INTRUSION DISCOVERY USING SELF TRAINING SUPPORT VECTOR MACHINE

Naseer Ahmed Shah[1], Jyoti Arora[2]
[1]*M.Tech Student, Desh Bhagat University, Mandi Gobindgarh*
[2]*Assistant Professor, Desh Bhagat University, Mandi Gobindgarh*

***Abstract*—**Intrusion is broadly defined as a successful attack on a network. The definition of attack itself is quite ambiguous and there exists various definitions of it. With the advent of Internet age and the tremendous increase in the computational resources available to an average user, the security risk of each and every computer has grown exponentially. Intrusion Detection System (IDS) is a software tool used to detect unauthorized access to a computer system or network. It is a dynamic monitoring entity that complements the static monitoring abilities of a firewall. Semi-Supervised systems for anomaly detection would reduce the demands of the training process by reducing the requirement of training labeled data. A Self Training Support Vector Machine based detection algorithm is presented in this paper. In the past, Self-Training of SVM has been successfully used for reducing the size of labeled training set in other domains. A similar method was implemented and results of the simulation performed on the KDD Cup 99 dataset for intrusion detection show a reduction of up to 90% in the size of labeled training set required as compared to the supervised learning techniques.

Keywords—SVM;*;IDS; style; styling; insert (key words)*

I. INTRODUCTION

Intrusion is generally defined as a successful attack on a network or system. In a technical report on the practice of Intrusion Detection [1], Julia et. al. have defined attack as \An action conducted by one adversary, the intruder, against another adversary, the victim. The intruder carries out an attack with a specific objective in mind. From the perspective of an administrator responsible for maintaining a system, an attack is a set of one or more events that may have one or more security significances. From the perspective of an intruder, an attack is a mechanism to fulfill an objective."

By its very definition, an intrusion is a subjective phenomenon and its presence or absence can be perceived differently by different observers. An attacker would deem an attack to be successful if he is able to achieve the objectives with which the attack was initiated. From the viewpoint of the victim, an attack is considered successful if it has consequences for him. It is important to note that an attack, though successful from the victim's perspective may still be unsuccessful from the intruder's perspective. For the purpose of detection, usually the victim's perspective is considered. Some common examples of intrusions at the network level would include Denial of Service (DoS) Attack, Packet Sniffing and Remote Login etc. Trojans and spywares are some of the mechanisms by which system level intrusions are achieved.

An intrusion detection system (IDS) is a software tool used to detect unauthorized access to a computer system or network. Ideally an intrusion detection system should be capable of detecting all types of malicious network traffic and computer usage. It is a dynamic entity that complements the static firewall. IDSs have been given the distinction of being dynamic entities by virtue of the fact that they take into account the present state of the system or network and can take actions accordingly. Consider the scenario of a guessing attack on login system. An IDS would be able to recognize the multiple failed attempts in a short span of time and would ag the activity as suspicious. However, the firewall would fail to do so as they are designed to work with a set of pre-configured rules.

Originally intrusion detection systems were tasked with the job of analyzing the network traffic or system activities and raise a ag in case of suspicious events. These systems were not capable of preventing the intrusion. Nowadays e orts are on to develop Intrusion Detection and Prevention Systems (IDPS). Apart from the detection module, these systems have a prevention system as well. The intrusion prevention system is supposed to take necessary actions required to prevent an intrusion detected by the detection system. The advances in the field of social media have significantly contributed to lowering of the skills required for launching a successful attack. In addition to that, the variety and complexity of the systems used today also lead to enhanced and more sophisticated exploits. With our increased dependence on computers and more specifically on the Internet, intrusions present a very serious threat to the three goals of security i.e. confidentiality, integrity and availability. Hence more efficient and accurate intrusion detection systems have become the need of the hour.

II. ARCHITECTURE OF IDS

An Intrusion detection system is considered to have the following components: Data Acquisition Module This module is used in the data collection phase. In the case of a Network Intrusion Detection System (NIDS), the source of the data can be the raw frames from the network or information from upper protocol layers such as the IP or UDP In the case of host based detection system, source of data are the audit logs maintained by the operating system. Feature Generator This module is responsible for extracting a set of selected features from the data acquired by the acquisition module. Features can be classified as low-level and high-level features. A low-level feature can be directly extracted from captured data whereas some deductions are required to be per-formed to extract the high-level features. Considering the example of a network based IDS, the source IP and destination IP of network packets would be the low level features whereas information such as

number of failed login attempts would be classified as high level features. Sometimes features are categorized based on the source of data as well.

