The Effect of Destination Linked Feature Selection In Real-Time Network Intrusion Detection

Phiwa Mzila
Modeling and Digital Science, Information Security
Council for Science and Industrial Research (CSIR)
Pretoria, South Africa
pmzila@csir.co.za

Eric Dube
Modeling and Digital Science, Information Security
Council for Science and Industrial Research (CSIR)
Pretoria, South Africa
Edube1@csir.co.za

***Abstract*—As internet usage rapidly increases in both private and corporate sectors, the study of network intrusion detection is continuously becoming more relevant and has thus been evolving substantially in recent years. One of the most interesting techniques in the network intrusion detection system (NIDS) is the feature selection technique. The ability of NIDS to accurately identify intrusion from the network traffic relies heavily on feature selection, which describes the pattern of the network packets. The objective of this paper is to eliminate unnecessary features from the dataset, namely destination linked features of the network packet, and train a classification model on the remaining features using a k-Nearest Neighbor (k-NN) classifier. Elimination of the insignificant features leads to a simplified problem and may enhance detection rate, which is itself a problem in network intrusion detection system. Furthermore, removal of specifically the destination linked features will allow the trained model to be capable of identifying the attack/intrusion in real-time before it reaches its destination. To evaluate the accuracy of this method, we compare the results of our model trained without destination linked features to the same model trained with features incorporating destination linked features. The results show a similar detection rate for both trained models, but our model has a distinct advantage in that it treats the entire transaction in real-time.**

Keywords-feature selection; pattern recognition;data mining intrusion detection

I. INTRODUCTION

The internet has become a standard communication tool in the modern world. It plays an essential role in running most successful businesses. However, together with the advantages the internet brings, there is also a substantial disadvantage. It exposes both valuable and confidential information to high risks of intrusion and cyber-attacks. A vast amount of research has been conducted in order to prevent such attacks, and a multitude of systems have been designed or proposed in recent decades. Most intrusion detection systems are based on signatures that are developed by manual coding of expert knowledge. These systems match activity on the system being monitored to known signatures of attack. The major problem with this approach is that these network intrusion detection systems fail to generalize to detect new attacks or attacks without known signatures.

Recently, there has been an increased interest in data mining based approaches to build detection techniques for network intrusion detection systems. These techniques are constructed from models that are trained by both known attacks and normal behavior in order to detect unknown attacks. [1][2][3] describe some of the effective data mining techniques that have been developed recently for detecting intrusions in computer networks. However, successful data mining techniques are not sufficient to create effective network intrusion detection systems (NIDS). Although data mining techniques are successful in evaluating the detection rate of the NIDS, there are still some difficulties involved in the implementation. These difficulties can be grouped into three general categories: accuracy (e.g., detection rate), efficiency, and usability.

Another concern about the NIDS is that it should operate in real-time. Currently even existing so called real-time intrusion detection systems have been modelled with data processed off-line. Any NIDS that has been modelled using a dataset with features such as duration, destination port, totalDestinationBytes, totalDestinationPackets, destination, and even stopTime are in fact not real-time intrusion detection systems. Attacks should be prevented or identified before reaching its destination. An effective NIDS should work in real-time, as intrusions take place, to avoid compromising security.

Elimination of the insignificant and/or meaningless inputs leads to a simplification of the problem, as well as faster and more accurate detection results. Feature selection is therefore an important issue in intrusion detection. Any intrusion detection system has some inherent requirements. Its prime purpose is to detect as many attacks as possible with minimum number of false alarms, i.e., the system must

 ISBN: 978-1-61208-281-3

be accurate in detecting attacks. However, an accurate system that cannot handle large amounts of network traffic and is slow in decision making will not fulfil the purpose of an NIDS. Data mining techniques like pattern recognition, data reduction, data classification, and feature selection techniques thus play an important role in the design of an NIDS.

