Intrusion Detection System using Wrapper Approach

Ajit A Muzumdar
SRESCOE
Kopargaon

Sandip A. Shivarkar
SRESCOE
Kopargaon

Prakash N Kalavadekar
SRESCOE
Kopargaon

ABSTRACT

Intrusion detection is the process of monitoring and analyzing the events occurring in a computer system in order to detect signs of security problems. Today most of the intrusion detection approaches focused on the issues of feature selection or reduction, since some of the features are irrelevant and redundant which results lengthy detection process and degrades the performance of an intrusion detection system (IDS).

The objective of this paper is to construct a lightweight Intrusion Detection System (IDS) aimed at detecting anomalies in networks. The crucial part of building lightweight IDS depends on preprocessing of network data, identifying important features and in the design of efficient learning algorithm that classify normal and anomalous patterns. The design of IDS is investigated from these three perspectives. The goals are to remove redundant instances that causes the learning algorithm to be unbiased, identify suitable subset of features by employing a wrapper based feature selection algorithm and realizing proposed IDS with neurotree to achieve better detection accuracy.

General Terms

Genetic Algorithm, Back Propagation Algorithm, Enhanced C4.5 Algorithm.

Keywords

Sensitivity, Specificity, Error rate.

1. INTRODUCTION

In recent years, due to dramatic growth in networked computer resources, a variety of network-based applications have been developed to provide services in many different areas, e.g., ecommerce services, public web services, and military services. The increase in the number of networked machines has lead to an increase in unauthorized activity, not only from external attackers, but also from internal attackers, such as disgruntled employees and people abusing their privileges for personal gain [11].

Intrusion detection as defined by the SysAdmin, Audit, Networking, and Security (SANS) Institute is the art of detecting inappropriate, inaccurate, or anomalous activity [12]. Today, intrusion detection is one of the high priority and challenging tasks for network administrators and security professionals. More sophisticated security tools mean that the attackers come up with newer and more advanced penetration methods to defeat the installed security systems [13] and [14]. Thus, there is a need to safeguard the networks from known vulnerabilities and at the same time take steps to detect new and unseen, but possible, system abuses by developing more reliable and efficient intrusion detection systems.

This paper is focused on constructing a lightweight Intrusion Detection System (IDS), aimed at detecting anomalies in networks. Intelligent IDS is a dynamic defensive system that is capable of adapting to dynamically changing traffic pattern and is present throughout the network rather than only at its boundaries, thus helping to catch all types of attacks[1]. The trivial factor that complicates such an IDS model construction is the demand for an automatically evolving system. The typical reasons that challenge this process are huge network traffic volumes, highly imbalanced data distribution, the difficulty to realize decision boundaries between normal and abnormal behavior, and a requirement for continuous adaptation to a constantly changing environment of course the toughest factor. The lightweight IDS can be developed by using a wrapper based feature selection algorithm that maximizes the specificity and sensitivity of the IDS as well as by employing a neural ensemble decision tree iterative procedure to evolve optimal features. [1].

Conventional intrusion prevention strategies, such as firewalls, access control schemes or encryption methods, have failed to prove themselves to effusively protect networks and systems from increasingly sophisticated attacks and malwares. The Intrusion Detection Systems (IDS) turn out to be the proper salvage to this issue and have become a crucial component of any security infrastructure to detect these threats before they induce widespread damage [1].

This paper is on developing advanced intelligent systems using ensemble soft computing techniques for intrusion detection. Integration of different soft computing techniques like neural network (NN), genetic algorithm (GA), and decision tree (DT) has lead to discovery of useful knowledge to detect and prevent intrusion on the basis of observed activity.

2. RELATED WORK

Most current approaches to the process of detecting intrusions utilize some form of rule-based analysis. Rule-Based analysis relies on sets of predefined rules that are provided by an administrator, automatically created by the system, or both. Expert systems are the most common form f rule-based intrusion detection approaches. The early intrusion detection research efforts realized the inefficiency of any approach that required a manual review of a system audit trail. While the information necessary to identify attacks was believed to be present within the voluminous audit data, an effective review of the material required the use of an automated system. The use of expert system techniques in intrusion detection mechanisms was a significant milestone in the development of effective and practical detection-based the knowledge of a human expert. These rules are used by the system to make conclusions about the security-related data from the intrusion detection system. Expert systems permit the incorporation of an extensive amount of human experience into a computer application that then utilizes that knowledge to identify activities that match the defined characteristics of misuse and attack [4].