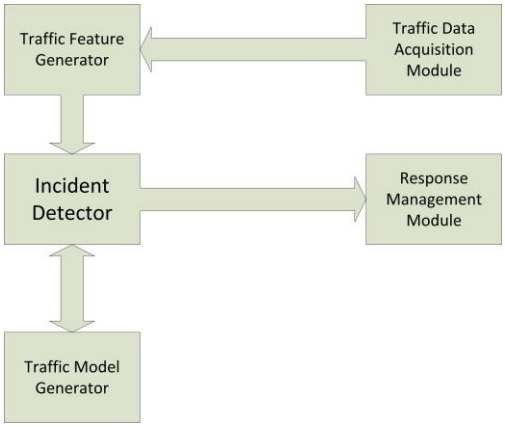

Figure 1.1: Architecture of a Network Intrusion Detection System

Incident Detector This is the core of an IDS. This is the module that processes the data generated by the Feature Generator and identifies intrusions. Intrusion detection methodologies are generally classified as misuse detection and anomaly detection. Misuse detection systems have definitions of attacks and they match the input data against those definitions. Upon a successful match, the activity is classified as intrusion. Anomaly detection systems are based on a definition of normal behaviour of a system. Any deviations from this normal pro le lead to the classification of the corresponding activity as suspicious. Irrespective of the detection methodology, upon detection of an intrusion, an alert is generated and sent to the Response Management module.

Traffic Model Generator This module contains the reference data with which the Incident Detector compares the data acquired by the acquisition modules and processed by the feature generator. The source of data of the Traffic Model Generator could be non-automated(coming from human knowledge) or automated (coming from automated knowledge gathering process).

Response Management Upon receiving an alert from the incident detector, this module initiates actions in response to a possible intrusion. A block diagram of the architecture of a Network Intrusion Detection is presented in g 1.1. The architecture for a Host Based Intrusion Detection System would be similar.

III. LITERATURE SURVEY

Automatic Network Intrusion Detection has been an area of active research for more than the last 20 years. In a survey paper by Catania et. al. [2], the evolution of this field of research and the issues with the existing systems have been discussed. The first Network Intrusion Detection Systems (NIDS) were misuse detection based system like P-BEST and SNORT. However since these systems rely deeply on human activity for traffic model acquisition process, they could not scale with the ever increasing variations of attacks. Data Mining was applied to some misuse based systems to re-duce the demand of human intervention. Various anomaly detection techniques have been applied to this problem domain. Porras and Valdes presented a fairly successful Statistical-Based approach and various Machine Learning techniques have also been applied to this problem. Application of SVM and ANN for intrusion detection was proposed by Chen et. al [3] and Eskin et. al [4] presented an unsupervised technique based on hierarchical clustering. A detailed taxonomy and extensive comparison of various existing methods have been presented in a comprehensive review of Intrusion Detection Systems, Liao et. al. [5].

Apart from the issues related to the requirement of high level of human interaction, other problems with Intrusion Detection Systems have been discussed by Catania et. al. [2]. Lack of model adjustment information, proper traffic feature identify cation, lack of resource consumption information and lack of public network traffic data-sets have been mentioned as some of the important issues. Patcha et. al [6] have given a review of open problems in anomaly detection based IDS. High computation complexity, noise in audit data, high false positive rate, lack of recent standard data-set, inability of IDS to defend itself from attacks, precise definition of normal behavior and inability of IDS to analyze encrypted packets have been cited as the prominent problems with these systems..

In 2011, Horng, Shi-Jinn, et al. proposed an SVM based intrusion detection system, which used hierarchical clustering algorithm, leave one out, and the SVM technique. The hierarchical clustering algorithm provided the SVM with fewer, abstracted, and higher-qualified training instances that are derived from the KDD Cup 1999 training set. It was able to greatly minimize the training time, and improve the performance of SVM. The simple feature selection procedure (leave one out) was applied to eliminate unimportant features from the training set so the obtained SVM model could classify the network traffic data more accurately [1].