Feature selection is one of the data preprocessing techniques used before classification in an NIDS [4]. Its purpose is to improve the classification detection accuracy through the removal of irrelevant, noisy and redundant features. Feature selection methods generate a new set of features by selecting only a subset of the original features [5].

There are two main feature selection methods: filter methods [6] and wrapper methods [7]. Filter methods evaluate the relevance of the features depending on the general characteristics of the data, without using any machine learning algorithm to select the new set of features [8]. Frequently used filter methods include Information Gain (IG) [9] and Chi-square [10]. Wrapper methods use the classification performance of a machine learning algorithm as the evaluation criterion to select the set of best features [11]. Wrapper methods include the Particle Swarm Optimization (PSO) algorithm [12] and Genetic Algorithm (GA) [13].

For developing intrusion detection systems, a large amount of traffic data is necessary, which must be collected in advance for analysis by the misuse detection or anomaly detection approaches. Based on the collected network audit trail, misuse detection techniques specify well defined attack signatures and anomaly detection techniques establish acceptable usage profiles to differentiate intrusions and normal activities from a future network traffic data stream. However, there are three major problems in the collected network traffic database: problem of irrelevant and redundant features, problem of uncertainty, and problem of ambiguity.

In this paper, we present a real-time feature selection method for an NIDS which avoids the problem of irrelevant features. The model is trained with a dataset excluding all destination linked features. Hence, the selection of features from the raw dataset constitutes a vital step in the process. The destination and arrival time (duration) of the attack are considered unimportant features in constructing a model that can detect an attack before it arrives at its destination.

For evaluating the detection performance of proposed feature selection method, we compare our results with Soft Computing Paradigms Feature Selection (SCPFS) [14], Correlation Based Feature Selection (CFS) [15] and Fast Correlation-Based Filter (FCBF) [16].

This paper is organized as follows. Section 2 describes the related work. Section 3 presents the methodology which includes the description of the dataset and tools used in this paper followed by a proposed framework in Section 4. We then demonstrate the experimental results in Section 5. Finally, we conclude our work and discuss future recommendations.

II. RELATED WORK

A number of systems aimed at improving network intrusion detection have been developed over the years. In [17], an anomaly network intrusion detection system was proposed using a Particle Swarm Optimization (PSO) feature selection method and Information Entropy Minimization (IEM) discretization with Hidden Naive Bays (HNB) classifier. The effectiveness of the proposed network IDS was evaluated by conducting several experiments on NSL-KDD network intrusion dataset. The results showed that the proposed PSO-Discritize-HNB IDS increases the accuracy and decreases the detection time.

Most of the related work in anomaly detection uses Self-Learning Artificial Neural Networks (ANN) as in HyperView [18]. The system's normal traffic is fed to an ANN, which subsequently learns the pattern of normal traffic. The new traffic, including possible attacks, is then applied to the ANN and the output is used to form the intrusion detection decision. Other systems utilize descriptive statistics by collecting uni-modal statistics from certain system parameters into a profile, after which a distance vector is constructed for the observed traffic and the profile. If the distance is great enough the system raises the alarm. Examples of these systems are NIDES [19], EMERALD [20] and Haystack [21].

A system developed by Girardin [22] used multiple self-organizing maps for intrusion detection. A collection of more specialized maps was used to process network traffic for each layered protocol separately. Girardin suggested that each neural network become a specialist, trained to recognize the normal activity of a single protocol. Another approach that differs from anomaly detection and misuse detection considers human factors to support the exploration of network traffic. Girardin used self-organizing maps to project the network events onto a space appropriate for visualization, and achieved their exploration using a map metaphor

Chen et al [23] used Rough Set Theory (RST) and Support Vector Machines (SVM) to detect intrusions. Initially, RST was used to preprocess the data and reduce the dimensions. Later, the features selected by RST were sent to an SVM model to learn and test. This method proved to be effective and also decreased the space density of data. The SVM is one of the most successful classification algorithms in the data mining area [24].

