



Intrusion Detection System using Support Vector Machine

Jayshree Jha

Department of Computer Engineering
Ramarao Adik Institute of Technology,
Navi Mumbai, India

Leena Raghya, Ph.D

Department Of Computer Engineering
Ramarao Adik Institute of Technology,
Navi Mumbai, India

ABSTRACT

As the communication industry has connected distant corners of the globe using advances in network technology, intruders or attackers have also increased attacks on networking infrastructure commensurately. System administrators can attempt to prevent such attacks by using intrusion detection tools and systems. In recent years Machine Learning (ML) algorithms has been gaining popularity in Intrusion Detection system (IDS). Support Vector Machines (SVM) has become one of the popular ML algorithm used for intrusion detection due to their good generalization nature and the ability to overcome the curse of dimensionality. As quoted by different researchers number of dimensions still affects the performance of SVM-based IDS. Another issue quoted is that SVM treats every feature of data equally. In real intrusion detection datasets, many features are redundant or less important. It would be better if we consider feature weights during SVM training. This paper presents a study that incorporates Information Gain Ratio (IGR) and K-mean algorithm to SVM for intrusion detection. In purposed framework NSL-KDD dataset is ranked using IGR and later feature subset selection is done using K-mean algorithm.

General Terms

Intrusion Detection System, Support Vector Machine.

Keywords

Support Vector Machines, k-nearest neighbor algorithm, Information Gain Ratio, feature ranking and selection, intrusion detection system.

1. INTRODUCTION

With the advent and increased reach of information technology over the last few years, there have been significant trade-offs among the dividends out of it. Subsequently, there has been an increased focus on the network security under the constant threat of black-hats. Over the last decade, there has been significant increase in network attacks. These attacks have been tremendously complex and severe in nature. Thousands of hackers probe and attack computer networks each day. These attacks range from relatively benign ping sweeps to sophisticated techniques exploiting security vulnerabilities [1]. To defend various cyber attacks and computer viruses, lots of computer security techniques have been studied in last decade, which include cryptography, firewalls and intrusion detection system (IDS) etc.

Among these techniques, intrusion detection has been more promising for defending complex and dynamic intrusion behaviors [2].

Intrusion is defined as any set of activities that attempt to compromise the integrity, confidentiality or availability of a resource. Intrusion detection system (IDS) are security tools

that, like other measures (antivirus software, firewalls etc) are intended to strengthen the security of information and communication system.

In other words, IDS is a device, typically a designated computer system, which monitors activity to identify malicious or suspicious alerts. IDS can be compared with a spam filter, which raises an alarm if specific things occur [3].

A number of Intrusion detection mechanisms have been proposed recently to detect intrusion which can be categorized into statistical methods, knowledge based, data-mining methods and machine learning based methods. In statistical-based techniques, the network traffic activity is captured and a profile representing its stochastic behavior is created. Statistical anomaly detection has no intelligent learning model which may lead to a high rate of false alarms or may not detect attacks reliably. In knowledge based techniques prior knowledge of usage behavior is required. Knowledge based Intrusion detection systems encode an expert's knowledge of known patterns of attack and system vulnerabilities as if-then rules. The acquisition of these rules is a tedious and error-prone process. The various problems with statistical and knowledge based methods, has generated a great deal of interest in the application of machine learning techniques to automate the process of learning the patterns. Several ML algorithms including neural networks, Decision trees, and Bayesian networks etc. has been investigated for the design of IDS by different researchers.. In this paper we investigate the performance of Support Vector Machine for IDS.

The rest of the paper is structured as follows: Section 2 provides background knowledge about ML based algorithm for IDS. Section 3 describes the advantages of using SVM in IDS domain and discusses the limitation of traditional two class SVM. Section 4 reviews the significant works in this field. This section also discusses the extent to which reviewed works overcome the limitations specified in section 3. Section 5 presents the proposed framework which uses hybrid approach for feature selection that combines the filter and wrapper models. In this approach feature is ranked using an independent measure: the information gain ratio. Then k-means classifier's predictive accuracy is used to reach an optimal set of the features which maximize the detection accuracy of the SVM classifier.

2. MACHINE LEARNING

Machine learning, a branch of artificial intelligence, is a scientific discipline concerned with the design and development of algorithms that allow computers to evolve behaviours based on empirical data, such as from sensor data or databases. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data.



