International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 08 | Aug 2020 www.irjet.net p-ISSN: 2395-0072

STUDY OF EVALUATION FRAMEWORK FOR NETWORK INTRUSION DETECTION USING FUZZY LOGIC

Nabonarayan Jha[1], Jaynarayan Jha[2], Anand Singh[3]

[1]*Research Scholar, Department of Mathematics, B.R.A. Bihar University, Muzaffarpur, Bihar, India.*
[2]*Department of Mathematics, Padma Kanya Multiple Campus, Tribhuvan University, Nepal.*
[3]*Kantipur City College, Kathmandu, Purbanchal University, Nepal.*

Abstract - *This paper focuses on mining the most useful network features for attack detection. Accordingly, we propose a new network feature classification schema as well as a mathematical feature evaluation procedure that helps us identify the most useful features that can be extracted from network packets. The network feature classification schema is intended to provide a better understanding, and enforce a new standard, upon the features that can be extracted from network packets, and their relationships. The classification has a set of 27 feature categories based on the network abstractions that they refer to (e.g., host, network, connection, etc.). We use our feature classification schema to select a comprehensive set of 671 features for conducting and reporting our experimental findings.*

Key Words: **Fuzzy Logic, Game Theory, Intrusion Detection System.**

1. INTRODUCTION

The design of a Network Intrusion Detection System (NIDS) is a delicate process which requires the successful completion of numerous design stages. The feature selection stage is one of the first steps that needs to be addressed, and can be considered among the top most important ones. If this step is not carefully considered the overall performance of the NIDS will greatly suffer, regardless of the detection technique, or any other algorithms that the NIDS is using. The most common approach for selecting the network features is to use expert knowledge to reason about the selection process. However, this approach is not deterministic, thus, in most cases researches end-up with completely different sets of important features for the detection process.

Furthermore, the lack of a generally accepted feature classification schema forces different researches to use different names for the same (subsets of) features, or the same name for completely different ones. It is our belief that these issues are not sufficiently studied and explored by the network security research community.

The feature evaluation procedure provides a deterministic approach for pinpointing those network features that are indeed useful in the attack detection process. The procedure uses mathematical, statistical and fuzzy logic techniques to rank the participation of individual features into the detection process. In particular, we purpose a new feature dependency measure for independent evaluation criteria that is, to our knowledge, a pioneer method designed for intrusion detection.

In our research we have identified several tuning parameters that directly influence the detection performance of each individual feature. To address this issue, our method takes into account the performance of each feature while using multiple tunings, making the evaluation process more robust to biases that could be accidentally introduced by a poor tuning combination.

2. RESEARCH MOTIVATION

The design of a NIDS is a delicate process that requires the successful completion of numerous stages so that the final outcome can be considered a success. The feature selection stage is one of the first steps that needs to be addressed, and can be considered among the top most important ones. This step will definitely influence the performance of any detection engine, regardless of the techniques that the engine uses.Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

One of the main challenges when dealing with the amount of information that can be extracted directly from network packets is to create a set of features that covers most of the information space. The network features are constructed around the main abstractions of the network security domain such a packet, connection, host, and network. The idea of constructing features that will cover a reasonable part of the information space is especially hard due to the diversity of data that passes through a network link and the lack of unanimously accepted network feature classification schema in the research community. Although theoretically it is possible to design a system that can extract and use a wide range of features, due to constraints such as, large computational time, diversity of protocols and applications that exist, and amount of memory that the NIDS needs, most of the implementation make tradeoffs concentrating only on a

particular set of intrusions (e.g., R2U, U2R, and DoS in [18;21]; Horizontal, Vertical, and Block Port Scanning for TCP and UDP in Denial of Quality of Service Worms in [7]; DDoS in TCP-SYN Flooding Attacks in TCP-SYN Flood, and ICMP Flood Attacks).