Statistical techniques usually assume an underlying distribution of data and require the elimination of data instances containing noise. Statistical methods are therefore computationally intensive but can be applied successfully to analyse the data [25]. Statistical methods are widely used to build a behaviour based IDS. The behavior of the system is measured by a number of variables sampled over time such as the resource usage duration, the number of processors, and memory disk resources consumed during that session. The model keeps averages of all the variables and detects whether thresholds are exceeded based on the standard deviation of the variable.

 ISBN: 978-1-61208-281-3

Al-Subaie et al [26] used Hidden Markov Models over Neural Networks in anomaly intrusion detection to classify normal network activity and attacks using a large training dataset. The approach was evaluated by analysing how it affected the classification results. Amor et al [27] designed a real time IDS using Naïve Bayes and Decision Trees and the results showed that the Naive Bayes gives higher detection speed and detection rate than the Decision Trees. Authors in [28] and [29] proposed a hybrid intelligent system using Decision Trees (DT), SVM and Fuzzy SVM for anomaly detection (unknown or new attacks). The results showed that the hybrid DT–SVM approach improved the performance for all the classes when compared to an SVM approach.

Most of these approaches address the feature selection process based mostly in random feature reduction which is not sufficient for intrusion detection in real-time.

III. METHODOLOGY

A. Dataset Description

To evaluate the performance, namely the detection rate and accuracy of the proposed feature selection method of real-time IDS system, we used the Information Security Centre of Excellence (ISCX 2012) dataset [30]. The ISCX 2012 dataset has been created by security researchers at ISCX including Ali Shiravi, Hadi Shiravi, and Mahbod Tavallaee. The dataset was designed to aid research efforts in developing, testing and evaluating algorithms for intrusion detection and anomaly detection. This came about due to the fact that anomaly-based approaches in particular suffer from inaccurate evaluation, comparison, and deployment which originate from the scarcity of adequate datasets. Many such datasets are internal and cannot be shared due to privacy issues, others are heavily anonymized and do not reflect current trends, or they lack certain statistical characteristics. At ISCX, a systematic approach to generate the required datasets was introduced to address this need. The data consists of 17 features shown in the Table I below and the "Tag" value indicates whether the flow is normal or an attack.

TABLE 1. LIST OF FEATURES COLLECTED

appName	sourceTCPFlagsDescription
totalSourceBytes	destinationTCPFlagsDescription
totalDestinationBytes	source
totalDestinationPackets	protocolName
totalSourcePackets	sourcePort
sourcePayloadAsBase64	destination
destinationPayloadAsBase64	destinationPort
direction	startDateTime
Tag	stopDateTime

B. RapidMiner

To implement our method we used RapidMiner [30]. RapidMiner is an international open-source data mining framework. It enables users to model complex knowledge discovery processes as it supports nested operator chains. There are several reasons which made RapidMiner the data mining tool of choice. RapidMiner can function on-line on a given data set, which was necessary for this application. It has many data loading, modeling, preprocessing and visualization methods. This avoids the need for preprocessing the data sets. It also enables visualization of the results. It has an easy to use yet robust graphical user interface that facilitates the modeling of different complex processes. It is also modular, which allows the use of additional functionalities, for example, the distance measures used for the anomaly detection operators. Finally it is easily extensible.

C. k-Nearest Neighbor

Our choice for the classification technique was the k-Nearest Neighbor algorithm since it easy to implement and produces better results. It is based on learning by analogy, that is, by comparing a given test example with training examples that are similar to it. The training examples are described by n attributes. Each example represents a point in an n-dimensional space. When given an unknown example, a k-nearest neighbor algorithm searches the pattern space for the k training examples that are closest to the unknown example. These k training examples are the k "nearest neighbors" of the unknown example. "Closeness" is defined in terms of a distance metric, such as the Euclidean distance. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an example is classified by a majority vote of its neighbors, with the example being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the example is simply assigned to the class of its nearest neighbor. The same method can be used for regression, by simply assigning the label value for the example to be the average of the values of its k nearest neighbors. It can be useful to weight the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. The neighbors are taken from a set of examples for which the correct classification (or, in the case of regression, the value of the label) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. The basic k-Nearest Neighbor algorithm is composed of two steps: Find the k training examples that are closest to the unseen example. Take the most commonly occurring classification for these k examples (or, in the case of regression, take the average of these k label values).