ML has a wide range of applications, including search engines, medical diagnosis, text and handwriting recognition, image screening, load forecasting, marketing and sales diagnosis, and so on. In 1994 ML was first utilised for Internet flow classification in the context of intrusion detection (J. Frank, 1994) [4]. It is the starting point for much of the work using ML techniques in Internet traffic classification.

2.1 Machine Learning Algorithms used in IDS

One of the rule-based methods which is commonly used by early IDS is the Expert System (ES) (Bauer & Koblenz, 1988) [5]. In such systems, the knowledge of human experts is encoded into a set of rules. This allows more effective knowledge management than that of a human expert in terms of reproducibility, consistency and completeness in identifying activities that match the defined characteristics of misuse and attacks. However, ES suffers from low flexibility and robustness. Unlike ES, Data Mining approach derives association rules and frequent episodes from available sample data, not from human experts. It utilizes statistical techniques to discover subtle relationships between data items, and from that, constructs predictive models. Using the derived rules, Lee et. al. developed a data mining framework for the propose of intrusion detection (W. Lee, Stolfo, & Mok, 1999) [6]. In particular, system usage behaviours are recorded and analyzed to generate rules which can recognize misuse attacks. The drawback of such frameworks is that they tend to produce a large number of rules and thereby, increase the complexity of the system. Decision Trees are one of the most commonly used supervised learning algorithms in IDS (Amor, Benferhat, & Elouedi, 2004) [7] due to its simplicity, high detection accuracy and fast adaptation. Another highly performing method is Artificial Neural Networks (ANN) which can model both linear and non-linear patterns. The resulting model can generate a probability estimate of whether given data matches the characteristics that it has been trained to recognize. Latter ANN-based IDS (Mukkamala, 2002) [8] have reportedly achieved great successes in detecting difficult attacks. For unsupervised intrusion detection, data clustering methods can be applied (Shah, Undercoffer, & Joshi, 2003 [9]). These methods involve computing a distance between numeric features and therefore they cannot easily deal with symbolic attributes, resulting in inaccuracy. Another well-known ML technique used in IDS is Naïve Bayes classifiers (Amor et al, 2004) [7]. Because Naïve Bayes assumes the conditional independence of data features, which is often not the case for intrusion detection, correlated features may degrade its performance. Beside popular decision trees and ANN, Support Vector Machines (SVMs) are also a good candidate for intrusion detection systems (Ambwani, 2003) [10] which can provide real-time detection capability, deal with large dimensionality of data. SVMs plot the training vectors in high dimensional feature space through nonlinear mapping and labeling each vector by its class. The data is then classified by determining a set of support vectors, which are members of the set of training inputs that outline a hyperplane in the feature space.

3. SUPPORT VECTOR MACHINE

The SVM is already known as the best learning algorithm for binary classification. The SVM, originally a type of pattern classifier based on a statistical learning technique for classification and regression with a variety of kernel

functions, has been successfully applied to a number of pattern recognition applications. Recently, it has also been applied to information security for intrusion detection. Support Vector Machine has become one of the popular techniques for anomaly intrusion detection due to their good generalization nature and the ability to overcome the curse of dimensionality. 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 [13]. One of the main advantage of using SVM for IDS is its speed, as the capability of detecting intrusions in real-time is very important. SVMs can learn a larger set of patterns and be able to scale better, because the classification complexity does not depend on the dimensionality of the feature space. SVMs also have the ability to update the training patterns dynamically whenever there is a new pattern during classification [14].

3.1 Limitation of Support Vector Machine in IDS

SVM is basically supervised machine learning method designed for binary classification. Using SVM in IDS domain has some limitation. SVM being a supervised machine learning method requires labelled information for efficient learning. Pre existing knowledge is required for classification which may not be available all the time [13]. SVM has the intrinsic structural limitation of the binary classifier i.e. it can only handle binary-class classification whereas intrusion detection requires multi-class classification [14]. Although there are some improvements, the number of dimensions still affects the performance of SVM-based classifier [16]. SVM treats every feature of data equally. In real intrusion detection datasets, many features are redundant or less important. It would be better if feature weights during SVM training are considered [16]. Training of SVM is time-consuming for IDS domain and requires large dataset storage. Thus SVM is computationally expensive for resource-limited ad hoc network [12]. Moreover SVM requires the processing of raw features for classification which increases the architecture complexity and decreases the accuracy of detecting intrusion [12].

