, Sigmoid or Gaussian. The weights determine the slope of the transfer function and the Bias allows shifting the transfer function horizontally along the axis while leaving the curvature unaltered. Like the biological neurons, learning arises in updating the weights and Bias in order to reduce the error rate. The perceptron weight adjustment is denoted by:

$$\Delta W = \mu \times \delta \times X \qquad (2)$$

With $\mu < 1$ as the learning rate and $\delta = (\text{predicted output} - \text{ desired output})$.

A MLP has the same structure of a single layer with the addition of one or more hidden layers with all the nodes connecting each other between layers (see Fig. 1). The network trains itself with an algorithm called backpropagation. This supervised learning algorithm first computes outputs using a sigmoid function and then propagates the errors backwards. Each unit receives the amount of error it generated this way and the weights are adjusted. In short, backpropagation uses the output error to adjust the weights of inputs at the output error and then continues this adjustment for the previous layers. The error in one of the output nodes is then denoted as:

$$\delta_0 = output \times (1 - output) \times (expected - outpt) \quad (3)$$

And the error rate for a node h in the hidden layer can be calculated as:

$$\delta_h = output \times (1 - output_h) \times (W_h - \delta_h) \quad (4)$$

The system has to predict if an attack is happening from all the inputs it receives. The problem can therefore be expressed as in (1). If the inputs exceed a certain threshold, then an incoming attack is likely. The weight can be adjusted by the backpropagation algorithm. For the activation function in the nodes, the sigmoid function will be implemented:

$$\sigma(t) = \frac{1}{1+e^{-x}} \quad (5)$$

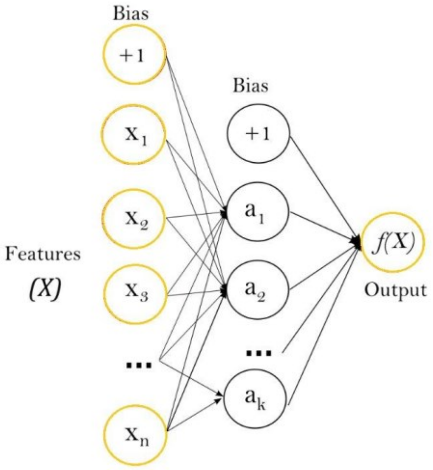

Fig. 1 Multilayer perceptron with one hidden layer

There are three different ways a network based intrusion detection system can classify network traffic data. The first method is port-based, where the classification is based on the 16 bit transport layer port numbers used by servers for traffic flow. It unwraps the predefined layers of the packet and inspects them with a protocol analysis method. Anything that is deviating from the standard use of the protocol is likely malicious. There are some problems with this technique, such as the fact that the FTP protocol can assign ports dynamically according to the traffic load. Additionally, many protocols used by applications today are not registered to IANA (Internet Assigned Numbers Authority) or they use a varying set of ports in order to remain anonymous (like P2P applications). Due to these issues, the results from this port-based technique suffer from a low degree of accuracy [21]. The second technique for network traffic classification is the inspection of packet payload data [11]. This technique yields satisfactory accuracy but it demands a thorough investigation of the data. Privacy cannot be maintained unless the data is encrypted, which renders inspection of the packet payload data ineffective. The third technique is called flow-based classification [22], where the important features of network flows are collected from the packet headers (the transport ports can be features as well). Each traffic flow is characterized by a set of determined features with values that depend on the network class. Other features can then be created with these sequences of packets, like the connection duration or the amount of packets retrieved from one source. Since this data is gathered from the packet headers, privacy is maintained, data encryption is possible and less data is needed to be examined in comparison with the packet payload data technique or the protocol technique (since flow-based only investigates an aggregated set of packets instead of every packet) [23]. It however got some drawbacks regarding certain attack types. Some malware, for example, worms or viruses, deliver their malicious code in the payload data of the packets and will therefore be hard to detect by just investigating the flow between source and destination.