D. Experimental Conditions

There were two phases in the experiment. In the first phase we evaluated the performance of our method (off-line classification), using full feature sets of the network traffic dataset. In the second phase (real-time) only selected features, excluding all destination linked features of the network traffic datasets, were used.

IV. PROPOSED FRAMEWORK

Fig. 1 shows the overall framework of the process involved in the proposed feature selection model for an

 ISBN: 978-1-61208-281-3

NIDS. We adopted the TCM-KNN (Transductive Confidence Machines for K-Nearest Neighbors) [31]

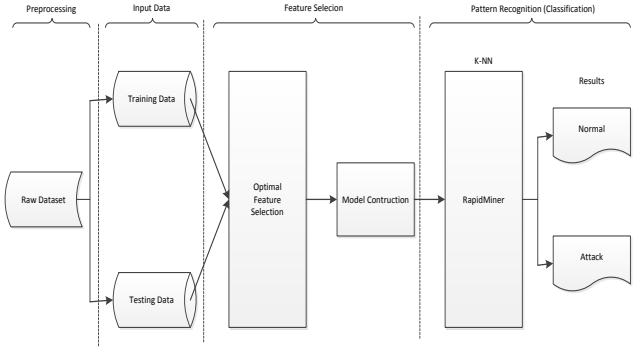

Figure 1. Architecture of the model.

The framework includes four phases: preprocessing, input data, feature selection, and classification.

- Preprocessing: The raw dataset is collected and preprocessed for easy interfacing with RapidMiner format. Each instance of this data is labelled as either an attack or normal.
- Input Data: As input, the preprocessed dataset is split into training and testing datasets. The training dataset is used for training the model with the pattern of both attack and normal network traffic. The testing dataset is unlabelled during the preprocessing phase, and its purpose is to evaluate how well the model is trained for the unknown traffic to detect if it is an attack or normal.
- Feature Selection: Feature selection occurs during model construction. This is the phase where we select features that can construct a model for real-time intrusion detection. All features which are destination linked such as destination port, duration and stop time are deselected in this phase.
- Classification: A well trained model using k-Nearest Neighbour algorithm takes in a testing dataset without label and classifies whether each entry's pattern is closer to those derived from the normal or attack entries of the training dataset.

V. TCM-KNN ALGORITHM

RapidMiner has the capability to compute the confidence using algorithmic randomness theory which was introduced by Transductive Confidence Machines (TCM) [32]. Unlike traditional methods in data mining, transduction can offer measures of reliability to individual points, and uses very broad assumptions except for the main assumption (the training as well as new (unlabelled) points are independently and identically distributed). The calculated p-value serves as a measure of how well the data supports or rejects a null hypothesis (that the point belongs to a certain class). The smaller the p-value, the greater the evidence against the null hypothesis (i.e., the point is an outlier with respect to the current available classes). Users of transduction as a test of confidence have approximated a universal test for randomness (which is in its general form, non-computable) by using a p-value function called strangeness measure [33]. The concept is that the strangeness measure corresponds to the uncertainty of the point being measured with respect to all the other labelled points of a class. Imagine we have an intrusion detection training set $\{(x_1,x_1),..., (x^n, y^n)\}$ of n elements, where $\{X_i = x^1_i, x^2_i, \ldots x^n_i\}$ is the set of feature values (such as the connection duration time, the packet length, etc.) extracted from the raw network packet (or network flow such as TCP flow) for point i, and y_i is the classification for point i, taking values from a finite set of possible classifications (such as normal, attack, intrusion, etc.), which we identify as {1,2,3,..., c} . A test set of s points similar to the ones in the training set is used. The goal is to assign to every test point one of the possible classifications. For every classification confidence measures are also assigned.

