Backdoor Detection Using Machine Learning

Mohammed A. El Zarouq Eshkanti, S.C. Ng

School of Information Technology, SEGi University
No.9, Jalan Teknologi, Taman Sains Selangor Kota Damansara, PJU 5, 47810, Petaling Jaya,
Selangor, Malaysia

moh.esh1990@yahoo.com, ashleyng@segi.edu.my

Abstract

This research aims to design a backdoor detection technique by using machine learning approach. The implication of this research is to ensure a more effective backdoor detection in network platforms. Security measures against these problems include the use of software programs such as backdoor detection programs. One technique is machine learning (ML); that is a set of tools by which a machine can learn new concepts and new patterns based on a history of learned patterns. The work concentrated on application backdoors which are embedded within the code of a legitimate application. The proposed program in this research is an improvement to backdoor detection based on machine learning techniques. It is developed in Java and employs both supervised and unsupervised methods in the WEKA tool. This helps improving the detection compared to previous methods such fuzzy logic. The results of the experiments have proven that the proposed program is better at detecting backdoors than fuzzy logic when valuated with similar data set. This proves that combining K-Nearest Neighbour and Naive Bayes algorithms is better than using Fuzzy Logic method. The program encountered less false positives and detected all backdoors in the dataset.

Introduction

Computer security is concerned with four main areas; confidentiality, integrity, availability, and authentication (Corona, Giacinto, Mazzariello, Roli, & Sansone, 2009; Kraemer, Carayon, & Clem, 2009).Problems that affect computer security include viruses, worms, Trojan horses, and other types of malware (Dube et al., 2012;

Mamalakis, Diou, Symeonidis, & Georgiadis, 2014; B. K. Mishra & Pandey, 2010). Security measures against these problems can include the use of software programs such as anti-virus, anti-malware, firewalls, or user dependent measures such as using strong passwords, activating or deactivating certain computer features like JavaScript or ActiveX (Ahn, Oh, & Park, 2015; Bayoğlu & Soukpinar, 2012).

One security technique is machine learning (ML); that is a set of tools by which a machine can learn new concepts. According to (Simon, 2013) ML is the subject of continuous research in computer science as it extends its application to almost every field of activity, whether private, economical or industrial. One aspect that connects all these activities is the network technology. Thus, there is a growing need to protect valuable information from any cyber-attack. Thus, developing new intelligent methods to detect attacks such as backdoors is needed. Intrusion detection is the process of dynamically monitoring events occurring in a computer system or networks, analyzing them for signs of possible incidents and often interdicting the unauthorized access (Peter, 2007). The traditional way of protecting computer networks, such as firewalls, access control mechanisms, and encryptions are insufficient and have several limitations.

Novelty detection is the identification (and classification) of new or unknown data that machine learning system has not been trained with and was not previously aware of, (Markou and Singh, 2003) using either statistical approaches or machine learning based algorithms. A machine learning system can never be trained with all the possible object classes and hence the performance of the network will be poor for those classes that are under-represented in the training set (Marsland, 2011) .This research looks into different techniques designed to detect backdoor attacks. It will also introduce a new method to deal with both seen as well as unseen backdoor threats.

Backdoors are often used for securing unauthorized remote access to a computer. The threat of backdoors surfaced with the network technology being adopted on a universal scale. A backdoor may take the form of a hidden part of a program (Wysopal and Eng, 2015) (application backdoor), a separate program (system backdoor), or a hardware feature. In any case, it remains a real challenge that computer security specialists have to deal with. Novelty detection is the identification (and classification) of new or unknown data that a machine learning system has not been trained with and was not previously aware of, (Markou and Singh, 2003) using either statistical approaches or machine learning based algorithms. A machine learning system can never be trained with all the possible object classes and hence the performance of the network will be poor for those classes that are under-represented in the training set (Marsland, 2011). This research looks into different techniques designed to detect backdoor attacks. It also introduces a new method to deal with both seen as well as unseen backdoor threats.

Related Work

This section presents a review of existing literature concerning on machine learning and the existing proposed techniques to tackle backdoor attack problems. It identifies some research gaps on machine learning research that motivated this research work.

