A Distributable Hybrid Intrusion Detection System for Securing Wireless Networks

David Tahmoush[1]
[1]University of Maryland, University College, Maryland, USA

Abstract - *We developed a hybrid design to a NIDS that enables the seamless insertion of a machine learning component into a signature NIDS system that significantly improves throughput as well as captures additional networking traffic that is similar to known attack traffic. The throughput improvement by incorporating a normalcy classifier is significant, estimated to be the inverse of the false alarm rate which can easily net a factor of 1000. However, this can be diminished by updates that can trigger a retraining of the normalcy classifier. The addition of a normalcy classifier front-end also makes the system more highly scalable and distributable than the signature-based NIDS. The new hybrid design also allows distributed updates and retraining of the normalcy classifier to stay up-to-date with current threats, and makes a number of important performance and quality guarantees. The distributable hybrid implementation is very useful for securing wireless networks with multiple access points.*

This system design also has the capability to recognize new attacks that are similar to known attack signatures. The hybrid design also can provide significant information on new attack traffic. By finding the signature of suspicious traffic that is similar to the signature of a known attack, it can be isolated and analyzed as a potential variant of a known attack.

Keywords: IDS, hybrid

1 Introduction

Machine learning classifiers can be used to discover the patterns hidden within large data sets, and one of the largest datasets is the information being passed through a network every day. Many information technology applications have been proposed and also used to classify network traffic [1, 2, 3, 4, 5]. Intrusion detection systems (IDS) monitor the system or network events and detect violations or threats to computer security policies, acceptable use policies, or standard security practices [6], and are one of the most significant counter measures [7, 8, 9, 10] against security threats. Intrusions can be found using signature based detection of known threats, but there are also anomaly detectors. Signature based detectors look for specific log entries or a specific payload in a data packet known to be indicative of misuse.

The IDS monitors the network traffic from a system or through a network and looks for any abnormal behavior in the network activity which indicates a possibility of unwanted and malicious network traffic and take appropriate action if such situation occurs. The IDS uses signature detection for specific known threats or anomaly detection for unknown threats to analyze the data. However, many unknown threats are merely updated versions of known threats. Since machine learning techniques can determine whether new threats are similar to known threats, there is the potential to combine anomaly detection with approximate signature detection. One of the most significant aspects of an IDS is the use of artificial intelligence [11] to train the IDS about possible threats. The Intrusion Detection can gather information about the various traffic patterns and rules can be formed based on these patterns, to distinguish between normal traffic and anomalous traffic in the network. Machine learning techniques have the ability to generalize from limited, noisy data that is not complete to broader categories on new data. This generalization capability provides the potential to recognize patterns similar to known patterns but not exactly matching. The IDS should ideally recognize not only previously observed attacks but also future attacks that have not yet been seen [12].

2 Machine Learning in Intrusion Detection Systems

Some significant contributions to IDS have been made using Fuzzy Logic. Fuzzy inference combined with artificial neural networks were used for real time traffic analysis by building a signature pattern database using protocol analysis and neuro-fuzzy learning techniques [13]. Fuzzy rule-based classifiers for IDS were modeled [14]. A fuzzy intrusion recognition engine (FIRE) used Fuzzy Logic and data mining techniques to produce fuzzy sets based on the input traffic data to detect security threats [15]. Association-based classification of normal and anomalous attacks was performed on the basis of a compatibility threshold [16]. Association rules along with data mining techniques and classification was used on suspicious events in real-time [17]. Fuzzy rules gave the best detection rate when compared to linear generic programming, decision trees, and support vector machines on the DARPA 1998 dataset [18]. Fuzzy logic with an expert system performed better than 91.5% detection rate over all attack types with a reduced complexity over traditional fuzzy number ranking techniques [19]. Fuzzy adaptive resonance theory have also been used to implement network IDS [20] as well as fuzzy rules [21, 22].

A lot of work has been done on IDS using genetic algorithms. Genetic algorithms using both temporal and spatial information of the network connection during the encoding phase were used to identify anomalous network behaviors [23]. Genetic algorithms were used to find the best possible fuzzy function and select the most significant network features [24]. Genetic programming was used to derive classification rules with traffic data on the network [25]. Multiple agent technology with genetic programming was used to detect anomalies in the network [26]. A combination of information theory to filter the traffic data with genetic algorithms was used to detect anomalous behaviors in the network with reduced complexity [27].