Shun and Malki presented a neural network-based IDS for detecting internet-based attacks on a computer network [1]. Neural networks are used to identify and predict current and future attacks. Feed-forward neural network with the back propagation training algorithm was employed to detect intrusion. Randomly selected data points form KDD Cup (1999) is used to train and test the classifier. The process of learning the behavior of a given program by using evolutionary neural network based on system-call audit data is proposed by Han and Cho [1].

A host based IDS using combinatorial of K-Means clustering and ID3 decision tree learning algorithms for unsupervised classification of abnormal and normal activities in computer network presented [2]. The K-Means clustering algorithm is first applied to the normal training data and it is partitioned into K clusters using Euclidean distance measure. Decision tree is constructed on each cluster using ID3 algorithm. Anomaly scores value from the K-Means clustering algorithm and decisions rules from ID3 are extracted. Resultant anomaly score value is obtained using a special algorithm which combines the output of the two algorithms. The threshold rule is applied for making the decision on the test instance normality. Performance of the combinatorial approach is compared with individual K-Means clustering, ID3 classification algorithm and the other approaches based on Markovian chains and stochastic learning automata. Unlike existing decision tree based IDS discussed above the generated rules fired in this work are more efficient in classification of known and unknown patterns because the proposed neurotree detection paradigm incorporates neural network to pre-process the data in order to increase the generalization ability. But the existing decision tree based approaches discussed in the literature lack generalization and so the ability to classify unseen pattern is reduced [7][8].

Thomas and Balakrishna addressed the problem of optimizing the performance of IDS using fusion of multiple sensors. The trade-off between the detection rate and false alarms highlighted that the performance of the detector is better when the fusion threshold is determined according to the Chebyshev inequality. The major limitation with this approach is it requires large computing power and no experimental results are available for their proposed approach.

Intrusion Detection System using Neural Network based Modelling for detection of anomalous activities. The major contributions of this approach are use and analyses of real network data obtained from an existing critical infrastructure, the development of a specific window based feature mining technique (Fayyad &Uthurusamy, 2002), construction of training dataset using randomly generated intrusion vectors and the use of a combination of two neural network learning algorithms namely the Error Back Propagation and Levenberg–Marquardt, for normal behavior modeling [2].

Koutsoutos, Christou, and Efremidis present a neural network classifier ensemble system using a combination of neural networks which is capable of detecting network attacks on web servers[6]. The system can identify unseen attacks and categorize them. The performance of the neural network in detecting attacks from audit dataset is fair with success rates of more than 78% in detecting novel attacks and suffers from high false alarms rates [6].

Prema Rajeswari and Kannan discusses a rule based approach using enhanced C4.5 algorithm for intrusion detection in order to detect abnormal behaviours of internal attackers through classification and decision making in networks[5]. The enhanced C4.5 algorithm derives a set of classification rules from KDD data set and then the generated rules are used to detect network intrusions in a real-time environment. Unlike existing decision tree based IDS discussed above the generated rules fired in this work are more efficient in classification of known and unknown patterns because the proposed neurotree detection paradigm incorporates neural network to pre-process the data in order to increase the generalization ability [5].

3. PROPOSED SYSTEM

The proposed system has three different phases. The first phase is data pre-processing which removes redundancy from KDD cup dataset 1999.The second phase is feature extraction which extracts features whose fitness function is more than threshold value because KDD cup dataset contains huge records and we are extracting the best features from it using genetic algorithm ,hence the system is light weight intrusion detection system. The third phase is classification of traffic pattern in which feed forward neural network along with enhanced C4.5 algorithm is used for classification. The system overview is as shown in Figure 1.

3.1 Traffic pre-processing

The 1998 DARPA Intrusion Detection Evaluation Program was prepared and managed by MIT Lincoln Labs. The objective was to survey and evaluate research in intrusion detection. A standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, was provided. The 1999 KDD intrusion detection contest uses a version of this dataset.

The major weakness with KDD data set is the presence of redundant records. The occurrence of redundant instances causes the learning algorithm to be biased towards frequent records and unbiased towards infrequent records. As the percentage of records for R2L class is very less in original KDD dataset the learning algorithm is unbiased towards R2L records due to the redundant an enormous records present in class like DoS. These redundant records are removed in order to improve the detection accuracy [2].