VI. EXPERIMENTAL RESULTS

To evaluate the effect of our method of eliminating destination linked features, a dataset of 2000 labelled instances, either as an attack or normal traffic, was used. The k-NN classifier was applied in a cross validation for performance evaluation using RapidMiner operators tool. Cross-validation is a standard statistical method to estimate the generalization error of a predictive model, where each subset of the data is used as a test set with the other subsets forming the training set. We divided the dataset into k=10 equal-sized subsets for k-fold cross-validation. The following procedure was repeated for each subset: Our model was built using the other (k-1) subsets as the training set and its performance was evaluated on the current subset. This means that each subset was used for testing exactly once. The result is the average of the performances obtained from the k=10 rounds.

To determine the detection rate, we used normal evaluation equations with standard measurements, *detection rate* (*DR*) and *false positive rate (FPR*) as described in equation 1 and 2, respectively.

$$DR = \frac{TP}{TP+FN} \quad (1)$$

$$FPR = \frac{FP}{TN+FP} \quad (2)$$

The denotations of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) are defined as follows:

- True Positives (TP): The number of malicious records that are correctly identified.
- True Negatives (TN): The number of legitimate records that are correctly classified.
- False Positives (FP): The number of records that were incorrectly identified as attacks when they are in fact legitimate activities.

 ISBN: 978-1-61208-281-3

- False Negatives (FN): The number of records that were incorrectly classified as legitimate activities when in fact they are malicious.

Tables 2 and 3 below show the performance vector of both the full feature set and reduced feature set (destination linked features removed) respectively. The performance vector comprises of different parameters, which are: accuracy, precision, recall, confusion matrix and class prediction.

TABLE 2.PERFOMANCE VECTOR ON FULL FEATURE SET

PerformanceVector:

Accuracy: 92.05% +/- 2.04% (mikro: 92.05%)
Precision: 82.14% +/- 5.06% (mikro: 81.78%) (Positive class: Attack)
Recall: 93.86% +/- 3.80% (mikro: 93.88%) (Positive class: Attack)

ConfusionMatrix:

True:	Normal	Attack
Normal:	1289	36
Attack:	123	552

Class Prediction: Normal: 97.28%
Class Prediction: Attack: 81.78%

TABLE 3. PERFOMANCE VECTOR ON REDUCED FEATURE SET

PerformanceVector:

Accuracy: 90.70% +/- 1.49% (mikro: 90.70%)
Precision: 79.26% +/- 4.09% (mikro: 78.96%) (positive class: Attack)
Recall: 93.18% +/- 3.99% (mikro: 93.20%) (positive class: Attack)

ConfusionMatrix:

True:	Normal	Attack
Normal:	1266	40
Attack:	146	548

Class Prediction: Normal: 96.94%
Class Prediction: Attack: 78.96%

We compare the DR and FPR rate results of our method with the results of three other methods which are Soft Computing Paradigms Feature Selection (SCPFS), Correlation Based Feature Selection (CFS) and Fast Correlation-Based Filter (FCBF). The results are depicted in Table 4.

The result when using the reduced feature set is lower than the accuracy of the full set of features. The reason for this is the significantly reduced number of features from 17 to 10. Reduction in the number of features is not our primary objective in this work but we do not rule out the possibility that it played a huge role in achieving better results for our method. Our objective is to eliminate precisely all destination linked features of the dataset for training a model that can detect attacks in real-time.