Artificial neural networks are a popular machine learning technique, and it has been applied to IDS. A hybrid neural network was proposed using a combination of Self-Organizing Map (SOM) and Resilient Back-Propagation Neural Network (BPNN) [28]. Another hybrid system using a BPNN and a C4.5 Decision Tree was built [29] which showed that the certain network attack types could not be detected without a hybrid system. A multi- layer artificial neural network was used to classify network activity [30]. A multi-classification IDS system was built that showed a higher detection rate in each classification category than when only a single class was used to classify all non-normal data [31].

Additional approaches have included graphlets [32], decision trees [33], clustering [34], and deviation from normal traffic [35].

3 Data

Many researchers have proposed IDS classification algorithms based on machine learning techniques, but they have used older datasets from DARPA and others to evaluate their approaches. This dataset used is a network packet dataset consisting of normal network activity as well as many network attack types. The dataset is based on the DARPA98 dataset from MIT Lincoln laboratory, which provides answer class (labeled data) for evaluation of intrusion detection [36]. This dataset was created in 1998 and lacks of many current attack types. This paper uses current signatures from an IDS as an oracle for machine learning to form a new, faster IDS with the generalization capabilities of a machine learning built in. This avoids the work of manually labeling a dataset and provides more current signature information, but the quality of the initial IDS information determines the baseline for the new artificial intelligence based IDS.

4 Real-Time Intrusion Detection

A system that can detect network intrusion while an attack is occurring is called a real-time detection system. There are very few real-time network IDS approaches. A real-time IDS using Self-Organizing Maps (SOM) to detect normal network activity and differentiate it from a DoS attack was proposed [37]. A Bayesian classification model for anomaly detection was also built [38]. A real-time IDS was built using two unsupervised neural network algorithms with a detection rate over 97%, separating normal traffic data from network attacks [39]. A real-time network IDS using fuzzy association rules could separate the normal network activity from network attacks [40]. A high-speed intrusion detection model using TCP/IP header information was built to detect denial of service (DoS) attacks [41].

One of the most widely used and well-known IDS is called SNORT, and it has become a standard in IDS [42]. SNORT is a commercial tool that does not use machine learning, basing its detection on regular expressions that match to known signatures of network attacks. Its attack signature rules are available only to their registered customers. The signature rules or patches have to be frequently updated and installed in order to detect current attack types or variations in known attack types.

Although some researchers are investigating real-time IDS with machine learning techniques, most of the work is based on accurate learning without good real-time performance measures and without good generalization capabilities. This paper presents a real-time hybrid design that can guarantee improved real-time performance with equivalent false alarm rates.

5 Advantages of a Machine Learning System

There are multiple advantages to a machine-learning based system over a signature-based system. A signature-based system needs to store attack signatures and download new signatures when they are updated, while a machine-learning system merely updates the weights on its classifier. A signature system can be difficult to parallelize with a shared signature database, while a machine-learned system can run multiple instances due to its lightweight nature. The speed of a machine learned system can be faster, and that advantage only increases with the growth in the size of the signature database to search over. The machine-learned system will have slightly more false positives and will not give detailed information about the true positives, so a signature IDS can be run on the output from the machine-learned system for labeling as well as false-positive reduction.

The primary advantage of a machine learning system over a signature system is the ability to generalize to new but similar data. This was the dream of machine learning with an IDS, that the IDS should ideally recognize not only previously observed attacks but also future attacks that have not yet been seen [12]. There are some systems that can generalize their detection well from learned attack patterns to new attack patterns [47], especially on probing attacks [48]. A machine learning system also has some ability to generalize to patterns not seen in the training data, and this was seen anecdotally in this project.

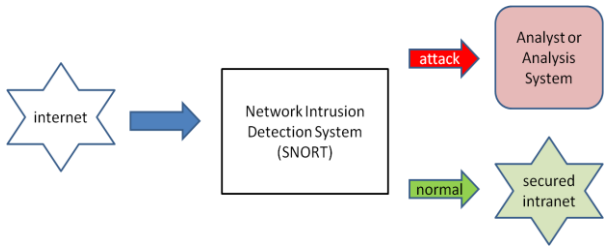

Figure 1. A standard setup for a network intrusion detection system.