Furthermore, researchers have empirically demonstrated that false correlations between the features that may be extracted by an IDS/IPS can lead to purer results concerning the accuracy and performance of the detection system. For example, if only 17 carefully selected features are used among all 41 features provided in the international Knowledge Discovery and Data Mining Competition (KDD-1999) [2], the detection rate will not change, but the speed of detection algorithm will improve by about 50%. The lack of a widely accepted network feature classification schema also adds to the general confusion regarding the set of features that should be used for detection purposes. Moreover, different researchers use different name for same subset of features, while other use the same name for completely different types. There seems to be at least two main type of features that are widely accepted in the literature. This distension are done between features that are computed based on a single TCP connection and features that are computed based on multiple TCP connections. However, the names of previous mentioned categories are different from researcher to researcher. Accordingly, the single TCP connection features are referred to as *Basic Features in* [18;21], *Essential Attributes in Basic Features of individual TCP connection in* [2], and Basic TCP features. A superset of this feature category that also includes connectionless protocols is mentioned as *Flow Statistic.* Similarly, different names are used for the second features of category too, such as *Derived Features* in [18;21], and *Traffic Feature* in [2]. The importance of selecting a set of appropriate features has been identified several decades ago. Starting in the last seventies and continuing to date, in the fields of pattern recognition [6], statistics , as well as Artificial Intelligence (AI) [16;19;13;20;14]. There are two main approaches to evaluate the performance of a feature namely the *dependent evaluation criteria* and the *independent evaluation criteria.* The *dependent evaluation criteria* uses the performance of the detection engine as a primary factor to evaluate the suitability and effectiveness of a feature or a set of features. Thus, better the detection engine performs the better feature. Even though this type of feature evaluation method is straightforward it has two main disadvantages. First, it is computationally intensive and secondly the final result is biased by the detection algorithm which tends to reward those features that perform better with that particular detection engine or algorithm mining. Conversely, the *independent evaluation criteria* uses the intrinsic characteristics of the data alone to evaluate the selected features. By studying the actual value of the feature while in the intrusive and normal stages, the final outcome would not be biased by any detection algorithm and thus we consider this method to be more adequate for the feature selection purposes. Furthermore, we believe that defining an *independent evaluation criteria* metric for evaluating the network features will allow the researchers to use the deterministic way to identify those features that are highly important to the detection process. Despite the need for such a method, to our knowledge, there isn't any work that uses *independent evaluation criteria* for features in intrusion detection. This thesis provides both, a hybrid feature evaluation method that combines statistical and fuzzy logic concepts for feature evaluation, as well as, a feature classification scheme for network intrusion detection, which provides a better understanding of the huge number of features that can be extracted from network packets. Despite the fact that we concentrate our research on Transport, Network, and Network Access layers of the TCP/IP Architecture Model, the proposed feature evaluation method as well as the feature classification schema can easily be applied to other layers of the TCP/IP Architecture Model, or OSI standard. We analyze the performance of a set of 671 network features that are directly extracted from the network packets. Furthermore, each of the studied features has various tuning parameters that play an important role in evaluation process. The tuning sensitivity of a feature is factored in by studying a set of 180 different tuning values for each one of the studied features. In order to evaluate our results, we have carried out our experiments on three different real-world network datasets. The experimental results, empirically confirm that the proposed feature evaluation model can successfully be applied to determine the importance of a feature in the detection process.

3. SUMMARY OF CONTRIBUTIONS

The main aim of this research is to identify and classify the network features that can be extracted from Ethernet, IP, TCP, UDP, and ICMP protocols, as well as to statistically study their importance in context of detecting the main types of network attacks. The main contributions of this paper are listed below.

- Identifying and studying more than 671 network features related to each packet encountered in the network;
- Introducing a novel classification schema regarding the network features that can be used for intrusion detection.
- Proposing and identifying the types of tunings that can be used to adjust the performance of each features based on its feature type.

- Proposing and implementing a real-time feature construction module that extracts all the 671 features.
- Proposing, for the first time, an *independent criteria* method for single feature evaluation in the domain of intrusion detection and,
- Identifying the set of features that are most suitable for intrusion detection in the case of Denial of Services (DoS), Probing, and Remote to Local (R2L) attacks.

4. DATA COLLECTION IN INTRUSION DETECTION

Data acquisition is one of the biggest challenges that a network security system must undertake. The decision on the amount of data, and the type and place of the data capturing process dramatically influences not only the performance of the system, but also its trustworthiness and detection scope. Based on the type of data that is collected, the IDSs can be classified into five main categories as follows:

i. Application-integrated IDSs: are those IDSs that collect the data out of a single application. They have an embedded sensor inside the application itself that collects and sends the extracted features to the IDS for processing.

ii. Application-based IDSs: are those IDSs that monitor only one application by transparently collecting necessary data. This is done by the use of external sensors that detect and capture the data exchanged between the monitored application and those third party entities (e.g., applications, hosts) that it interacts with.

iii. Host-based IDSs: evaluate and keep track of the wellness of a host as an entity by monitoring its applications.

iv. Network-based IDSs: are those IDSs that are meant to protect the network from itself. The network is pictured as a collection of hosts that interact by exchanging messages that are transferred through wires (i.e., in the case of a wired network) or radio waves (i.e., in the case of wireless network). The sensor of IDS collects data by sniffing it directly from the network traffic in a transparent way.

v. Hybrid IDSs: use two or more of the previously described techniques in order to collect data.