Types of Machine Learning Techniques

Machine Learning techniques are categorized into three main types namely, single classifiers, hybrid classifiers, and ensemble classifiers. In single classifiers, the intrusion detection problem can be approached by using one single machine learning algorithm. In the literature, machine learning techniques such as k-nearest neighbour, support vector machines, artificial neural network, decision trees, self-organizing maps, etc. have been used to solve these problems.

In the development of an intrusion detection system, the ultimate goal is to achieve the best possible accuracy for the task at hand. This objective naturally leads to the design of hybrid approaches for the problem to be solved. The idea behind a hybrid classifier is to combine several machine learning techniques so that the system performance can be significantly improved. More specifically, a hybrid approach typically consists of two functional components. The first one takes raw data as input and generates intermediate results. The second one will then take the intermediate results as the input and produce the final results (P. Mishra et al., 2017).

Types of Machine Learning Techniques include artificial neural networks (ANN), decision trees (DT), fuzzy logic (FL), genetic algorithm (GA), K-nearest neighbour (k-NN) (Masri, Abou Assi, & El-Ghali, 2014), Naïve Bayes Networks, Self-organizing map (SOM) (Khorshed, Ali, & Wasimi, 2012), Support vector machines (SVM) (Ni, Li, Li, Zhang, & Ye, 2016), Figure shows taxonomy of malware detection using machine learning. In the same way also, we can use machine learning techniques and algorithms to detect backdoor programs with viruses and other malware.

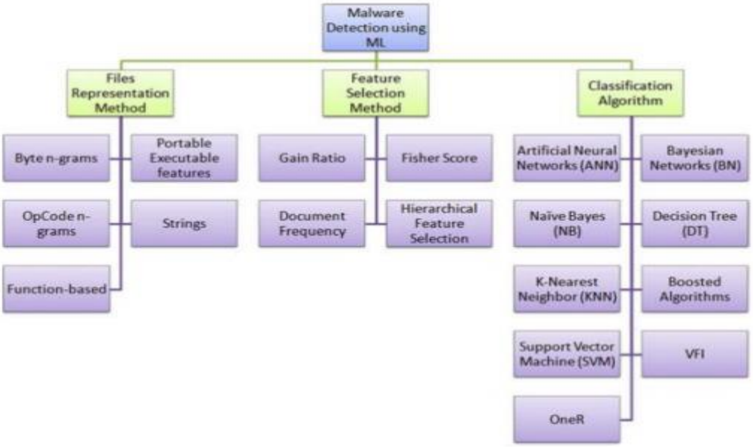

Figure A Taxonomy of Malware Detection Using Machine Learning Classifiers adapted from (Shabtai et al., 2009)

Waikato Environment for Knowledge Analysis (Machine Learning

The Waikato Environment for Knowledge Analysis (WEKA) tool was first developed at the Waikato University in New Zealand in 1995(Garner, 1995). The WEKA machine learning workbench is a modern platform for applied machine learning. WEKA is an acronym which stands for Waikato Environment for Knowledge Analysis. It is also the name of a New Zealand bird the WEKA. A machine learning workbench is a platform or environment that supports and facilitates a range of machine learning activities reducing or removing the need for multiple tools. The main reason to use WEKA is because a beginner can go through the process of applied machine learning using the graphical interface without having to do a lot of programming. This is important because getting a handle on the process, handling data and experimenting with algorithms is what a beginner should be learning about, not learning yet another scripting language.

Research Methodology

This section describes in detail the steps involved in the methodology of this research in order to solve the problem. An overview of the methodology steps is presented first to demonstrate the relation and interaction between the different phases of the research as shown in Figure 3.1. The next sub sections discuss the different research activities carried out in order to complete this research work.