6 Normalcy Classifier and a Hybrid Intrusion Detection System

A current network IDS setup is shown in Figure 1. Adding in a front-end with the capability to replicate the detections of a signature NIDS creates a hybrid system that can significantly improve the speed at equivalent false alarm rates but with a slightly higher false negative rate. For a hybrid system, the labeling and analysis of detection does not need to be implemented because a version of the signature

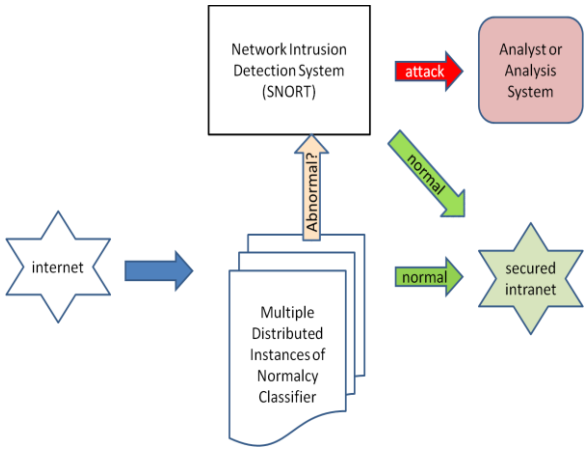

Figure 2. A hybrid setup for a network intrusion detection system with a distributed normalcy classifier. The network traffic considered abnormal by the normalcy classifier but normal by the signature system may contain new attack traffic that can also be analyzed.

NIDS should be run on the detections for labeling and analysis of the suspicious network traffic as shown in Figure 2. This hybrid system will outperform the signature NIDS as a standalone in speed since the high percentage of network traffic will be classified as normal and not sent to the labeler. It will also be more scalable, since additional normalcy classifiers can be run with significantly less overhead. The resulting system will produce the same level of labeling quality since the abnormal traffic would be routed through the signature component. The level of false alarms would not rise since the signature component would be run on the abnormal output from the normalcy classifier to reduce false alarms. The hybrid system would run faster, scale more easily, and use far less resources than a series of signature NIDS instances. The cost is the slightly increased false negative rate caused by missed detections in the normalcy classifier. However, the abnormal output from the normalcy classifier may contain significant information about a new or unrecognized attack pattern. This output can be sent to an analyst or to an anomaly classifier.

To understand the improvement in speed and capabilities of a hybrid IDS, consider a signature based system with a number of packets that it can analyze per second. Adding in a normalcy classifier with a 2% false positive rate would improve the number of packets that could be classified by a factor of fifty. If you can achieve a 1% false positive rate, the improvement in packets per second jumps to a factor of 100. For a given false positive rate *fp*, the performance boost can be estimated at *1/fp*. This estimate neglects the overhead of running the normalcy classifier or classifiers, but that is typically negligible compared to the signature-based component and can be run in parallel. With a 0.1% false positive rate, the performance boost is a factor of one thousand. Unfortunately, no normalcy classifier can be perfect, so there will be a cost in the false negatives from the normalcy classifier that will be passed through into the network. Typically there is an inverse relationship between the false positives that would require signature processing and the false negatives which incurs risk to the network. This implies that an optimal performance can be achieved by varying the normalcy classifier to process all of the network data given the constraint provided by the signature section. However, this adds complexity to the normalcy classifier which would complicate the retraining.

Retraining is a significant issue in a hybrid system, since the normalcy classifier should be retrained every time the signature database is updated. The complexity of a deployable normalcy classifier can be limited by the allowable retraining time if there are not other workarounds while the normalcy classifier is retraining. In the case where the normalcy classifier is taken offline while retraining and the signature component runs without hybrid support, the performance gains can be eroded. Given a training downtime *d*, the expected performance boost drops by a similar factor of *d*, for example a downtime of 20% drops the performance boost by 20%. In selecting a normalcy classifier, the cost of this training downtime may be a significant consideration. However, updates are one large advantage of a hybrid system over developing a brand-new system. If the normalcy classifier is trained on the signature NIDS outputs, a new signature inserted into the system can trigger retraining of the classifier and redistribution of the training weights. This leaves the development of new signatures in the signature-based component of the hybrid system and then distributes the signatures to the normalcy classifier.