In 2012, Gaspar, Paulo, Jaime Carbonell, and José Luís Oliveira et al. gave the review on strategies that are used to improve the classification performance in term of accuracy of SVMs and perform some experimentation to study the influence of features and hyper-parameters in the optimization process, using kernels function. Huang et al provide a study on the joint optimization of C and g parameters (using the RBF kernel), and feature selection using Grid search and genetic algorithms [2].

In 2014, Ahmad, Iftikhar, et al. proposed a genetic algorithm to search the genetic principal components that offers a subset of features with optimal sensitivity and the highest discriminatory power. The support vector machine (SVM) is used for classification. The results show that proposed method enhances SVM performance in intrusion detection [3].

In 2008, Zhou, Jianguo, et al. Proposed system a Culture Particle Swarm Optimization algorithm (CPSO) used to

optimize the parameters of SVM. By using the colony aptitude of particle swarm and the ability of conserving the evolving knowledge of the culture algorithm, this CPSO algorithm constructed the population space based on particle swarm and the knowledge space. The proposed CPSO-SVM model that can choose optimal values of SVM parameters was test on the prediction of financial distress of listed companies in China [5].

In 2011, Kolias, Constantinos, Georgios Kambourakis, and M. Maragoudakis et al. suggested that the RBF has certain parameter that affects the accuracy. PSO is used along with RBF artificial neural network it will improve the accuracy. If it is used in IDS it will improves the accuracy of classification [6].

Furthermore, Heba F. Eid effectively introduced intrusion detection system by using Principal Component Analysis (PCA) with Support Vector Machines (SVMs) as an approach to select the optimum feature subset [25] they verified the effectiveness and the feasibility of the proposed IDS system by several experiments on NSL-KDD dataset.

J.F Joseph, A. Das, B.C. Seet in their paper proposed an autonomous host-based ID for detecting sinking behavior in an ad hoc network [26]. The proposed detection system uses a cross-layer approach to maximize detection accuracy. To further maximize the detection accuracy SVM is used for training the detection model.

However, SVM is computationally expensive for resource-limited ad hoc network nodes. Hence, the proposed IDS preprocess the training data for reducing the computational overhead incurred by SVM. Number of features in the training data is reduced using predefined association functions. Also, the proposed IDS uses a linear classification algorithm, namely Fischer Discriminants Analysis (FDA) to remove data with low-information content (entropy). The above data reduction measures have made SVM feasible in ad hoc network nodes.

T. Shon, Y. Kim, C. Lee and J. Moon in their paper proposed a Machine Learning Model using a modified Support Vector Machine (SVM) that combines the benefits of supervised and unsupervised learning [27]. Moreover, a preliminary feature selection process using GA is provided to select more appropriate packet fields.

Peddabachigari, A. Abraham, C. Grosan conducted an empirical investigation of SVM and Decision Tree, in which they analyzed their performance as standalone detectors and as hybrids [28]. Two hybrids models were examined, a hierarchical model (DT-SVM), with the DT as the first layer to produce node information for the SVM in the second layer, and an ensemble model comprising the standalone techniques and the hierarchal hybrid. For the ensemble approach, each technique is given a weight according to detection rate of each particular attack type during training. Thereafter, when the system is tested, only the technique with the largest weight for the respective attack prediction is chosen to output the classification. The approaches were tested on the KDD Cup ’99 data set.

R. C. Chen, K.F Cheng and C. F Hsieh in their paper used RST (Rough Set Theory) and SVM (Support Vector Machine) to detect intrusions [29]. First, RST is used to preprocess the data and reduce the dimensions. Next, the features selected by RST are sent to SVM model to learn and test respectively. The method is effectively decreased the space density of data.

Kyaw Khaingin in his paper proposed an enhanced SVM Model with a Recursive Feature Elimination (RFE) and kNearest Neighbor (KNN) method to perform a feature ranking and selection task of the new model [30].

Generalization ability of SVM is obviously superior to other traditional learning methods. This basic SVM deals with two-class problems, known as Binary classification problems in which the data are separated by a hyper plane defined by a number of support vectors. Support vectors are a subset of training data used to define the boundary between the two classes. Each instance in the training set contains one “target value" (class labels: Normal or Attack) and 41 features. The goal of SVM is to produce a model which predicts target value of data instance in the testing set which consists of only features. To achieve this goal, we have used Radial Basis Function (RBF) kernel functions [29, 31] available with SVM. In situations where SVM cannot separate two classes, it solves this problem by mapping input data into high-dimensional feature spaces using a kernel function [14, 33].