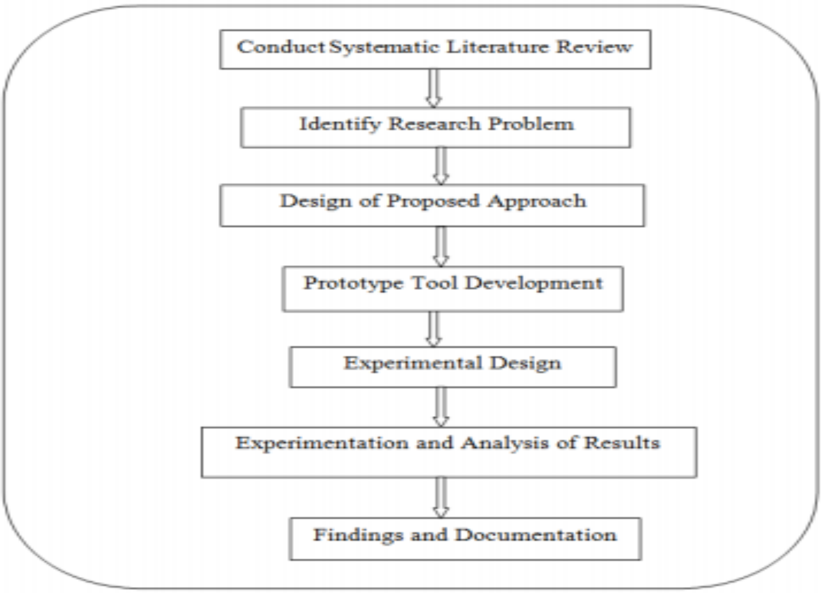
Conduct Systematic Literature Review
Identify Research Problem
Design of Proposed Approach
Prototype Tool Development
Experimental Design
Experimentation and Analysis of Results
Findings and Documentation

Figure 3.1. Research Methodology

The system developed during this research work uses a novelty detection method to detect if one or a set of examples differ from the previously seen examples. An intrusion detection system (IDS) (Seewald & Gansterer, 2010; Yu, Wang, Champion, Xuan, & Lee, 2011)generally has to deal with problems such as large network traffic volumes, highly uneven data distribution, the difficulty to realize decision boundaries between normal and abnormal behaviour, and a requirement for continuous adaptation to a constantly changing environment. Meeting these challenges means coming up with a system able to deal with present as well as future backdoor threats. Strategies for classification of network behaviours are typically divided into two categories: misuse detection and anomaly detection (Nissim et al., 2014).

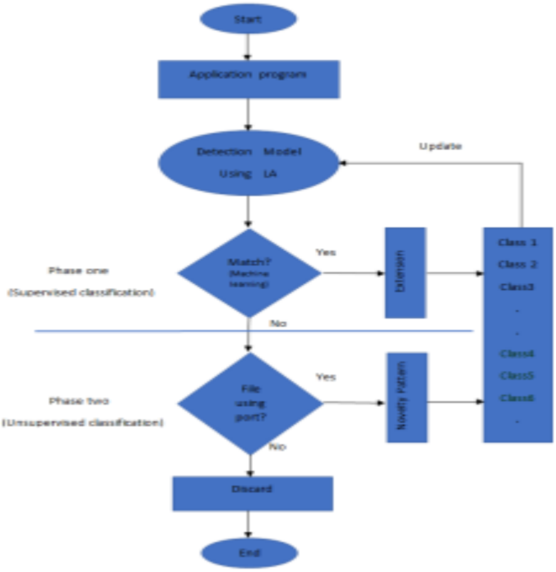

Figure 3.2. Backdoor Detection Model using multiclass learning algorithm (LA)

In phase one (Supervised Classification), of the proposed decision model, if a match is found the file is a backdoor and will be classified as such. In this case existing classes may change in size but not in number, which means no new classes are being created, but one of the existing classes is extended as the decision model is being updated.

In phase two (unsupervised Classification), a match was not found, the file will enter a process where we check if any open port is being used, if yes, the file will be classified as novelty (Backdoor), as such a new class will be created and the decision model is updated. Otherwise the file will not be classified as backdoor and will be discarded by the system.

Experimental Setup

Experiments were conducted to evaluate the effectiveness and efficiency of the proposed program. In order to evaluate the effectiveness of the proposed program, the results of the experiments are compared and analyzed. Based on the experimental design, experiments were conducted to get the difference in means between the measures of precision ratios of the compared programs. Any improvement by the proposed program in terms of effectiveness and efficiency is highlighted

Proposed Program Framework

The flow chart of the program algorithm used is shown in Figure 4.1. The algorithm starts by selecting the dataset to be used for the experiment. Next, it runs the KNearest Neighbour and Naïve Bayes algorithms against the selected dataset and detects any available backdoor in the program. For every detected backdoor, the program gives an alert and generates report using clustering and classification methods. The generated reports are saved in the database and are used in the future by the algorithms to detect similar backdoors as it continues learning.