One large advantage of a hybrid system is the guarantee of no increase in false alarms. Since the positives of the normalcy classifier are analyzed by the signature-based component, the

output will be the same as if the signature-based component was run on all of the data except for the increase in false negatives.

Another advantage to a hybrid system is the consistent labeling when running the output through the signature component. This provides additional incentives for developing a hybrid system over building from scratch. Utilizing a signature-based approach to consistent labeling of any suspicious traffic enables the use of additional software that analyzes those labeled packets. By using a hybrid approach, the insertion of a machine learning component into the current system can be relatively seamless because the signature system is not changed or replaced, merely augmented and improved.

The use of a distributable network IDS system can be very useful in wireless networks, where the network could be infiltrated at almost any node. A normalcy classifier front-end provides a small distributable section that could be used to help provide network security on wireless networks.

7 Possible Implementation

Network packets are small collections of text. An N-gram can be used to break up the text into series of letters of a specified length to be used for classification [43]. This maps a network data packet into a high dimensional space where machine learning can be challenging. The high dimensional space can be hashed into a lower dimensional space without losing the ability to directly match the same packet [44, 45]. However, the hash is not a unique identifier and other similar packets may have the same hash. This approximation makes the system run faster, but the approximation can result in a large number of false positives if the dimension of the hash is too small. The tradeoffs for development of a hybrid IDS include the complexity of the algorithm, the required size and speed for the target platform, the training time for processing updates, the acceptable loss in detections. The runtime and retraining time can also be affected by the complexity of the algorithm. One possible implementation like this explored some of the performance tradeoffs [46] like size and speed, but this is an area for greater exploration with a larger and more relevant data set.

Though the abnormal output from the normalcy classifier has been shown to contain some information about new or unrecognized attack patterns, this has not been well characterized and represents a significant area for future research. The use of a normalcy classifier to capture relevant attack network traffic and a signature NIDS to remove known attacks leaves a much smaller set of network traffic that is similar to a known attack, or suspicious traffic. By matching the signature of the suspicious traffic to the signature of the known attack that it is similar to may provide additional insights to the suspicious traffic. The suspicious traffic that is similar to known active attack traffic can be isolated and analyzed as a potential variant of a known attack.

8 Hybrid Design Guarantees

Several guarantees can be made with this hybrid design pattern for improved NIDS performance. First, the resulting system will produce the same level of labeling quality as original NIDS. Second, the level of false alarms would not rise since original NIDS would be run on the abnormal output from the normalcy classifier. Third, the hybrid system would run faster, scale more easily, and use far less resources than a series of NIDS instances. Fourth, the false negative rate is going to increase slightly. Fifth, the cost of development and more significantly the cost of support and maintenance are significantly less that developing a new NIDS. These guarantees can help mitigate the development risk and can be used to understand the system tradeoffs when considering the overall design.

9 Conclusions

We developed a hybrid design to a NIDS that enables the seamless insertion of a machine learning component into a signature NIDS system that significantly improves throughput as well as captures additional networking traffic that is similar to known attack traffic. The throughput improvement by incorporating a normalcy classifier is significant, estimated to be the inverse of the false alarm rate which can easily net a factor of 1000. However, this can be diminished by updates that can trigger a retraining of the normalcy classifier. The addition of a normalcy classifier front-end also makes the system distributable across the network and more easily scalable.

The hybrid design also can provide significant information on new attack traffic. By finding the signature of suspicious traffic that is similar to the signature of a known attack, it can be isolated and analyzed as a potential variant of a known attack.

10 References

[1] Maiolini, G., Baiocchi, A., Iacovazzi, A., & Rizzi, A. (2009). Real time identification of SSH encrypted application flows by using cluster analysis techniques. In NETWORKING 2009 (pp. 182-194). Springer Berlin Heidelberg.

[2] Chen, R. C., Cheng, K. F., & Hsieh, C. F. (2010). Using rough set and support vector machine for network intrusion detection. arXiv preprint arXiv:1004.0567.

[3] Este, A., Gringoli, F., & Salgarelli, L. (2009). Support Vector Machines for TCP traffic classification. Computer Networks, 53(14), 2476-2490.

[4] Horng, S. J., Su, M. Y., Chen, Y. H., Kao, T. W., Chen, R. J., Lai, J. L., & Perkasa, C. D. (2011). A novel intrusion detection system based on hierarchical clustering and support

vector machines. Expert systems with Applications, 38(1), 306-313.