4. LITERATURE SURVEY:

As basic SVM cannot be used for IDS domain due to previously mentioned shortcomings, various authors have suggested variant in SVM framework to address the mentioned limitation. Some of the related works are mentioned here.

- Heba F. Eid, Ashraf Darwish, Aboul Ella Hassanien, and Ajith Abraham 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 [11]. 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 behaviour in an ad hoc network [12]. 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 [13]. 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 [14]. 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 [15]. 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.
- KyawThetKhaingin his paper proposed an enhanced SVM Model with a Recursive Feature Elimination (RFE) and k-Nearest Neighbor (KNN) method to perform a feature ranking and selection task of the new model [16].

Different techniques have been implemented to tackle the problem of feature selection. Some of them method uses the predictive accuracy of a classifier as a means to evaluate the “goodness” of a feature set, while other uses measures such as information, consistency, or distance measures to compute the relevance of a set of features. These approaches suffer from many drawbacks: the first major drawback is that feeding the classifier with arbitrary features may lead to biased results, and hence, we cannot rely on the classifier’s predictive accuracy as a measure to select feature. A second drawback is that for a set of N features, trying all possible combinations of features (2^N Combinations) to find the best combination to feed the classifier is not a feasible approach.

5. PROPOSED OPTIMIZATION

This research presents a complete framework to select the best set of NSL-KDD dataset features that efficiently characterize normal traffic and distinguish it from abnormal traffic using Support vector machine. This research uses hybrid approach for feature selection that combines the filter and wrapper models. In this approach feature has been ranked using an independent measure: the information gain ratio. The k-means

classifier’s predictive accuracy is used to reach an optimal set of the features which maximize the detection accuracy of the SVM classifier.

The feature selection algorithm starts with an empty set S of the best features, and then, proceeds to add features from the ranked set of features F into S sequentially. After each iteration the “goodness” of the resulting set of features S is measured by the accuracy of the k-means classifier. The selection process stops when the gained classifier’s accuracy is below a certain selected threshold value or in some cases when the accuracy of the current subset is below the accuracy of the previous subset. The proposed algorithm use to select the important feature set from the NSL-KDD dataset. The reduced feature NSL-KDD dataset is then used for training and designing detection model on SVM classifier.

Framework of the proposed model consists of following components:

- NSL-KDD dataset Pre-processing,
- IGR based feature ranking,
- Best feature subset formation,
- Modelling of detection model using SVM.

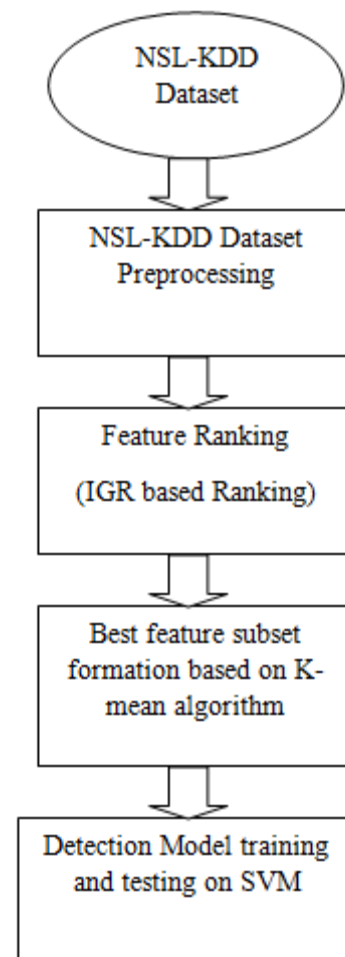


Fig.1. Proposed system architecture

5.1 NSL-KDD Dataset

The dataset to be used in this research is the NSL-KDD dataset [17] which is a new dataset for the evaluation of



researches in network intrusion detection system. It consists of selected records of the complete KDD 99 dataset. NSL-KDD dataset solve the issues of KDD 99 benchmark and connection record contains 41 features. Among the 41 features, 34 features are numeric and 7 features are symbolic or discrete. The NSL-KDD training set contains a total of 22 training attack types; with an additional 17 types in the testing set only. Table I gives the description of NSL-KDD Dataset Features.