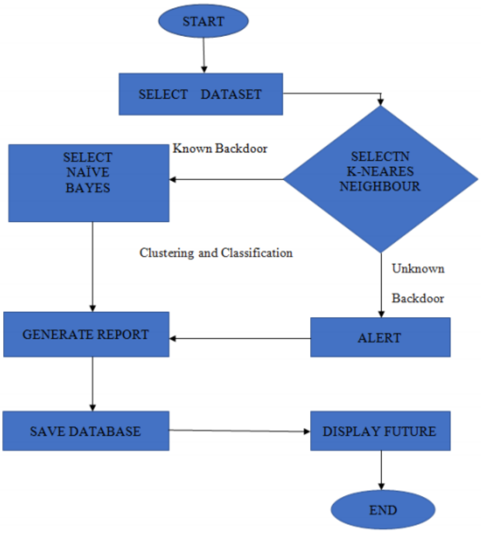

Figure. 4.1 Algorithm Flow Chart

Results and Discussion

The experiment results are described in the following sub sections with corresponding graphs. The first section shows the detection result for the program with the K-Nearest algorithm and the second section shows the results of the program with the Naive Bayes algorithms. A combination of these two algorithms is done and the result is shown in the third section by comparing it with Fuzzy Logic algorithm.

Results for K-Nearest Program

Figure 5.1 shows the results for the KNN program algorithm using clustering and classification unsupervised learning methods, which are clustering and classification are used to improve the methods. This is the first step in the backdoor detection process. Both supervised and effectiveness of the program. Once the program is run, if there is no problem it displays the clustering and classification graph to indicate that the program is working well. If the program is not working, then the graph will not be displayed. The program will then display an information message to indicate it is applying K-Nearest Neighbour algorithm. No detection is done yet because the program needs to run the second algorithm, which is shown in the next section.

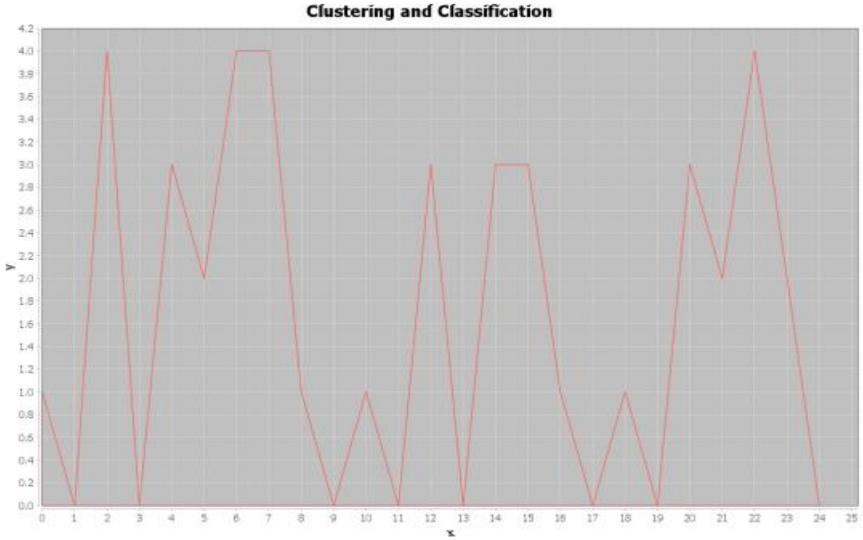

Figure 5.1 Graph Results for KNN

Results for Naive Bayes Program

Figure 5.2 shows the results for the Naïve Bayes program algorithm using clustering and classification methods. This is the second part of the detection process after running the K-Nearest Neighbour algorithm. In this part the Naive Bayes algorithm is run next to complete the detection process. After the program finishes running it shows an information message which indicates the results and the detection percentage.