5. PROPOSED MODEL FOR THE FEATURE EVALUATION

The proposed model for the feature evaluation to perform a comprehensive feature evaluation, we need to apply a good evaluation technique over a relative large number of network features. To ensure that the selected features reasonably cover all the types of data can be extracted at the network level, we purpose a *Feature Classification Schema* for network features that is also intended to impose the first centralized standard in the research community regarding the various types and names of the network features

We study various tuning parameters that play a significant role in the performance of each feature. It is widely known that a poor tuning step will lead to poor detection performance, as opposed to a reasonable tuning that will significantly boost the attack detection chances. Finally, we explain our proposed feature evaluation technique. The technique uses a combination of statistical and fuzzy logic methodologies to access the performance of a feature against a particular type of attack. The assessment takes into consideration not only the performance of each features against the current type of attack, but also the potential of the feature for producing misclassified examples and false positives. More specifically, the proposed work falls into the *feature dependency measure* class of the *independent evaluation criteria* technique for the subset evaluation step of the *feature evaluation* procedure.

5.1 Defining the membership functions for the output

We have previously identified the number of linguistic terms used to describe the output, as well as their semantical order, that is: *best, better, good, bad, worse,* and *worst.* Next, the procedure for defining the membership functions uses the same method that was previously used in except that in the case there are six linguistic terms. Figure 1 graphically depicts the resulting membership functions.

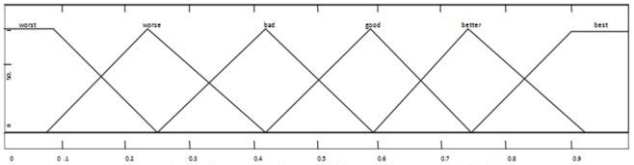

Figure 1: The membership functions for the output of the second FIE –graphical detail.

6. STUDY OF ARCHITECTURE AND IMPLEMENTATION

The whole system is implemented by using a mixture if C++, Java and MATLAB languages as we show it fit for certain tasks. For instance, we used C++ under Linux to implement the feature extraction process, since this task is time critical and is designed to work on-line and in real-time. Conversely, some parts of the feature evaluation process were implemented in MATLAB and Java under Windows since those task were not time critical ones, and also because MATLAB and Java languages offer a high number of libraries that facilities the implementation process.

The remaining part of this chapter is organized as follows. The overall top-level design of the implementation, followed by implementation details for all of the major processing blocks given. The conclusions are drawn.

6.1 Top-level Design

The overall feature extraction and evaluation process is depicted in figure 2. As seen in the figure, there are three types of input data that the system requires as follows: the tuning combinations, the actual data, and the labels of the data. The system is design to read from already saved TCP dump files, and can be easily extended to sniff packets directly from the network. In order to analyze the data, the system requires it to be labeled. In the best case scenario this can be provided upfront through a labeling file, or can be appropriated at the runtime. Thus, the system is designed to be independent of the data labels, which makes the third input of the system optional .The first processing unit that the data goes through is the *Framework for Real Time Network Feature Construction* module. This module is design to be a highly customizable, stand-alone application that, based on the feature classification schema proposed produces a total of 671 features for each individual packet that it encounters.