TABLE 4. RESULTS COMPARISON OF DR AND FPR PERFOMED ON K-NN USING FULL FEATURE SET AND REDUCED FEATURE SET

Parameter	Full Set	Proposed Method	SCPFS	CFS	FCBF
DR	70.01	70.00	83.21	12.20	6.87
FPR	0.23	0.22	9.59	0.25	0.13
Accuracy	**92.05**	**90.70**	-	-	-

In Table 4, we have included the accuracy of our method based on the results obtained during our experiments. These results are affected by the significantly decreased number of features participating in the experiment and the size of the dataset used. With a larger dataset the outcome is likely to be different between the full feature set and reduced feature set results. Detection rate for proposed method is lower than SCPFS possibly due to the reduced feature set, but higher than CFS and FCBF. An ideal network intrusion detection system must have lowest false positive rate and the proposed method is lower than SCPFS but higher than CFS and FCBF. There was no significant difference between DR and FPR for the two feature sets, but the main advantage of the proposed method is the real-time application.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a feature selection model for a real-time intrusion detection system. We created a model based on de-selection of destination linked features of the network traffic with the aim of removing features that are irrelevant for a real-time application. The real time intrusion detection system should be able to detect if the traffic instance is an attack before reaching its destination, hence all destination linked features become unnecessary. We then compared our results with the model constructed using a full set of features of the network traffic. The approach using a full feature set of the network traffic represents an off-line intrusion detection scenario while the approach with de-selection of destination linked features represents the real-time intrusion detection scenario. Only a small difference in accuracy were found between the two feature sets, indicating that the proposed reduced feature set did not significantly reduce the accuracy, while still having the main advantage of its suitability for real-time use. Our future work is to develop a real-time network intrusion detection system that is able to detect any intrusion pattern in the network before an attack reaches its target destination using a larger dataset, as well as to improve the accuracy.

REFERENCES

[1] I.T. Jolliffe, “Principal component analysis”, New York Springer-Verlag, 1986.
[2] J. S. Chandrasekaran, B.S. Manjunath, Y.F. Wang, J. Winkeler and H. Zhang, “An eigenspace update algorithm for image

analysis," Graphical Models and Image Processing, 59(5), pp.321-332, 1997.

[3] J. Winkeler, B.S. Manjunath and S. Chandrasekaran, "Subset selection for active object recognition," In CVPR, volume 2, IEEE Computer Society Press, pp.511-516, 1999.

[4] M. Ben-Bassat, Pattern recognition and reduction of dimensionality, Handbook of Statistics II, vol. 1, North-Holland, Amsterdam, pp.773-791, 1982.

[5] H. Liu and H. Motoda, Feature Extraction, Construction and Selection: A Data Mining Perspective, Kluwer Academic, second printing, Boston, 2001.

[6] H. F. Eid and A. Hassanien, "Improved real-rime discretize network intrusion detection model", Seventh International Conference on Bio-Inspired Computing: Theories and Application (BIC-TA 2012), December 14-16, Gwalior, India, 2012.

[7] L. Yu and H. Liu, "Feature selection for high-dimensional data: a fast correlation based filter solution", In Proc. of the Twentieth International Conference on Machine Learning, pp. 856-863, 2003.

[8] Y. Kim, W. Street and F. Menczer, "Feature selection for unsupervised learning via evolutionary search", In Proc. of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 365-369, 2000.

[9] H. Almuallim and T.G. Dietterich, Learning Boolean Concepts in the Presence of Many Irrelevant Features, Artificial Intelligence, vol. 69, pp. 279-305, 1994.

[10] X. Jin, A. Xu, R. Bie and P. Guo, Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles, Lecture Notes in Computer Science, 3916, DOI: 10.1007/1169173011, pp. 106-115, 2006.

[11] Y. Kim, W. Street and F. Menczer, "Feature selection for unsupervised learning via evolutionary search", In Proc. of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 365-369, 2000.

[12] L. Chuang, C. Ke and C. Yang, "A hybrid both filter and wrapper feature selection method for microarray classification", In Proc. of the International Multi Conference of Engineers and Computer Scientists (IMECS), Hong Kong, vol. 1, pp. 19-21, 2008 .

[13] C. Yang, L. Chuang and C. Hong Yang, "IG-GA: A hybrid filter/wrapper method for feature selection of microarray data", Journal of Medical and Biological Engineering, vol. 30, pp. 23-28, 2009.