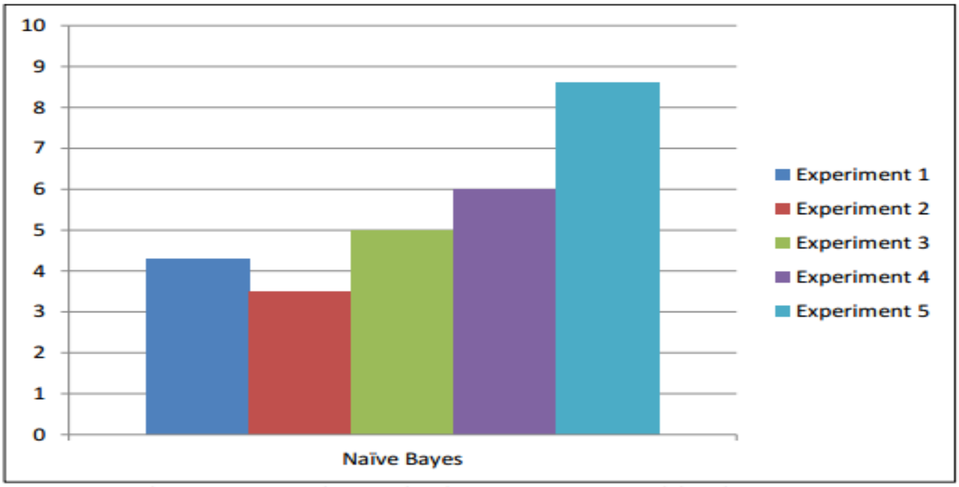

Figure 5.2 Graph Results for Naïve Bayes with Histogram

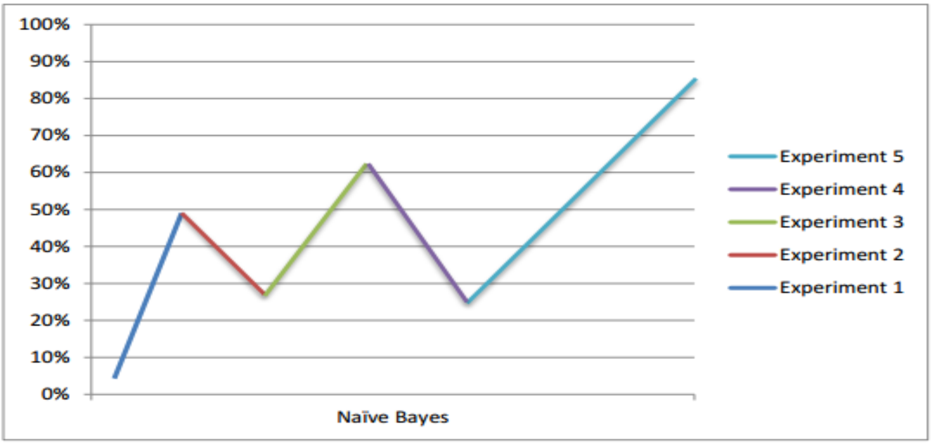

Figure 5.3 Graph Results for Naïve Bayes with Graph

Comparing KNN and Naive Bayes with Fuzzy Logic Clustering and Classification Methods

This section compares the combination of K-Nearest Neighbour algorithm and Naïve Bayes algorithm with Fuzzy Logic and other clustering and classification methods. After the comparison is found that the combination of K-Nearest Neighbour algorithm and Naïve Bayes algorithm works better than the other methods. Figures 5.4 and 5.5 show the comparison results.