In high dimensional space it is possible to create a hyper plane that allows linear separation (which corresponds to a curved surface in the lower- dimensional input space). Accordingly, the kernel function plays an important role in SVM. The kernel functions can be used at the time of training of the classifiers which selects support vectors along the surface of this function. SVM classify data by using these support vectors that outline the hyper plane in the feature space. In practice, various kernel functions can be used, such as linear, polynomial or Gaussian. The SVM is already known as the best learning algorithm for binary classification [11] [15] [16].

However, it is not the reason that we have chosen SVM. The most significant reason we chose the SVM is because it can be used for either supervised or unsupervised learning. The SVM, originally a type of pattern classifier based on a statistical learning technique for classification and regression with a variety of kernel functions [7, 19], has been successfully applied to a number of pattern recognition applications [15]. Recently, it has also been applied to inform security for intrusion detection [17] [8].

Another positive aspect of SVM is that it is useful for finding a global minimum of the actual risk using structural risk minimization, since it can generalize well with kernel tricks even in high-dimensional spaces under little training sample conditions. The SVM can select appropriate setup parameters because it does not depend on traditional empirical risk such as neural networks. In the case of supervised SVM learning, it has relatively fast processing and high detection performance when compared to existing artificial neural networks and the unsupervised SVM , as shown in[24][25].

However, one of the main disadvantages of the supervised method is that it requires labelled information for efficient learning. Moreover, it cannot deal with the KDD99 Dataset Pre Processing SVM Train SVM Test Result Analysis relationship

between consecutive variations of learning inputs without additional pre-processing. Therefore, Taeshik Shon and Jongsub Moon have proposed the real time intrusion detection system using Enhanced SVM, which combines soft margin SVM using supervised learning and one-class SVM approach using the unsupervised learning. The enhanced SVM approach inherits the advantages of both SVM approaches, namely high performance and unlabeled capability. The SVM is generally used as a supervised learning method.

Vapnik proposed the initial idea of SVM for the separable case (hard margin SVM) in which the positive and negative samples can be definitely separated by a unique optimal hyper plane with the largest margin. However, this algorithm will find no feasible solution when applied to the non-separable case. Cortes and Vapnik extended this idea to the no separable case (soft margin SVM or the so called standard SVM) by introducing positive slack variables I=1, l .In order to decrease misclassified data, a supervised SVM approach with a slack variable is called soft margin SVM. Additionally, single class learning for classifying outliers can be used as an unsupervised SVM. After considering both SVM learning schemes, an Enhanced SVM approach is proposed.

IV. PROPOSED SYSTEM

Traditionally machine learning has had two types of tasks i.e supervised learning and unsupervised learning[9]. Supervised learning methods require a set of labeled examples, called the training set, over which the algorithm trains by adjusting its parameters. Artificial Neural Networks, K-Means classifiers and Bayesian Belief Networks are some of the examples of supervised learning methods. Unsupervised learning methods attempt to nd the inherent structure in the data, without the use of any previously labeled data. Methods such as the various clustering algorithms and outlier detection algorithms fall under the class of unsupervised learning methods.

Semi-Supervised learning is an amalgamation of the two previously discussed learning methodologies. In this paradigm, training process involves the use of unlabeled data along with some labeled examples. Self-Training, also known as self-learning, self-labeling or decision-directed learning is a wrapper-algorithm that uses a super-vised learning.

Intially it starts labeling the unlabeled points according to the model learned with the help of the intial set of the labeled points. Thereafter a part of the unlabeled points is labeled using the current model and the using the labels of those points, retraining occurs and a new model is learned. This process is reapeated untill the required model accuracy is achieved or the algorithm converges.

Self-Training of SVM has been used in the past for applications such as recognition of Transcription start sites[10], Pixel classification for Remote Sensing Imagery[11] and EEG-based brain computer interface speller system [12].