3.2 Feature Extraction

There are two common approaches for feature reduction. A Wrapper uses the intended learning algorithm itself to evaluate the usefulness of features, while filter evaluates features according to heuristics based on general characteristics of the data. The wrapper approach is generally considered to produce better feature subsets but runs much more slowly than a filter. Hence genetic algorithm is used to extract the feature.

In order for a GA to efficiently search optimal features from such large spaces, careful attention has to be given to both the encoding chosen and the fitness function [2]. For IDS there is a natural encoding of the space of all possible subsets of a feature set, namely, a fixed-length binary string encoding in which the value of the ith gene {0, 1} indicates whether or not the ith feature (i = 1 to 41) from the overall feature set is included in the specified feature subset. Thus, each individual chromosome in a GA population consists of fixed length, i.e., 41-bit binary string representing some subset of the given feature set.

Each member of the current GA population represents a competing feature subset that must be evaluated to provide fitness feedback to the neurotree. This is achieved by invoking neurotree with the specified feature subset and a set of training data (which is condensed to include only the feature values of the specified feature subset). The neurotree produced is then tested for detection accuracy on a set of unseen evaluation data. To enhance the detection accuracy of the IDS this is indirectly achieved by maximizing the sensitivity and specificity of the classifier. Hence, this

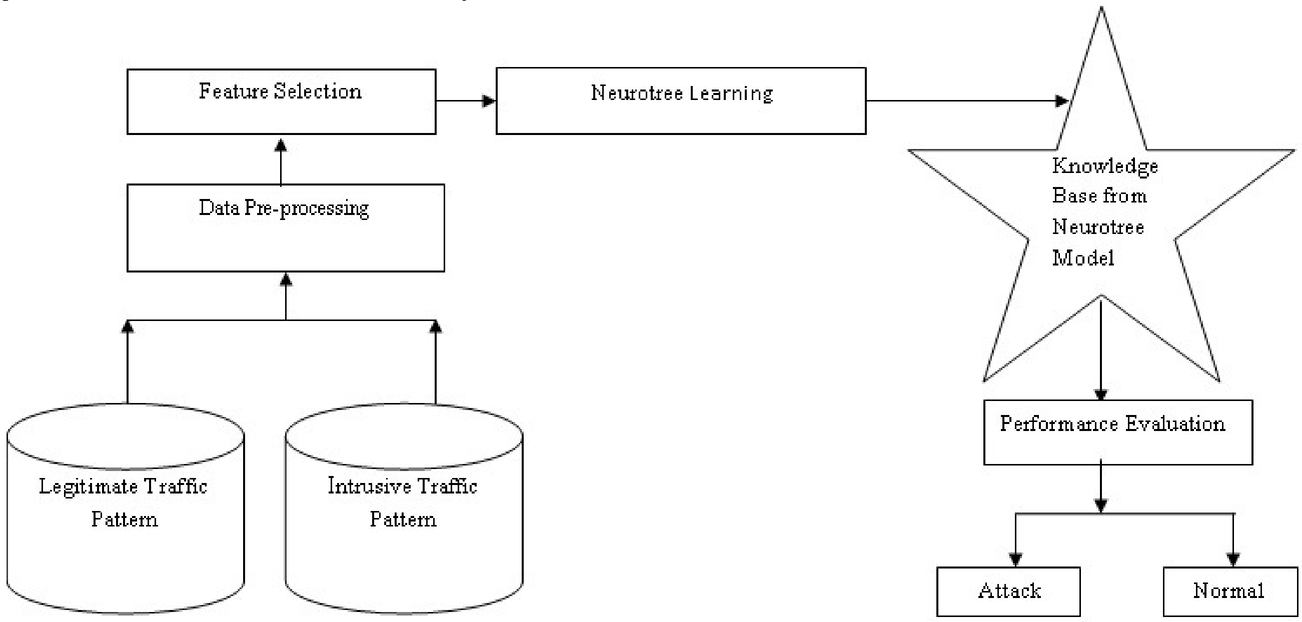

Figure 1: System Overview

knowledge is imparted into the IDS through the fitness function of the GA module. As a result the fitness function is formulated as

Fitness = α*(1/count of one's)+β*sensitivity*specificity... (1)

Sensitivity = ………….(2)

…………...(3)

Where, Specificity =

True positive (TP) - Classifying normal class as normal class.