[5] Nguyen, T. T., & Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. Communications Surveys & Tutorials, IEEE, 10(4), 56-76.

[6] Scarfone, K., & Mell, P. (2007). Guide to intrusion detection and prevention systems (idps). NIST special publication, 800(2007), 94.

[7] Yao, J. T., Zhao, S. L., & Saxton, L. V. (2005, March). A study on fuzzy intrusion detection. In Defense and Security (pp. 23-30). International Society for Optics and Photonics.

[8] Bace, R. G. (2000). Intrusion detection. Sams Publishing.

[9] Dasarathy, B. V. (2003). Intrusion detection. Information Fusion, 4(4), 243-245.

[10] Allen, J., Christie, A., Fithen, W., McHugh, J., & Pickel, J. (2000). State of the practice of intrusion detection technologies (No. CMU/SEI-99-TR-028).

[11] Bobor, V. (2006). Efficient Intrusion Detection System Architecture Based on Neural Networks and Genetic Algorithms. Department of Computer and Systems Sciences, Stockholm University/Royal Institute of Technology, KTH/DSV.

[12] Han, S. J., & Cho, S. B. (2005). Evolutionary neural networks for anomaly detection based on the behavior of a program. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 36(3), 559-570.

[13] Chavan, S., Shah, K., Dave, N., Mukherjee, S., Abraham, A., & Sanyal, S. (2004, April). Adaptive neuro-fuzzy intrusion detection systems. In Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on (Vol. 1, pp. 70-74). IEEE.

[14] Abraham, A., Jain, R., Thomas, J., & Han, S. Y. (2007). D-SCIDS: Distributed soft computing intrusion detection system. Journal of Network and Computer Applications, 30(1), 81-98.

[15] Dickerson, J. E., & Dickerson, J. A. (2000). Fuzzy network profiling for intrusion detection. In Fuzzy Information Processing Society, 2000. NAFIPS. 19th International Conference of the North American (pp. 301-306). IEEE.

[16] Tajbakhsh, A., Rahmati, M., & Mirzaei, A. (2009). Intrusion detection using fuzzy association rules. Applied Soft Computing, 9(2), 462-469.

[17] Barbará, D., Couto, J., Jajodia, S., & Wu, N. (2001). ADAM: a testbed for exploring the use of data mining in intrusion detection. ACM Sigmod Record, 30(4), 15-24.

[18] Abraham, A., & Jain, R. (2005). Soft computing models for network intrusion detection systems. In Classification and clustering for knowledge discovery (pp. 191-207). Springer Berlin Heidelberg.

[19] Liao, N., Tian, S., & Wang, T. (2009). Network forensics based on fuzzy logic and expert system. Computer Communications, 32(17), 1881-1892.

[20] Ngamwitthayanon, N., Wattanapongsakorn, N., & Coit, D. W. (2009). Investigation of fuzzy adaptive resonance theory in network anomaly intrusion detection. In Advances in Neural Networks–ISNN 2009 (pp. 208-217). Springer Berlin Heidelberg.

[21] Toosi, A. N., & Kahani, M. (2007). A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. Computer communications, 30(10), 2201-2212.

[22] Tsang, C. H., Kwong, S., & Wang, H. (2007). Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. Pattern Recognition, 40(9), 2373-2391.

[23] Li, W. (2004). Using genetic algorithm for network intrusion detection. Proceedings of the United States Department of Energy Cyber Security Group, 1-8.

[24] Bridges, S. M., & Vaughn, R. B. (2000, October). Fuzzy data mining and genetic algorithms applied to intrusion detection. In Proceedings twenty third National Information Security Conference.

[25] Lu, W., & Traore, I. (2004). Detecting new forms of network intrusion using genetic programming. Computational Intelligence, 20(3), 475-494.

[26] Crosbie, M., & Spafford, G. (1995, November). Applying genetic programming to intrusion detection. In Working Notes for the AAAI Symposium on Genetic Programming (pp. 1-8). MIT, Cambridge, MA, USA: AAAI.

[27] Xia, T., Qu, G., Hariri, S., & Yousif, M. (2005, April). An efficient network intrusion detection method based on information theory and genetic algorithm. In Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International (pp. 11-17). IEEE.