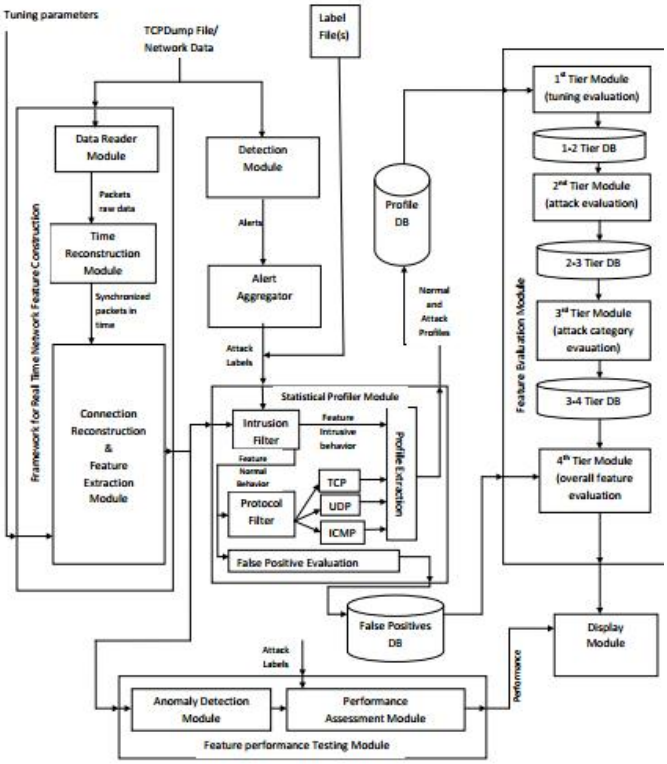

Figure 2: The overall view of the implementation block diagram.

Once the features are produced they are sent to the *Statistical Profiler Module,* where profile creation, protocol and attack filtering happens. Since for attack filtering the system needs to know the exact label of the data, in parallel with the feature extraction process, there is a packet labeling process which consists of a *Detection Module* (see Section 4.2), and an *Alert Aggregator* (see Section 4.3) that works together for attack label extraction. These two modules will be bypassed in the event when the labels are already provided through an external labeling file.

The profiles produced by the *Statistical Profiler Module* are temporarily saved in the custom-made database, which allows the next processing level to use them once all the profiles are computed. Furthermore, a second functionality of the *Statistical Profiler Module* is to estimate the potential number of false positives that each feature may produce under certain tunings. Once this task is done, the results are saved in the *False Positives DB.* The last processing step is performed by the *Feature Evaluation Module.* This module implements the core of our proposed evaluation techniques as described in Section 3.3. It has input both the feature profile as well as the feature false positive estimation, and it produces the final results.

In order to verify the experimental results, the packets and the associated features that are produced by the *Connection Reconstruction & Feature Extraction Module* are also sent to a *Feature Performance Testing Module* that empirically evaluates the performance of each individual feature and sends the results to Display Module.

7. CONCLUSION AND FUTURE WORK

Internet and data fraud has become one of the most challenging cybernetic acts that security officers around the world try to combat. The more critical and confidential the data is more appealing it is for attackers. The impact of a successful attack on an institution can have disastrous consequences such as privacy breach, data loss, or service interruption. Researchers around the world constantly develop and improve NIDS that are meant to combat such threats. For a NIDS to function properly all of its building blocks and processing components need to be properly designed. The feature selection stage is one of the first steps that needs to be addressed. This step can be considered among the top most important ones, since the overall performance and detection scope of the NIDS directly depends on it. The lack of a generally accepted feature classification schema as well as the lack of a deterministic network feature evaluation process make the feature selection stage an expert driven task that does not necessarily lead to the selection of the best features for the detection purposes. Despite its importance we believe that the feature selection phase for intrusion detection has not been sufficiently studied and explored by the research community. The main focus of this research is on mining the most useful network features for attack detection. In order to do this, we proposed a network feature classification schema as well as a deterministic feature evaluation procedure that helps to identify the most useful feature that can be extracted from network packets. To our knowledge, the proposed work pioneer the research in the field of *independence evaluation criteria* for network intrusion detection. The network feature classification schema is meant to provide a better understanding upon the types of features that can be extracted from the network packets. The feature evaluation procedure is meant to provide a deterministic way for pinpointing those features that are useful in the attack detection process. In order to be able to conduct our experimental results, we first used our proposed feature classification schema to select a comprehensive set of network feature. This set includes features from each of the 27 feature categories that the classification schema has. Next, based on the underlying implementation of each type of feature we have identified a set of four tuning parameters that dramatically influence the performance of each feature. Throughout the experimental procedure we used a set of 180 different values to observe the performance of each feature, its stability and sensitivity of different tunings. Next, we consider three distinct datasets that helps us experiment the results of our evaluation procedures against different types of data. These three data sets are collected from three completely different networks. The commonality between the three datasets is that they have real network attacks. The difference however is in the time of collection, size of the network, throughput, and also the types of users that the networks have. This diversity helps us better understand the output of our evaluation techniques under different networks and configurations. Finally, the proposed feature evaluation method is applied on all the features, tunings, and datasets to produce the final results. The procedure uses mathematical, statistical and fuzzy logic techniques to rank the participation of individual features into the detection process. The presented experimental results empirically confirm that the feature evaluation model can successfully be applied to mine the importance of a feature in the detection process. To evaluate our results we used a mining algorithm and study its performance over the selected features. The empirical results show that the performance of the mining algorithm was directly proportional with the feature importance as computed by the feature evaluation algorithm. Furthermore, we have also shown that features that are found in two or more datasets are ranked higher by the proposed feature that are found in two or more are ranked higher by the proposed feature evaluation algorithm than the ones that are found In only one dataset.