True negative (TN) - Classifying anomaly class as anomaly class.

False positive (FP) - Classifying normal class as an anomaly class.

False negative (FN) - Classifying anomaly class as a normal class.

Fitness of a chromosome is evaluated based upon the sensitivity and specificity from the validation dataset and number of features present in a chromosome. Here TP and TN are the number of records correctly classified in normal and abnormal classes respectively. Similarly FP and FN are the number of records incorrectly classified in normal and abnormal classes respectively. Count of one's is the number of one's present in the chromosome. If two subsets attain the same performance, while having different number of features, the subset with fewer features have been chosen [9].

3.2.1 Algorithm for feature selection

GA_Feature_Selection()

Input: Encoded binary string of length n (where n is the number of features being passed), number of generations, population size, crossover probability (Pc), mutation probability (Pm).

Output: A set of selected features that maximize the sensitivity and specificity of IDS.

1. Initialize the population randomly with the size of each chromosome to be 41.Each gene value in the chromosome can be '0' or '1'. A bit value of '0' represent the corresponding feature is not present in chromosome and '1' represent the feature is present.

2. Initialize α = 0.2, β = 0.4, γ = 0.4, N (total no. of records in the training set), Pc and Pm.

3. for each chromosome in the new population

a. Apply uniform crossover and mutation operator to the chromosome with the specified probability Pc and Pm.

b. Evaluate Fitness=α*(1/count of one's)+ β*sensitivity *specificity

4. If (Current_fitness-Previous_fitness < 0.0001) then Quit.

5. Select the top best 50% of chromosomes into new population using tournament selection.

6. If number of generations is not reached, go to step 3.

The execution of the combined GA and neurotree algorithm is a wrapper approach. Each iteration results in a decision tree. After n iterations, a series of trees will be obtained, the best of which could be used to generate rules. The tree with the highest sensitivity and specificity is identified to be the best tree.

3.3 Classification of Traffic Patterns

The learning steps of a neurotree detection methodology are as follows. First, a network structure is defined with a fixed number of inputs, hidden nodes and outputs. Second, an algorithm neurotree is chosen to realize the learning process. This algorithm uses bagging approach which generates multiple training datasets from the original KDD dataset and then trains multiple neural networks (back propagation) named neural network ensemble. The predicted results produced by these neural networks are combined based on the voting algorithm. Then, the trained NN ensemble is employed to generate a new training set through replacing the desired class labels of the original training examples, with those output from the trained NN ensemble [10].

Error rate = w1*false positive rate+ w2 * false negative rate

Where w1 and w2 are weight values for false positive and false negative error rate respectively. Best results are obtained when w1 = 0.8 and w2 = 0.2.

3.3.1 Neurotree Algorithm

Procedure NeuralNetwork (TrainingSet Ti)

Begin

1. Get input file Ti for training

2. Read records from Ti

3. Train the network by specifying the number of input nodes, hidden nodes, output nodes, learning rate and momentum.

4. Initialize weights and bias to random values.

5. Calculate output for each node

Node input =P(weight+output of previous layer cells) + Bias value of nodes

Node output = 1/(1 + exp(-(Node input)))

Repeat until final output node is reached

/*Back propagating the errors*/

6. Calculate Error rate (ER) = E (FP, FN)

Therefore,

$$\text{Error\_rate} = w \ast FP + w \ast FN$$

where FP is false positive rate, FN is false negative rate, W1 and W2 are their respective weight values.

7. Output cell error=Logistic function derivative * Error rate

where logistic function derivative df(X)/dy=

1/(1- exp(-x)) _ (1 - (1/(1 - exp(-x)))) * Error rate

8. Hidden Cell error = Logistic function derivative* Sum of (output layer cell error *weight of output layer cell connection)

/*Adjusting weights and bias*/

9. Net weight = Current Weight between hidden layer and output + (output cell error * hidden layer cell value *learning rate)

10. Net Bias value = Current bias Value + (learning rate *output cell error)

11. Training is completed.

12. Return trained neural network.

End.