A similar algorithm is proposed for developing a Self-Training SVM for Intrusion Detection The last trained SVM is considered as thefinal classification model. The proof of convergence of the algorithm is given in Li et. al. [12]

V. RESULTS AND DISCUSSION

The KDD Cup 1999 Dataset[13] was used for the purpose of this simulation.In 1998 MIT Lincoln Labs had prepared a data set under the DARPA Intrusion Detection Evaluation Program[14]. The Third International Knowledge Discovery and Data Mining Tools Contest, which was held along with the The Fifth International Conference on Knowledge Discovery and Data Mining, used a version of the DARPA Intrusion Detection Data Set. The data set, generated from the raw TCP dump data had more than 40 features.

The simulation was run with various sizes of the labeled and unlabeled set, where the maximum ratio between the labeled and unlabeled set was maintained to be 1:10. This ratio was decided on an empirical observation of results obtained by Li et. al. [12].

It was observed that the minimum size of labeled training set required for effective Self-Training was around 500 records. For labeled sets having very few examples, e.g 50-60, the overall accuracy of detection either did not change or in some cases it got reduced from its original value.

This may be explained by considering the fact that in case of limited labeled points in the original case, the decision boundary obtained may not be accurate and upon use of the model on the unlabeled set, the points belonging to the set may be classified incorrectly. This may further lead to a reduction in the overall accuracy of detection.Results obtained for a labeled set of 500 records with an unlabeled set of 5000 records is presented. Results for another simulation with a labeled set of 5000 records and unlabeled set 25000 records are also presented.

It can be inferred from the results that Self-Training process as given in algorithm 1 converges and for the given examples, it converges pretty quickly (after around 6 iterations in both the cases).The degree of improvement in the detection accuracy with the iterations of the Self-Training algorithm depends on the size of the labeled and unlabeled training set. This result can be inferred from the fact that after 6 iterations, the change in the detection accuracy for the simulation with 5000 labeled records set is almost double that of the simulation with 500 labeled records set. This observation is also rea rmed by the fact that for very small labeled training sets, there was virtually no positive improvement in the detection accuracy.

The results also show that the the overall accuracy is most sensitive to the size of the labeled set. In case of the simulation with 500 labeled records, the nal detection accuracy was around 75.5% whereas for the simulation with 5000 labeled records, it was found to be around 86%.

Finally the results validate the hypothesis that Self-Training can be used for reduction of the labeled training set size in the domain of Intrusion Detection as well. A reduction of upto 90% has been achieved in the number of labeled training examples required. A comparison of the performance of Standard SVM and Self-Training SVM has been given in figure 3.4.

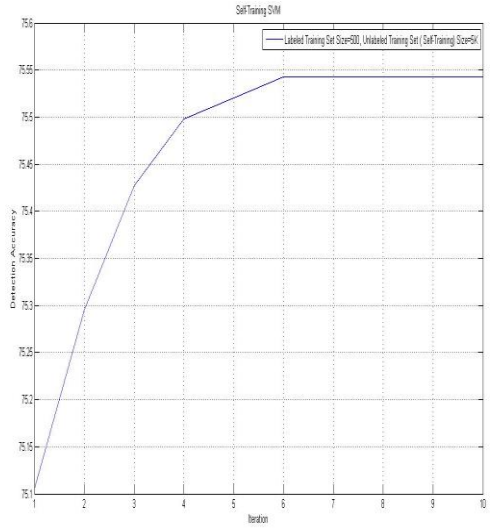

Figure 3.2: Self Training SVM with a Labeled Training Set of Size 500 and Unlabeled Training Set (Self-Training Set) of Size 5K

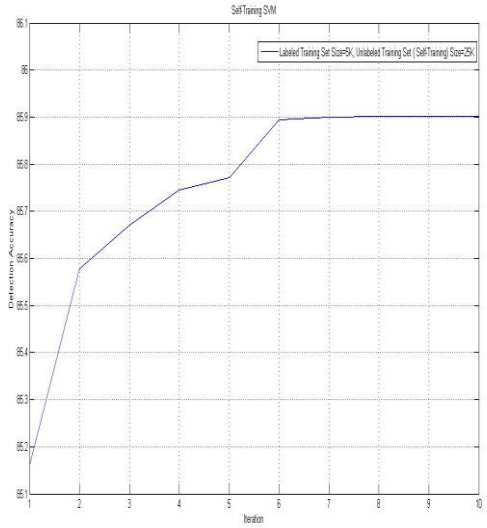

Figure 3.3: Self Training SVM with a Labeled Training Set of Size 5K and Unlabeled Training Set (Self-Training Set) of Size 25 K

VI. CONCLUSION

A new method for Intrusion Detection under the Semi-Supervised Learning paradigm has been presented and evaluated in this thesis. The correctness of the algorithm and its effectiveness for the Intrusion Detection Problem domain has been verified by simulation on the standard KDD Cup 1999 dataset. Further, the given algorithm achieves good results in reduction of requirement of labeled training data. In the simulations run for the purpose of this thesis, a reduction of upto 90% was achieved. This value may vary from case of case, depending upon the compositions of the labeled training set.