7.1 Future work

The proposed work explores a new direction in the feature selection process using a independent evaluation criteria procedure for intrusion detection. Thus, there are many extensions avenues of this work that can be explored in the future; some of those are:

- The current work explores the performance of each feature independent of others. This should be only the first step in the evaluation process. Since this step is already established, it can be further used for design of a feature evaluation method that will consider, at the next level, the feature interdependencies and their influence over the final detection performance.

- The current uses the three databases for the evaluation process; however, none of the databases are ideal for this task. An ideal database for the feature selection process will need to satisfy several requirements such as:

1. Labels need to be provided at packet level

2. The normal data needs to be collected (and filtered of attacks) from a real network.

3. There must be a considerable number of attacks from each attack type that is to be considered in the feature evaluation process.

4. Both attacks and normal data must be real, not simulated.

- The current work does not differentiate the final results based on the speed of the attacks. We believe that an interesting further study would be to analyze the set of features that are appropriate for fast or slow attacks. However, to do that, the dataset that will be used needs to have an equal number of attacks in each of the attack categories that will be studied.

- Even though the current work considers different tuning values for evaluating the features, further exploration in this research direction will ideally lead to tuning recommendations for different networks. This task is quite challenging since, to do this, several factors need to be considered as follows:

1. Multiple datasets need to be considered at the same time. These datasets need to be extracted from a set of diverse networks.

2. Each dataset needs to have the same set of attacks so that a comparison between the feature/tuning performance can be made. Having the same set of intrusions in multiple databases will allow the researchers to consistently compare the feature/tuning performance between datasets and to report the best tuning values.

3. The reporting of the tuning values needs to be done proportional with various attributes (such as throughput and size) of the network since a specific value (e.g., 5) will not have any meaning with respect to other databases

The result of the current work can be used to create a labeling technique for unsupervised anomaly detection systems. This goal should be easily achieved since it is already known which features are best to detect certain types of attacks, and by monitoring their values in parallel with the detection system, a method can be proposed to create meaningful attack labels

The normal data needs to be collected (and filtered of attacks) from a real network.

BIBLIOGRAPHY

[1] Tcpdump public repository, http://www.tcpdump.org/, March 10, 2005, last access.

[2] KDD 99, The fifth international conference on knowledge discovery and data mining, http://kdd.icus.uci.edu, Website, Last accessed October 2005.

[3] J. L. Verdegay A. Sancho-Royo, Methods for construction of membership functions, vol. 14, 1999, pp. 1213 {1230.

[4] Magnus Almgren and Ulf Lindqvist, Application-integrated data collection for security monitoring, Processing of Recent Advances in Intrusion Detection, 4th International Symposium, (RAID 2001) (Davis, CA, USA) (W, L. M Lee, and A. Wespi, eds.), Lecture Notes in Computer Science, Springer-Verlag Heidelberg, October 2001, pp.22{36.

[5] R. Basu, R.K. Cunningham, S. E. Webster, and R. P. Lippmann, Detecting low-profile probes and novel denial-of-services attacks, Proceedings of the workshop on Information Assurance and Security, United States Military Academy (IEEE, ed.), June 2001, pp. 5{10.

[6] M. Ben-Bassat, Handbook if statistic 2: Classification, pattern recognition and reduction of dimensionality, ch. Use of Distance Measures, Information Measures and Error Bounds in Feature Evaluation, pp. 773{791, North Holland, 1982.

[7] V. Berk, G. Bankos, and R. Morris, Designing a framework for active worm detection on global networks, Proceedings of the IEEE International Workshop on Information Assurance (Darmstadt, Germany), March 2003, pp. 13{23.