The information gain measure used in Enhanced C4.5 algorithm is used to select the test attribute at each node in the tree. Such a measure is referred to as an attribute selection measure or a measure of the goodness of split. The attribute with the highest information gain (or greatest entropy reduction) is chosen as the test attribute for the current node. This attribute minimizes the information needed to classify the samples in the resulting partitions. Such an information-theoretic approach minimizes the expected number of tests needed to classify an object and guarantees that a simple (but not necessarily the simplest) tree is found [5].

Split info is the information due to the split of T on the basis of the value of the categorical attribute *D*, which is defined by,

$$\text{Split} = \sum_{'(}^{\&} \frac{!\ "!}{!\ !} + x \log \frac{!\ "!}{!\ !}\% \ldots\ldots\ldots.. (4)$$

And the gain ratio is then calculated by,

$$\text{Gain ratio } (D,T) = \frac{)^{*}\ (,,-)}{0123\_\ \ (,,-)} \ldots\ldots\ldots\ldots (5)$$

The gain ratio, expresses the proportion of useful information generated by split, i.e., that appears helpful for classification. If the split is near trivial, split information will be small and this ratio will be unstable. To avoid this, the gain ratio criterion selects a test to maximize the ratio above, subject to the constraint that the information gain must be large, at least as great as the average gain over all tests examined [5].

4. CONCLUSION

This paper is aimed at making improvements on existing work in three perspectives. Firstly, the input traffic pattern is pre-processed and redundant instances are removed. Next, a wrapper based feature selection algorithm is adapted which has a greater impact on minimizing the computational complexity of the classifier. Finally, a neurotree model is employed as the classification engine which will improve detection rate.

5. REFERENCES

[1] Siva S. Sivatha Sindhu , S. Geetha , A. Kannan" Decision tree based light weight intrusion detection using a wrapper approach", Expert Systems with Applications 39 (2012) 129–141.

[2] Kapil Kumar Gupta, Baikunth Nath, Ramamohanarao Kotagiri," Layered Approach Using Conditional Random Fields for Intrusion Detection" IEEE Transactions On Dependable And Secure Computing, Vol. 7, No. 1, January- March 2010.

[3] Mehdi MORADI and Mohammad ZULKERNINE," A Neural Network Based System for Intrusion Detection and Classification of Attacks", Natural Sciences and Engineering Research Council of Canada (NSERC).

[4]Bertrand Portier ,Froment-Curtil," Data Mining Techniques for Intrusion Detection", The University of Texas at Austin, Spring 2000 Dr. Ghosh – EE380L Data Mining Term Paper.

[5] L Prema RAJESWARI, Kannan ARPUTHARAJ," An Active Rule Approach for Network Intrusion Detection with Enhanced C4.5 Algorithm", I. J. Communications, Network and System Sciences, 2008, 4, 284-359Published Online November 2008 in SciRes .

(http://www.SRPublishing. org/ journal/ijcns/)

[6] Ahmed H. Fares* and Mohamed I. Sharawy ," Intrusion Detection: Supervised Machine Learning", Journal of Computing Science and Engineering, Vol. 5, No. 4, December 2011.

[7] Nahla Ben Amor, Salem Benferhat," Naive Bayes vs Decision Trees in Intrusion Detection Systems" , SAC'04, March 14-17, 2004, Nicosia, Cyprus.

[8] Dr. Saurabh Mukherjeea, Neelam Sharma," Intrusion Detection using Naive Bayes Classifier with Feature Reduction", Procedia Technology 4 (2012) 119 – 128.

[9] Tom V. Mathew," Genetic Algorithm", Indian Institute of Technology Bombay, Mumbai-400076.

[10]Simon Haykin,"Feed Forward Neural networks:an introduction"

[11] Richard Power. 1999 CSI/FBI computer crime and security survey. Computer Security Journal,Volume XV (2), 1999

[12] SANS Institute—Intrusion Detection FAQ, http://www. sans.org/resources/idfaq/, 2010.

[13] Overview of Attack Trends, http://www.cert.org/archive/pdf/attack_trends.pdf, 2002.

[14] K.K. Gupta, B. Nath, R. Kotagiri, and A. Kazi, "Attacking Confidentiality: An Agent Based Approach," Proc. IEEE Int'l Conf. Intelligence and Security Informatics (ISI'06), vol. 3975, pp. 285-296, 2006.