The work presented in this thesis may be extended to the case of host based intrusion detection. The performance of this method may also be compared with that of other supervised learning approaches. Additionally the application of Self-Training scheme to other classification techniques used in intrusion detection such as the Bayesian Belief Network can be worked upon.

REFERENCES

[1] Julia Allen, Alan Christie, William Fithen John McHugh, Jed Pickel, and Ed Stoner. State of the practice of intrusion detection technologies. Technical report, Carnegie Mellon University, 2001.

[2] Carlos A. Catania and Carlos Garcia Garino. Automatic network intrusion detection - current techniques and open issues. Computers and Electrical Engineering, 2012.

[3] Wun-Hwa Chen, Sheng-Hsun Hsu, and Hwang-Pin Shen. Application of svm and ann for intrusion detection. Computers and Operations Research, 2005.

[4] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leoniod Portnoy, and Sal Stolfo. A geometric framework for unsupervised anomaly detection detecting intrusions in unlabeled data. Advances in information security, 2002.

[5] Hung-Jen Liao, Kuang-Yuan Tung, Chun-Hung Richard Lin, and Ying-Chih Lin. Intrusion detection system - a comprehensive review. Journal of Network and Computer Applications, 2013.

[6] Animesh Patcha and Jung-Min Park. An overview of anomaly detection techniques- existing solutions and latest technological trends. Computer Networks, 2007.

[7] Pang-Ning Tan, Vipin Kumar, and Michael Steinbach. Introduction to Data Mining. Pearson, 2006.

[8] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. A training algorithm for maximal margin classifiers. In The proceedings of the Fifth Annual Workshop of Computational Learning Theory, pages 144{152. ACM, 1992.

[9] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien, editors. Semi-Supervised Learning, chapter Introduction to Semi-Supervised Learning. MIT Press, 2006.

[10] Jun Cai Huang, Feng Bi Wang, Huan Zhang Mao, and Ming Tian Zhou. A self-training semi-supervised support vector machine method for recognizing transcription start sites. Interna-tional Conference on Apperceiving Computing and Intelligence Analysis (ICACIA), 2010.

[11] Ujjwal Maulik and Debasis Chakraborty. A self-trained ensemble with semisupervised svm - an application to pixel classificationof remote sensing imagery. Pattern Recognition Letters, 2011.

[12] Anderson, J.P," Computer Security Threat Monitoring and Surveillance," Technical Report, Vol.3, pp.234 - 267, 1980.

[13] Yang Li, Li Guo, "An Active Learning Based TCM-KNN Algorithm for Supervised Network Intrusion Detection," Computers & Security, vol.26, pp.459-467, 2007.

[14] Ahmad, Iftikhar, et al. "Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components." Neural Computing and Applications 24.7-8 (2014): 1671-1682.

[15] Hashem, Soukaena Hassan. "Efficiency of Svm and Pca to Enhance Intrusion Detection System." Journal of Asian Scientific Research 3.4 (2013): 381-395.

[16] Zhou, Jianguo, et al. "The study of SVM optimized by Culture Particle Swarm Optimization on predicting financial distress." Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on. IEEE, 2008.

[17] Kolias, Constantinos, Georgios Kambourakis, and M. Maragoudakis. "Swarm intelligence in intrusion detection: A survey." computers & security 30.8 (2011): 625-642.

[18] Lee W and Stolfo S., "Data Mining techniques for intrusion detection", In: Proc. of the 7th USENIX security symposium, San Antonio, TX, 1998.

[19] Dokas P, Ertoz L, Kumar V, Lazarevie A, Srivastava J, and Tan P., "Data Mining for intrusion detection", In: Proc. of NSF workshop on next generation data mining, 2002.

[20] De Boer P., Pels M. "Host-Based Intrusion Detection Systems".Availablehttp://staff.science.uva.nl/~delaat/snb -2004-2005/p19/report.pdf.