Table I: Description of NSL-KDD Dataset Features

S.No.	Name of features	Description
1	Duration	Length (no. of seconds) of the connection
2	protocol_type	Type of the protocol
3	Service	Network service on the destination
4	Flag	Status flag of the connection
5	src_bytes	No. of data bytes from source to destination
6	dst_bytes	No. of data bytes from destination to source
7	Land	1 if connection is from/to the same host/port; 0 otherwise
8	wrong_fragment	No. of wrong fragments
9	Urgent	No. of urgent packets
10	Hot	No. of “hot” indicators
11	num_failed_logins	Logins no. of failed logins
12	logged_in	1 if successfully logged in; 0 otherwise
13	num_compromised	No. of “compromised” conditions
14	root_shell	1 if root shell is obtained; 0 otherwise
15	su_attempted	1 if “su root” command attempted; 0 otherwise
16	num_root	No. of “root” accesses
17	num_file_creations	No. of file creation operations
18	num_shells	No. of shell prompts
19	num_access_files	No. of operations on access control files
20	num_outbound_cmds	No. of outbound commands in an ftp session
21	is_host_login	1 if the login belongs to the “hot” list; 0 otherwise
22	is_guest_login	1 if the login is a “guest” login; 0 otherwise
23	Count	No. of connections to the same host as the current connection in the past two seconds
24	srv_count	No. of connections to the same service as the current connection in the past two seconds
25	error_rate	% of connections that have “SYN” errors

26	srv_error_rate	% of connections that have “SYN” errors
27	rerror_rate	% of connections that have “REJ” errors
28	srv_rerror_rate	% of connections that have “REJ” errors
29	same_srv_rate	% of connections to the same service
30	diff_srv_rate	% of connections to different services
31	srv_diff_host_rate	% of connections to different hosts
32	dst_host_count	Count of connections having the same destination host
33	dst_host_srv_count	Count of connections having the same destination host and using the same service
34	dst_host_same_srv_rate	% of connections having the same destination host and using the same service
35	dst_host_diff_srv_rate	% of different services on the current host
36	dst_host_same_src_port_rate	rate % of connections to the current host having the same src port
37	dst_host_srv_diff_host_rate	% of connections to the same service coming from different hosts
38	dst_host_error_rate	% of connections to the current host that have an S0 error
39	dst_host_srv_error_rate	% of connections to the current host and specified service that have an S0 error
40	dst_host_rerror_rate	% of connections to the current host that have an RST
41	dst_host_srv_rerror_rate	% of connections to the current host and specified service that have an RST error

5.2 NSL – KDD Dataset Preprocessing

SVM classification systems are not able to process NSL - KDD dataset in its current format. Hence preprocessing was required before SVM classification system could be built.

Preprocessing contains the following processes:

- Mapping symbolic features to numeric value.
- Implementing scaling since the data have significantly varying resolution and ranges. The attribute data are scaled to fall within the range [-1, 1].
- Attack names were mapped to one of the two classes, 0 for Normal, 1 for Attack.

5.3 Feature Ranking based on Information Gain Ratio

Information Gain and Information Gain Ratio Measure

Information Gain (IG) is based on information theory using the concept of entropy, which measures the impurity of a data items. The value of entropy is small when the class



distribution is uneven, that is when all the data items belong to one class. The entropy value is higher when the class distribution is more even, that is when the data items have more classes. Information gain is a measure on the utility of each attribute in classifying the data items. It is measured using the entropy value. Information gain measures the decrease of the weighted average impurity (entropy) of the attributes compared with the impurity of the complete set of data items. Therefore, attributes with the largest information gain are considered as the most useful for classifying the data items. The Information Gain Ratio (IGR) is a quantitative measure used to grade the relevance of the features based on the values of such features in the dataset. Information Gain Ratio is used as a measure to determine the relevance of each feature based on measuring the relative entropy reduction. In this research IGR is selected as a measure and not Information Gain because the latter is biased toward the features with a large number of distinct values [18].

IGR is defined in [19] as

$$IGR(Ex,f) = \frac{Gain(Ex,f)}{SplitInfo(Ex,f)}$$

where Ex is the KDD'99 dataset that contains the feature information and the corresponding class.

$$Gain(Ex,f) = Entropy(Ex) - \sum_{v \in Values(f)} \frac{|cEx,v|}{|Ex|} * Entropy(Ex,v),$$

$$Ex,v = \{ x \in Ex / value(x,f) = v \}$$

The entropy function is the Shannon's entropy defined as

$$Entropy(Ex) = - \sum P_i \log_2(P_i),$$

where P_i is the probability of a class i .