[8] Joachim Biskup and Ulrich Flegel, Transaction-based pseudonyms in adult data for privacy respecting intrusion detection, Proceeding of Recent Advances in Intrusion Detection, 3rd International Symposium, (RAID 2000) (Toulouse, France) (H. Debar, L. M, and S. F. Wu, eds.), Lecture notes in Computer Science, Springer-Verlag Heidelberg, October 2000, pp. 28{48.

[9] A.Blum and R. L. Rivest, Training a 3-node neural network is np-complete, COLT '88: Processing of the first annual workshop on Computational learning theory (San Francisco, CA, USA), Morgan Kaufmann Publisher Inc., 1988, pp. 9{18.

[10] A.L. Blum and P. Langley, Selection of relevant features and examples in machine learning, Artificial Intelligence 97 (1997), no. 1,245{271.

[11] DARPA, Darpa intrusion detection and evaluation dataset 1999, http://www.ll.nit.edu, Website, Last accessed February 2006.

[12] K. Das, The development of stealthy attacks to evaluate intrusion detection systems, Master's thesis, MIT Department of Electrical Engineering and Computer Science, June 2000.

[13] M. Dash, K. Choi, P. Scheuermann, and H. Liu, Feature selection for clustering –a filter solution, Data mining, 2002. ICDM 2002. Proceedings. 2002 IEEE International conference on (2002), 115{122.

[14] M. Dash, H. Liu, Feature selection for clustering, PADKK '00: Proceeding of the 4th Pacific-Asia Conference on Knowledge and Discovery and Data Mining, Current Issues and New Applications (London, UK), Springer-Verlag, 2000, pp. 110{121.

[15] M. Dash, H. Liu, and J. Yao, Dimensionality reduction of unsupervised data, Tools with Artificial Intelligence, 1997. Proceeding., Ninth IEEE International Conference on, no. 3-8, November 1997, pp. 532{539.

[16] P. A. Devijver and J. Kittler, Pattern recognition a statistical approach, Prentice Hall International, 1982.

[17] J. Doak, An evaluation of feature selection methods and their application to computer security, Tech. Report CSE-92-18, University of California at Davis, 1992.

[18] P. Dokas, L. Etroz, V. Kumar, A. Lazarevic, J. Srivastava, and P. Tan, Data mining for network intrusion detection, Proceeding of NSF Workshop on Next Generation Data Mining (Baltimore, MD), November 2002, pp. 21{30.

[19] P. Domingos, Context-sensitive feature selection for lazy learners, (1997), 227{253.

[20] J. G. Dy and C. E. Brodely, Feature subset selection and order identification for unsupervised learning, ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning (San Francisco, CA, USA), Morgan Kaufmann Publishers Inc., 2000, pp. 247{254.

[21] L. Ertoz,E. Eilertson, A. Lazarevic, P.N. Tan, P. Dokas, V. Kumar, and J. Srivastav, Detection of novel network attacks using data mining, In ICDM Workshop on Data Mining for Computer Security (DMSEC) (Melbourne, FL), Nov. 19 2003, pp. 30{39.

[22] Chapman Flack and Mikhail J. Atallah, Better logging through formality applying formal specification techniques to improve audit logs and log consumes, Proceedings of Recent Advances in Intrusion Detection, 3rd International Symposium, (RAID 2000) (Toulouse, France) (H. Debar, L. M, and S.F. Wu, eds.), Lecture Notes in Computer Science, Springer-Verlag Heidelberg, October 2000, pp. 1{16.

[23] S. Forrest, S. Hofmeyr, A. Somayaji, and T. Longstaff, A sense of self for unix processes, Proceeding of the 1996 IEEE Symposium on Security and Privacy (Los Alamitos, CA), IEEE Computer Society Press, 1996, p. 120128.

[24] J. M. Garibaldi and R. I. Jhon, Choosing membership functions of linguistic terms, Proceeding 2003 IEEE International Conference on Fuzzy System, 2003, pp. 578{583.

[25] Anup K. Ghosh, Christoh Michael, and Michael Schatz, A real-time intrusion detection system based on learning program behavior, Proceeding of Recent Advances in Intrusion Detection, 3rd International Symposium, (RAID 2000)(Toulouse, France) (H. Debar, L. M, and S. F. Wu, eds.), Lecture Notes in Computer Science, Springer-Verlag Heidelberg, October 2000, pp. 93{109.