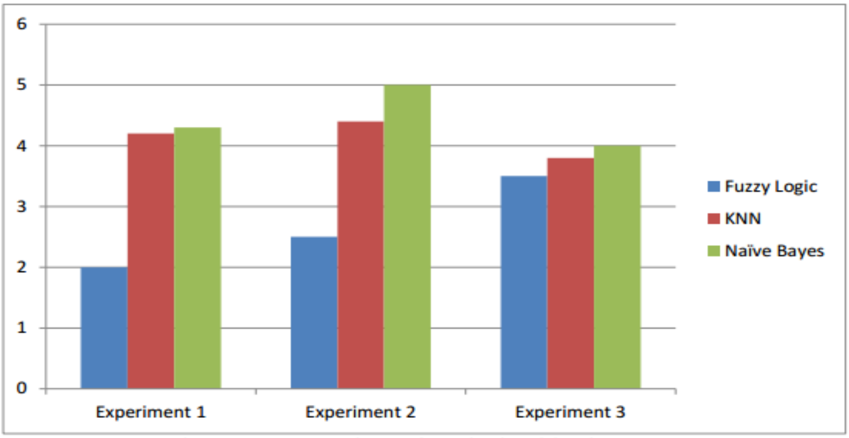

Figure 5.4 Comparison of Methods with Histogram

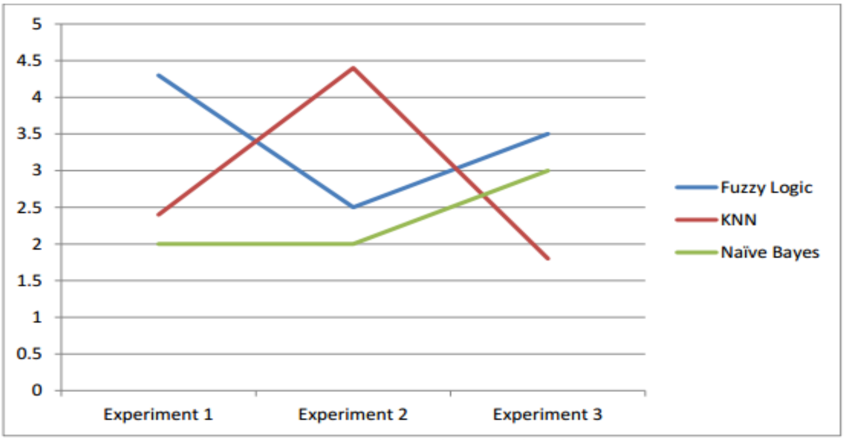

Figure 5.5 Comparison of Methods with Graph

The results of the experiment showed that the proposed program was able to effectively and efficiently detect backdoors in the datasets, thereby providing an improvement to backdoor detection techniques.

Conclusion

This work concentrated on application backdoors which are embedded within the code of a legitimate application. The proposed program in this research is an improvement to backdoor detection based on machine learning techniques. It is developed in Java and employs both supervised and unsupervised methods in the WEKA tool. This helps improving the detection compared to previous methods such fuzzy logic. The results of the experiments have proven that the proposed program is better at detecting backdoors than fuzzy logic when evaluated with similar data set. This proves that combining K-Nearest Neighbour and Naive Bayes algorithms is better than using Fuzzy Logic method. The program encountered less false positives and detected all backdoors in the dataset.

References

[1] Ilion Corona et al. Information fusion for computer security: State of the art and open issues. Information Fusion 10 (2009) 274-284.

[2] M. Marko, S. Singh, Novelty detection: A review, part 1: Statistical approaches, Signal Processing 83, 2481–2497, 2003.

[3] El aine Ribeiro de Farria, André Carlos Ponce de León Ferreira Carvalho, Joao Gama. (2015). MINAS: multiclass learning algorithm for novelty detection in data streams. Springer.

[4] Thompson, Ken, “Reflections on Trusting Trust”, Communication of the ACM Vol. 27, No 8, http//www.acm.org/classicas/sep95/, Sep 1995.

[5] Chi-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. Review: Intrusion detection by machine learning: A review. Expert Syst. Appl., 36(10):11994– 12000, December 2009.

[6] Manchu, S., & Girolami, M. A. (2007). An empirical analysis of the probabilistic K-nearest neighbour classifier. Pattern Recognition Letters, 28, 1818–1824.

[7] Mitchell, T. (1997). Machine learning. New york: McGraw Hill.

[8] Vapnik, V. (1998). Statistical learning theory. New York: John Wiley.

[9] Haykin, S. (1999). Neural networks: A comprehensive foundation (2nd ed.). New Jersey: Prentice Hall.

[10] Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. Biological Cybernetics, 43, 59–69.

[11] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, P. J. (1984). Classification and regressing trees. California: Wadsworth International Group.

[12] Pearl, J. (1988). Probabilistic reasoning in intelligent systems. Morgan Kaufmann.

[13] Koza, J. R. (1992). Genetic programming: On the programming of computers by means of natural selection. Massachusetts: MIT.

[14] Zimmermann, H. (2001). Fuzzy set theory and its applications. Kluwer Academic Publishers.

[15] Jang, J.-S., Sun, C.-T., & Mizutani, E. (1996). Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence. New Jersey: Prentice Hall.

[16] Kittler, J., Hatef, M., Duin, R. P. W., & Matas, J. (1998). On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(3), 226–239.

[17] Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. Review: Intrusion detection by machine learning: A review. Expert Syst. Appl., 36(10):11994– 12000, December 2009.

[18] Peter Mell Karen Scarfone. Guide to intrusion detection and prevention systems (idps). National Institute of Standards and Technology, NIST SP - 800-94, 2007. Available at http://www.nist.gov/customcf/get_pdf.cfm?pub_id=50951.