SplitInfo(Ex,f) is defined as below

$$Split Info(Ex,f) = - \sum_{v \in values(f)} \frac{|Ex,f|}{|Ex|} \log_2\left(\frac{|Ex,f|}{|Ex|}\right).$$

Table II: Top 31 Features

Rank	Feature no.	Feature	IGR
1	26	srv_serror_rate	0.508
2	25	serror_rate	0.464
3	4	Flag	0.462
4	12	logged_in	0.456
5	39	dst_host_srv_serror_rate	0.442
6	30	diff_srv_rate	0.377
7	38	dst_host_serror_rate	0.374
8	6	dst_bytes	0.292
9	5	src_bytes	0.292
10	29	same_srv_rate	0.276
11	3	Service	0.221
12	11	num_failed_logins	0.22
13	22	is_guest_login	0.212
14	37	dst_host_srv_diff_host_rate	0.2
15	35	dst_host_diff_srv_rate	0.197
16	8	wrong_fragment	0.197
17	14	root_shell	0.197
18	10	Hot	0.171
19	34	dst_host_same_srv_rate	0.165
20	31	srv_diff_host_rate	0.165
21	33	dst_host_srv_count	0.156

22	27	error_rate	0.153
23	23	Count	0.15
24	9	Urgent	0.145
25	2	protocol_type	0.139
26	41	dst_host_srv_error_rate	0.131
27	36	dst_host_same_src_port_rate	0.129
28	32	dst_host_count	0.129
29	17	num_file_creations	0.128
30	18	num_shells	0.123
31	28	srv_error_rate	0.123
32-41		Remaining Features	<0.12

Using NSL-KDD Dataset, we could rank the features according to the score assigned by IGR measure. Top 31 ranked features are shown in Table II.

5.4 The Best Features Subset Selection

The k-means classifier is used to compute the detection rate(accuracy) for each subset of features. Initially, the set of features S contains only the top ranked feature. After each iteration, a new feature is added to the list S based on the rank which it is assigned by the IGR measure one by one as long as the accuracy of selected subset is non-decreasing on features rank in both approaches. When the accuracy drops, as an indication of model overfitting the algorithm is stopped.

Best Feature Selection Algorithm:

Input:

F – Full feature set

IGR: Information Gain Ratio Measure

C: K-means classifier

T: Gained Accuracy Threshold

For each feature f compute IGR(f)

Output:

S – Best feature subset

Algorithm

Initialize: S={}, ac=0

Repeat

- Assign acp= ac
- Evaluate f=getNext(F)
- Calculate S=SU{f}
- Calculate F=F-{f}
- Evaluate ac = accuracy(C,S)
- Continue the above steps until (ac-acp) < T Or ac < acp

Table III: Top ranked features using best feature set selection algorithm

No. of Features	Feature selected for model building	Gained Accuracy
23	26,25,4,12,39,30,38,6,5,29,3,11,2,2,37,35,8,14,10,34,31,33,27,23	99.32
30	26,25,4,12,39,30,38,6,5,29,3,11,2,2,37,35,8,14,10,34,31,33,27,23,9,2,41,36,17,32,18	99.37

Table III shows two examples of features subset for NSL-KDD dataset using best subset selection algorithm. Appropriate feature subset can be selected based upon the Gained Accuracy threshold value. Top ranked features from table II is selected incrementally until Accuracy



Gained drops below the threshold value. This reduced feature set is then given as input to SVM for IDS model building. As the reduction model uses both filter and wrapper approach, the efficiency and intrusion detection capability of SVM model remained unchanged even with reduced feature set.

5.5 Intrusion Detection model building using Support Vector Machines

SVM model is formed which is trained and designed using reduced NSL-KDD dataset using best feature set selection algorithm. Before using SVM classification scaling is needed to be done. This action is to increase the accuracy, decrease the overlap, and reduce complexity. SVM uses a high dimension space to find a hyper-plane to perform binary classification, where the error rate is minimal. SVM is trained using reduced NSL-KDD dataset, finding several support vectors that represent the training data. These support vectors will be formed a model by the SVM, representing a category. According to this model, the SVM will classify a given unknown data as an attack or a normal network data.