[14] T. S. Chou, K. K. Yen, and J. Luo, "Network intrusion detection design using feature selection of soft computing paradigms", International Journal of Information and Mathematical Sciences 4:3, 2008.

[15] M. Hall, "Correlation based feature selection for machine learning" Doctoral Dissertation, The University of Waikato, Department of Computer Science, 1999.

[16] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in Proceedings of The Twentieth International Conference on Machine Leaning, Washington, D.C., August, pp. 856-863, 2003.

[17] A. Ahmed Elngar1, A, Dowlat, A. El Mohamed and F. Fayed, "A real-time network intrusion detection system with high accuracy", Information & Computer science Faculty, Sinai University, El-Arish, 2012.

[18] H. Debar, M. Becker, D. Siboni, "A neural network component for an intrusion detection system". Proceedings of the 1992 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, 1992.

[19] P. Porras, P. Neumann, "EMERALD: Event monitoring enabling responses to anomalous live disturbances", Proceedings of the 20th National Information Systems Security Conference, Baltimore, Maryland, 1997.

[20] S. Smaha, "Haystack: An intrusion detection system" Proceedings of the IEEE forth Aerospace Computer Security Applications Conference, Orlando, Florida, 1988.

[21] B. Rhodes, J. Mahaffey, and J. Cannady, "Multiple Self-Organizing Maps for intrusion detection". Proceedings of the NISSC, 2000.

[22] L. Girardin, "An eye on network intruder- administrator shootouts". Proceedings of the Workshop on Intrusion Detection and Network Monitoring, Santa Clara, CA, USA, April 9-12, 1999.

[23] R. Chen, K. Cheng, and C. Hsieh, "Using rough set and support vector machine for network intrusion detect", International Journal of Network Security & Its Applications (IJNSA), 2009.

[24] J.M. Moguerza and Alberto Munoz, "Support vector machines with applications", Statistical Science, vol. 21, No. 3, pp. 322 – 336, 2006.

[25] G. M. Nazer, A. A. L. Selvakumar, "Current intrusion detection techniques in information", European Journal of Scientific Research, EuroJournals Publishing, Inc., pp. 611-624, 2011.

[26] M. Al-Subaie and M. Zulkernine, "Efficacy of Hidden Markov Models over neural networks in anomaly intrusion detection", In 30th Annual International Computer Software and Applications Conference (COMPSAC'06), pp. 325–332, 2006.

[27] B. Amor, S. Benferhat and Z. Elouedi, "Naive Bayes vs. Decision Trees in intrusion detection systems", In SAC '04: Proceedings of the 2004 ACM symposium on applied computing, New York, NY, USA, ACM. ISBN 1-58113-812-1, pp. 420–424, 2004

[28] S. Peddabachigari, A. Abraham, C. Grosanc, and J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems", Journal of Network and Computer Applications, Elsevier Ltd, 2005.

[29] S. Teng, H. Du, N. Wu, W. Zhang, and Jiangyi Su, "A cooperative network intrusion detection based on Fuzzy SVMs", Journal of Networks, vol. 5, pp. 474-483, 2010 .

[30] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection", Computers & Security, vol. 31, Issue 3, ISSN 0167-4048, 10.1016/j.cose.2011.12.012, pp. 357-374, 2012.

[31] Data Mining / Analytic Tools Used Poll". Data Mining / Analytic Tools Used Poll (May 2010). KDnuggets. Retrieved 4 July 2012.

[32] Y. Li, "An effective TCM-KNN scheme for high-speed network anomaly detection", International Journal of Advanced Science and Technology Vol. 24, Chinese Academy of Sciences, Beijing China, 100080, 2010

[33] A. Gammerman, and V. Vovk, "Prediction algorithms and confidence measure based on algorithmic randomness theory". Theoretical Computer Science, pp. 209-217, 2002.

 ISBN: 978-1-61208-281-3