Flow data normally derives from modules placed in network routers. The packets consists of 20 bytes of data and thirteen items, some examples are: IP Header Length (number of 32 -bit words forming the header), Size of Datagram (The combined length of the header and the data in bytes), Identification (uniquely identifies this packet together with the source address), Time To Live (TTL) (Number of hops which the packet may be routed over), Protocol (type of transport packet being carried (e.g. 1 = ICMP; 6 = TCP; 17= UDP), Source Address (the IP address of the source), Destination Address (the IP address of the destination) [24]. The last field of the packet is the data that is sent from the host to the source. With all the TCP or UDP headers, this data field starts with a TCP header. This header gives additional information to the application that will receive the incoming data. Some examples of the fields are: Source port (the port number of the application that sends the data), Destination port (the port number of the application that receives the data), Control flags (bit string that indicates which of the six control flags are on and off), Checksum (the checksum of the remaining data that needs to be received), Data (the actual data that is being sent from the host to the receiving entity) [24].

IV. Datasets

Finding the right dataset is one of the most difficult tasks for IDS. Most of the datasets available are outdated and contain old attacks. Moreover, almost all available sets contain unrealistic data. For example, the DARPA 98 and 99 are simulated in a military network environment and are some of the most commonly used datasets of network traffic; however they have three major issues. Firstly, the sets do not contain modern attack types. Secondly, Mahoney and Chan [2] discovered that all of the packages in the 99 set, that contain a time to live (TTL) of 126 or 253 are malicious data packages. This is undesirable for data mining algorithms as a predictor, since they would learn in an incorrect way. Thirdly, the TTL values are artificially high in comparison to real traffic data. For these reasons, the data is not representative and the DARPA 98 and 99 are no longer recommended for research. Another popular dataset is the KDD CUP 1999 set. However, it is essentially an extension of the DARPA 98 version (adding more features) and thus suffers from the same constraining characteristics. Moreover, the set is missing records and the training set contains redundant data. This is the reason why the NSLKDD dataset, an improvement of the KD99 set, was issued. However NLSKDD is still not a good representation of a real life environment [1].

A. Winter

The first publicly available flow-based dataset was captured by monitoring a honeypot, an environment that attracts attackers and analyses the network data [4]. The data contains ten features that provide flow information of the collected traffic. The dataset consists of malicious traffic, side effect traffic (not malicious) and unknown traffic (traffic that could not be classified). Winter [5] modified this dataset in some ways that could be beneficial for training algorithms. The dataset is reduced in size and some features were dropped (IP addresses since they have been anonymized) or combined for the training time. The old dataset was time consuming since it consists of 14.2 million flows (more than 98 % has been labelled). By selecting only the relevant flow attributes, deleting the unlabelled flows and all flows belonging to other protocols than SSH and HTTP and reducing the size, this dataset becomes easy to use and not time consuming. The new dataset consists of 22942 flows, gathered through a random sampling process with a probability chance of 1/600.

B. UNSWNB15

The UNSWNB15 dataset [25] was created in the Australian Centre for Cyber Security to simulate modern attack activities. The UNSWNB15 contains nine types of attacks and 49 generated features. Both anomaly detection and attack classification are conducted with the neural network on this dataset. The features and their descriptions can be divided into several different categories: basic features, flow features, content features, time features and additional generated features. The flow features are the results, saved as Packet capture (PCAP) files, from the TCP dump and are the contents of the network packets. They consist of the normal features that are necessary for a connection, such as IP-addresses, the length of the packet and the protocol used. All of the other features are generated by the tools BRO-IDS and Argus. The basic features can be derived from the packet headers without inspecting the payload of the packets. The content features are of flow statistics, information from the TCP/IP headers or payload information. Time features describe the data that supply a specific time or need to mature over a temporal window. Some examples of such features are the recorded start time, the recorded last time or the time between the SYN and the SYN_ACK packets of the TCP. The remaining features are depicted in the dataset. The labelled features 48 and 49: attack category and label, are the target features for data mining algorithms. The former refers to the name of each attack category among the nine categories: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. The latter feature specifies whether there is an attack or not, thus taking the binary form. Out of this dataset, most of the features can be gathered from the network flow or generated additionally from these features.