6. CONCLUSION

As network attacks have increased in number and severity over the past few years, intrusion detection system (IDS) is increasingly becoming a critical component to secure the network. Due to large volumes of security audit data as well as complex and dynamic properties of intrusion behaviors, optimizing performance of IDS becomes an important open problem that is receiving more and more attention from the research community. Among the variety of intrusion detection approaches, the Support Vector Machine (SVM) is known to be one of the best machine learning algorithms to classify abnormal behaviour. Many Intrusion Detection Systems are based on support vector machine. However, they are computationally very demanding. In order to mitigate this problem, dimension reduction techniques are applied to a given dataset to extract important features.

This research has the following two contributions. First, this paper provides a review on current trends in intrusion detection using SVM together with a study on technologies implemented by some researchers in this research area. Second it proposes a novel approach to select best feature for detecting intrusion. The proposed approach is based on hybrid approach which combines filter and wrapper models for selecting relevant features. This reduced dataset will increase the performance and detection accuracy of SVM based detection model. Moreover the training and testing time will also be reduced with reduced set of feature.

7. REFERENCES

- [1] Jackson, T., Levine, J., Grizzard, J., and Owen, H. (2004). An investigation of a compromised host on a honeynet being used to increase the security of a large enterprise network. In Proceedings of the 2004 IEEE Workshop on Information Assurance and Security.
- [2] D. Dennin, (1987) "An intrusion-detection model", IEEE Transactions on Software Engineering.
- [3] Pfleeger, C. and Pfleeger, S. (2003). Security in computing. Prentice Hall.
- [4] J. Frank, (1994) "Machine learning and intrusion detection: Current and future directions," in Proceedings

of the National 17th Computer Security Conference, Washington, D.C.

- [5] Bauer, D. S., & Koblenz, M. E. (1988). NIDX – an expert system for real-time network intrusion detection.
- [6] Lee, W., Stolfo, S., & Mok, K. (1999). A Data Mining Framework for Building Intrusion Detection Model. Proc. IEEE Symp. Security and Privacy, 120-132.
- [7] Amor, N. B., Benferhat, S., & Elouedi, Z. (2004). Naive Bayes vs. Decision Trees in Intrusion Detection Systems. Proc. ACM Symp. Applied Computing, 420-424.
- [8] Mukkamala, S., Janoski, G., & Sung, A. (2002). Intrusion detection using neural networks and support vector machines. Paper presented at the International Joint Conference on Neural Networks (IJCNN).
- [9] Shah, H., Undercoffer, J., & Joshi, A. (2003). Fuzzy Clustering for Intrusion Detection. Proc. 12th IEEE International Conference Fuzzy Systems (FUZZ-IEEE '03), 2, 1274-1278.
- [10] Ambwani, T. (2003). Multi class support vector machine implementation to intrusion detection. Paper presented at the Proceedings of the International Joint Conference of Neural Networks.
- [11] Heba F. Eid, Ashraf Darwish, Aboul Ella Hassanien, and Ajith Abraham, (2010) Principle Components Analysis and Support Vector Machine based Intrusion Detection System, IEEE.
- [12] 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, Mar-April 2011.
- [13] 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.
- [14] Sandya Peddabachigari, Ajith Abraham, Crina Grosan, Johanson Thomas (2005). Modeling Intrusion Detection Systems using Hybrid Intelligent Systems. Journal of Network and Computer Applications.
- [15] R.C. Chen, K.F. Cheng and C. F. Hsieh (2009), using support vector machine and rough set for network intrusion system.
- [16] Kyaw Thet Khaing (2010), Recursive Feature Elimination (RFE) and k-Nearest Neighbor (KNN) in SVM.
- [17] NSL-KDD Data set for Network-based Intrusion Detection Systems. Available at: <http://nsl.cs.unb.ca/NSL-KDD>.
- [18] H. Liu and H. Motoda (1998), Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic.
- [19] J.R. Quinlan, (1986) "Induction of Decision Trees," Machine Learning, vol. 1, pp. 81-106.

8. ACKNOWLEDGMENTS

Our sincere thanks to all the people who have contributed in anyway in carrying out this work.