[28] Jirapummin, C., Wattanapongsakorn, N., & Kanthamanon, P. (2002, July). Hybrid neural networks for intrusion detection system. In Proceedings of International Conference on Circuits, Computers and Communications (pp. 928-931).

[29] Pan, Z. S., Chen, S. C., Hu, G. B., & Zhang, D. Q. (2003, November). Hybrid neural network and C4. 5 for misuse detection. In Machine Learning and Cybernetics, 2003 International Conference on (Vol. 4, pp. 2463-2467). IEEE.

[30] Moradi, M., & Zulkernine, M. (2004, November). A neural network based system for intrusion detection and classification of attacks. In Proceedings of the 2004 IEEE international conference on advances in intelligent systems-theory and applications.

[31] Ngamwitthayanon, N., Wattanapongsakorn, N., Charnsripinyo, C., & Coit, D. W. (2008). Multi-stage network-based intrusion detection system using back propagation neural networks. In Asian International Workshop on Advanced Reliability Modeling (AIWARM), Taiwan (pp. 609-619).

[32] Pukkawanna, S., Visoottiviseth, V., & Pongpaibool, P. (2007, November). Lightweight detection of DoS attacks. In Networks, 2007. ICON 2007. 15th IEEE International Conference on (pp. 77-82). IEEE.

[33] Lee, J. H., Lee, J. H., Sohn, S. G., Ryu, J. H., & Chung, T. M. (2008, February). Effective value of decision tree with KDD 99 intrusion detection datasets for intrusion detection system. In Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on (Vol. 2, pp. 1170-1175). IEEE.

[34] Katos, V. (2007). Network intrusion detection: Evaluating cluster, discriminant, and logit analysis. Information Sciences, 177(15), 3060-3073.

[35] Chen, C. M., Chen, Y. L., & Lin, H. C. (2010). An efficient network intrusion detection. Computer Communications, 33(4), 477-484.

[36] Lee, W., Stolfo, S. J., & Mok, K. W. (1999, August). Mining in a data-flow environment: Experience in network intrusion detection. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 114-124). ACM.

[37] Labib, K., & Vemuri, R. (2002). NSOM: A real-time network-based intrusion detection system using self-organizing maps. Networks and Security, 1-6.

[38] Puttini, R. S., Marrakchi, Z., & Mé, L. (2003, March). A Bayesian classification model for real-time intrusion detection. In AIP Conference Proceedings (pp. 150-162).

[39] Amini, M., Jalili, R., & Shahriari, H. R. (2006). RT-UNNID: A practical solution to real-time network-based intrusion detection using unsupervised neural networks. Computers & Security, 25(6), 459-468.

[40] Su, M. Y., Yu, G. J., & Lin, C. Y. (2009). A real-time network intrusion detection system for large-scale attacks based on an incremental mining approach. Computers & security, 28(5), 301-309.

[41] Li, Z., Gao, Y., & Chen, Y. (2010). HiFIND: A high-speed flow-level intrusion detection approach with DoS resiliency. Computer Networks, 54(8), 1282-1299.

[42] Chakrabarti, S., Chakraborty, M., & Mukhopadhyay, I. (2010, February). Study of snort-based IDS. In Proceedings of the International Conference and Workshop on Emerging Trends in Technology (pp. 43-47). ACM.

[43] Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. Computational linguistics, 18(4), 467-479.

[44] Shi, Q., Petterson, J., Dror, G., Langford, J., Strehl, A. L., Smola, A. J., & Vishwanathan, S. V. N. (2009). Hash kernels. In International Conference on Artificial Intelligence and Statistics (pp. 496-503).

[45] Shi, Q., Petterson, J., Dror, G., Langford, J., Smola, A., & Vishwanathan, S. V. N. (2009). Hash kernels for structured data. The Journal of Machine Learning Research, 10, 2615-2637.

[46] Chang, R. J., Harang, R. E., & Payer, G. S. (2013). Extremely Lightweight Intrusion Detection (ELIDe), ARL-CR-0730.

[47] Kumar, G., Kumar, K., & Sachdeva, M. (2010). The use of artificial intelligence based techniques for intrusion detection: a review. Artificial Intelligence Review, 34(4), 369-387.

[48] Hwang, T. S., Lee, T. J., & Lee, Y. J. (2007, June). A three-tier IDS via data mining approach. In Proceedings of the 3rd annual ACM workshop on Mining network data (pp. 1-6). ACM.