C. IDS Performance indicators

The performance of an IDS in detecting attacks can be defined as follows:

- True Positive (TP): how often an IDS correctly reports a predefined attack in the dataset.
- False Negative (FN): how often an IDS fails to identify an attack.
- True Negative (TN): how often an IDS does not find normal packets as predefined in the dataset.
- False Positive (FP): how often an IDS incorrectly reports an attack.

Precision of IDS can be seen as the fraction of the correctly reported attacks and is defined as the number of correctly reported attacks (True Positives) divided by the total number of attacks found by the IDS (True Positives and False Positives). Recall represents the fraction of real found attacks and is defined as the number of real attacks (True Positives) divided by the number of all the attacks that exist (True positives and false negatives). The F-score describes an IDS performance as follows:

$$F - score = \frac{Precision*Recall}{Precision+Recall} \tag{6}$$

V. Experimental Work

In this section, the experimental work is presented. Building the MLP was done using Weka [26]. In addition, MLP was compared to another machine learning approach called J48 decision tree. The J48 tree is the open source implementation of the C4.5 decision tree algorithm, which itself is an extension of the earlier ID3 algorithm. The algorithm utilises a treelike structure with a root, nodes and leaves for the decisions in the classification problems. Like the MLP, it is a supervised classification algorithm, since it needs all the outcomes of possible attributes to build itself. The outcome as a tree structure is easy to understand for the end users, unlike the blackbox model of MLP whereas the analysis of the data responses occurring in the network does not give any insight of the structure of the function being approximated. Additionally, the decision tree can handle various inputs or missing data as well and is therefore useful in a wide range of applications. The main experiment in this research was the execution of a flow based anomaly detection (binary classifier) IDS using MLP compared to a J48 tree classifier on both datasets of Winter and UNSWNB15.

The results of these flow-based systems are shown in Table 1 and Table 2. Results derived from Winter's dataset are better than those obtained by UNSWNB15. Moreover, with only 7 features in Winter's dataset to predict the attacks, the time complexity becomes very low compared to 28 features in UNSWNB15. The MLP performed inferior on the UNSWBNB15 dataset with a 93.3 % of the instances correctly classified and a 20.96 % false alarm rate. Even with the 28 features presented, the MLP could not perform better in separating the malicious behaviour from the benign. The contribution of the second dataset shows the importance of right feature selection and how results can differ given the input. The J48 tree performed almost perfectly on Winter's dataset with 0 successful attacks and only 2 false alarms. The results generated from the UNSWNB15 dataset differ from the

MLP in the severity of the system. The MLP only let 26 attacks through but showed a 20.96 false alarm rate, which renders the system as severe. The decision tree did not detect 18582 attacks but only has a 2.72 % false alarm rate.

Table 1 the flow-based anomaly detection system on different datasets using MLP and J48 tree

Data set	Algorithm Type	Total instances	Normal instances	Attacks	Detection rate	Avg. TPR	Avg. FPR	Precision	Recall	F-score
Winter	MLP	8630	942	7688	99.59%	99.6 %	1.8%	0.998	0.998	0.998
	J48 tree	8630	942	7688	99.98%	100 %	0.2%	1.00	1.00	1.00
UNSW-NB15	MLP	175341	56000	119341	93.29%	93.3%	14.3%	0.910	1.00	0.953
	J48 tree	175341	56000	119341	88.53%	88.5%	6.8%	0.985	0.844	0.909