[21] Scarfone K., Mell P. "Guide to Intrusion Detection and Prevention Systems". Available at http://csrc.nist.gov/publications/nistpubs/80094/SP80094 .pdf, 2007.

[22] C. Cortes and V. Vapnik, "Support-vector network," Machine Learning, vol. 20, pp. 273–297, 1995

[23] S. Mukkamala, G.1. Janoski, A.H. Sung. Intrusion Detection Using Neural Networks and Support Vector Machines. In Proceedings of IEEE International Joint Conference on Neural Networks, Vol 2, Honolulu, 2002.5, pp. 1702-1707.

[24] V. N. Vapnik. The nature of statistical learning theory. Springer Verlag, New York. NY, 1995

[25] C.J.C. Burges, A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, vol 2(2), Springer US, 1998, pp.121-167.

[26] K.-P. Lin and M.-S. Chen, "Efficient kernel approximation for large-scale support vector machine classification," in Proceedings of the Eleventh SIAM International Conference on Data Mining, 2011, pp. 211– 222

[27] H. Byun, S.W. Lee, A survey on pattern recognition applications of support vector machines, International Journal of Pattern Recognition and Artificial Intelligence 17 (2003) 459–486

[28] Amit Konar, Uday K. Chakraborty, Paul P. Wang, Supervised learning on a fuzzy Petri net, Information Sciences 172 (2005) 397–416

[29] B. Scho ¨lkopf, estimating the support of a high dimensional distribution, Neural Computation 13 (2001) 1443–147.

[30] T. Joachims, Estimating the Generalization Performance of an SVM efficiently, in: Proc. the Seventeenth International Conference on Machine Learning, San Francisco, CA, 2000, pp. 431–438

[31] B.V. Nguyen, An Application of Support Vector Machines to Anomaly Detection, CS681 (Research in Computer Science – Support Vector Machine) report, 2002

[32] S. Dumais, H. Chen, Hierarchical classification of Web content, in: Proc. The 23rd annual international ACM SIGIR conference on Research and development in information retrieval, Athens, Greece, 2000, pp. 256–263

[33] S. Keerthi and C.-J. Lin, "Asymptotic behaviors of support vector machines with Gaussian kernel," Neural Computation, vol. 15, no. 7, pp. 1667–1689, 2003.

[34] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. Journal of Machine Learning

[35] Research, 2:265–292, 2001.

[36] N. Cristiani and J. Shawe-Taylor. An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge, Cambridge University Press, 2000.

[37] Heba F. Eid, Ashraf Darwish, Aboul Ella Hassanien, and Ajith Abraham, (2010) Principle Components Analysis and Support Vector Machine based Intrusion Detection System, IEEE.

[38] J.F Joseph, A. Das,B.C. Seet, (2011) Cross-Layer Detection of Sinking Behavior in Wireless Ad Hoc Networks Using SVM and FDA. IEEE Transaction on dependable and secure computing, Vol. 8, No. 2, MarhApril 2011.

[39] T.Shon, Y. Kim, C.Lee and J.Moon,(2005), A Machine Learning Framework for Network Anomaly Detection using SVM and GA, Proceedings of the 2005 IEEE.

[40] Sandya Peddabachigari, Ajith Abraham, Crina Grosan, Johansson Thomas (2005). Modeling Intrusion Detection Systems using Hybrid Intelligent Systems. Journal of Network and Computer Applications.

[41] R.C. Chen, K.F Cheng and C. F Hsieh (2009), using support vector machine and rough set for network intrusion system.

[42] KyawThetKhaing (2010), Recursive Feature Elimination (RFE) and k-Nearest Neighbor (KNN) in SVM.

[43] Yuanqing Li, Cuntai Guan, Huiqi Li, and Zhengyang Chin. A self-training semi-supervised svm algorithm and its application in an eeg-based brain computer interface speller system. Pattern Recognition Letters, 2008.

[44] Mark Hall, Eibe Frank, Geo rey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. SIGKDD Explorations, 11, 2009. Software available at http://www.cs.waikato.ac.nz/ml/weka.

[45] Chang, Chih-Chung, Lin, and Chih-Jen. LIBSVM: A library for support vector machines. ACM

[46] Transactions on Intelligent Systems and Technology, 2:27:1{27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.