Table 2 Confusion Matrices on different datasets using MLP and J48 tree

Algorithm Type	Winter			UNSW-NB15		
	Classified as normal	Classified as attack		Classified as normal	Classified as attack	
MLP	923	19	Normal	44265	11735	Normal
	16	7672	Attack	26	119315	Attack
J48 tree	940	2	Normal	54475	1525	Normal
	0	7688	Attack	18582	100759	Attack

VI. DISCUSSION

The most reasonable architecture for an MLP as an IDS is well investigated in the literature. Different number of hidden layers for example could generate different outcomes, but the results do not increase with the addition of more layers. Some MLP with just one layer of hidden nodes could detect attacks in an acceptable rate. In this paper, different neural networks with different number of hidden layers were tested. It was noticed that the increase of number of hidden layers did improve the results, but too little for a satisfactory result. A good example is the MLP with 50 hidden layers instead, had an overall higher correctly classified instance rate with 0.12 percent, but the time complexity to build this model became three times higher. This research is searching for a fast adaptable model that could be implemented in a real world environment. The default learning rate of 0.3 is chosen in the Weka Tool. The system should not take too long time since this is one of the strongest points compared to a payload based system. If the learning rate is too high, the system is not capable anymore to handle high bandwidths in networks. Moreover, a MLP with a learning rate of 0.4 scored 0.26 percent lower than the proposed system. However we highly recommend future research to come up with a better architecture to improve the results and maintain the low complexity.

A function could be written to calculate the maximum allowed time complexity for systems these days, and the MLP with the best suitable architecture could be generated for this or the learning rate could be increased if allowed. Also the functions could be tested more, a MLP with a radial basis function for example could result in more satisfactory system, more fit for the outside world.

Further experiments were conducted using the MLP and J48 approaches on both datasets using payload packets. It was noticed that MLP flow-based overlooked 26 attacks while the payload-based overlooked 27 attacks. This concludes little difference in performance between the two network classification techniques on that particular dataset. This finding confirms other researches [9, 10, 27] which recommend flow-based systems since they can handle the growth in internet bandwidth and still can achieve high detection rates. Although the results from Winter’s dataset are promising, Winter’s dataset consists of a smaller number of network flows and therefore may not be representative completely to the real world. Although the J48 decision tree could perform better as a classification technique for network traffic than the MLP [12, 20], we think that the high complexity associated with implementing J48 compared to MLP can still argue for the use of MLP approaches in anomaly detection.

VII. Conclusions

In this paper, a flow-based multilayer perceptron approach for anomaly detection was applied on two different datasets against a J48 decision trees classifier. Multilayer perceptron was selected because of its ability to adapt to changes, resilience to noisy information, and fault tolerance. A J48 decision tree was chosen for comparison due to its proven high precision. Flow-based systems can accommodate with the growing network traffic size and hence protect user privacy. Results of the experimental work show that a flow-based IDS performs very close to a payload based IDS. Also, seen from the results derived from Winter's dataset, with the right feature selection, high precision and recall values can be achieved. Our results confirm that flow-based systems seem to be, apart from inevitable, the right way for IDS in the future and that MLP can still be useful in flow-based detection reducing the complexity of J48 decision trees. Further studies will be conducted using a neuro-fuzzy approach since adding a fuzzy component can help reduce the time needed by the learning algorithm.

References

[1] McHugh, J., 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., 2000. 3(4): p. 262-294.

[2] Mahoney, M.V. and P.K. Chan, An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection, in Recent Advances in Intrusion Detection. 2003. p. 220-237.

[3] Sabhnani, M. and G. Serpen, Why machine learning algorithms fail in misuse detection on KDD intrusion detection data set. Intelligent Data Analysis, 2004. 8: p. 403-415.

[4] Sperotto, A., et al., A Labeled Data Set for Flow-Based Intrusion Detection, in Proceedings of the 9th IEEE International Workshop on IP Operations and Management. 2009, Springer-Verlag: Venice, Italy. p. 39-50.

[5] Winter, P., E. Hermann, and M. Zeilinger. Inductive Intrusion Detection in Flow-Based Network Data using One-Class Support Vector Machines. in 4th IFIP International Conference on New Technologies, Mobility and Security. 2011.

[6] J.Cannady, Artificial Neural Networks for Misuse Detection, National. National Information System Security Conference, 1998: p. 443-456.

[7] K.Tan. The application of neural networks to unix computer security. in Proceedings of IEEE International Conference on Neural Networks 1995.

[8] A.K.Gosh and A.Schwarzbard. A study in using neural networks for anomaly and misuse detection. in Proceedings of the 8th USENIX Security Symposium. 1999.

[9] Gogoi, P., et al. Packet and Flow Based Network Intrusion Dataset. in 5th International Conference on Contemporary Computing. 2012.

[10] Alaidaros, H., M.Mahmuddin, and A.A. Mazari. An overview of flowbased and packetbased intrusion detection performance in high speed networks. in The International Arab Conference on Information Technology (ACIT'2011). Riyadh.

[11] Wang, K. and S.J. Stolfo. Anomalous Payload-Based Network Intrusion Detection. 2004. Lecture Notes in Computer Science.

[12] Soysal, M. and E.G. Schmidt, Machine learning algorithms for accurate flowbased network traffic classification: Evaluation and comparison. Performance Evaluation, 2010. 67: p. 451-467.

[13] Jadidi, Z., et al. FlowBased Anomaly Detection Using Neural Network Optimized with GSA Algorithm. in 33rd IEEE International Conference on Distributed Computing Systems Workshops 2013.

[14] Moustafa, N. and J. Slay. UNSWNB 15: a comprehensive data set for network intrusion detection systems (UNSWNB15 network data set). in Military Communications and Information Systems Conference (MilCIS). 2015.

[15] Biermann, E., E.Cloete, and L.M.Venter, A comparison of Intrusion Detection systems. Computers & Security, 2001. 20 p. 676-683

[16] Ferreira, V.O., et al., A model for anomaly classification in intrusion detection systems,. Journal of Physics: Conference Series, 2015. 633.

[17] Ju, J., Z. Chunlin, and M. Kamel. RBF-based real-time hierarchical intrusion detection systems. in Proceedings of the International Joint Conference on Neural Networks, 2003. 2003.

[18] Wei, X., H. Huang, and S. Tian. A Modified RBF Neural Network for Network Anomaly Detection. in Third International Symposium on Neural Networks. 2006. Chengdu, China: LNCS

[19] Chan, A.P.F., et al. Comparison of different fusion approaches for network intrusion detection using ensemble of RBFNN. in International Conference on Machine Learning and Cyernetics 2005.

[20] Bouzida, Y. and F. Cuppens. Neural networks vs. decision trees for intrusion detection. in IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM). 2006.

[21] Moore, A.W. and K. Papagiannaki. Toward the Accurate Identification of Network Applications. in 6th International Workshop on Passive and Active Network Measurement (PAM 2005). 2005. Boston, MA, USA.

[22] Wu, S.X. and W. Banzhaf, The use of computational intelligence in intrusion detection systems: A review. Applied Soft Computing, 2010. 10: p. 135.

[23] Golling, M., R. Koch, and R. Hofstede. Towards Multilayered Intrusion Detection in HighSpeed Networks. in 6th International Conference on Cyber Conflict. 2014.

[24] Moustafa, N. and J. Slay. The significant features of the UNSW-NB15 and the KDD99 Data sets for Network Intrusion Detection Systems. in 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security. 2015.

[25] UNSWNB15. https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-NB15-Datasets/ - Last visited on 18/12/2016.

[26] Weka. http://www.cs.waikato.ac.nz/ml/weka/downloading.html - Last visited on 18/12/2016.

[27] Sperotto, A. and A. Pras, FlowBased Intrusion Detection. 12th IFIP/IEEE International Symposium on Integrated Network Management, 2011: p. 23-27.