DECISION TREE BASED IDS USING WRAPPER APPROACH

Uttam B. Jadhav[1] and Satyendra Vyas[2]
[1]Department of Computer Engineering, Kota University, Alwar, Rajasthan, India
[2]Department of Computer Engineering, I.E.T. Alwar, Rajasthan, India

ABSTRACT
The objective 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 (i) removing redundant instances that causes the learning algorithm to be unbiased (ii) identifying suitable subset of features by employing a wrapper based feature selection algorithm (iii) realizing proposed IDS with neurotree to achieve better detection accuracy. The lightweight IDS has been 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. An extensive experimental evaluation of the proposed approach with a family of six decision tree classifiers namely Decision Stump, C4.5, Naive Baye's Tree, Random Forest, Random Tree and Representative Tree model to perform the detection of anomalous network pattern has been introduced[1]..

KEYWORDS: *Intrusion Detection, NeuroTree, Wrapper, Perceptron, Kohonen..*

I. INTRODUCTION

The objective 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 (i) removing redundant instances that causes the learning algorithm to be unbiased (ii) identifying suitable subset of features by employing a wrapper based feature selection algorithm (iii) realizing proposed IDS with neurotree to achieve better detection accuracy. 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].
This project 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. Candidate instance subset is generated by removing the redundant and noisy records from the audit log. The GA component imparts the feature subset selection through a suitably framed fitness function. A neurotree paradigm which is a hybridization of neural network and decision tree is proposed for misuse recognition which can classify known and unknown pattern of attacks. The hybridization of different learning and adaptation techniques, overcome individual limitations and achieve synergetic effects for intrusion detection [1].

1.1 Literature Survey of the dissertation

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].

Neural network-based IDS for detecting internet-based attacks on a computer network. 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. The neural networks can work effectively with noisy data, they require large amount of data for training and it is often hard to select the best possible architecture for a neural network [2].

A novel multilevel hierarchical Kohonen net to detect intrusion in network. 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 [1]. The benefit of using evolutionary neural network is that it takes lesser amount of time to obtain better neural networks than when using conventional approaches. This is because they evolve the structures and weights of the neural networks simultaneously. They performed the experiment with the KDD intrusion detection evaluation data. [1] 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 inequality. A neural network supervised learner has been designed to determine the weights of individual IDS depending on their reliability in detecting a certain attack. The final stage of this data dependent fusion architecture is a sensor fusion unit which does the weighted aggregation in order to make an appropriate decision. The major limitation with this approach is it requires large computing power and no experimental results are available for their proposed approach.

An IDS-NNM – Intrusion Detection System using Neural Network based Modeling for detection of anomalous activities [3]. 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, 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. A neural network classifier ensemble system using a combination of neural networks which is capable of detecting network attacks on web servers. 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. An ensemble combining the conventional neural network with a second module that monitors the server's system calls results in good prediction accuracy. Comprehensibility, i.e., the explain-ability of learned knowledge is vital in terms of usage in reliable applications like IDS [2]. The existing NN based IDS discussed in the literature lack comprehensibility and this is incorporated by means of extended C4.5 decision tree. Also a variation in activation function is proposed in order to reduce the error rate thus increasing the detection performance.

An intrusion detection based on the AdaBoost algorithm [2] is proposed in 2008. In this algorithm, decision stumps are used as weak classifiers and decision rules are provided for both categorical and continuous features. They combined the weak classifiers for continuous attributes and categorical attributes into a strong classifier. The main advantage of this approach is that relations between these two different types of features are handled naturally, without any type conversions between continuous and categorical attributes. Additionally they proposed a strategy for avoiding over- fitting to improve the performance of the algorithm.

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 [6].

Different structures of MLP are examined to find a minimal architecture that is reasonably capable of classification of network connection records. The results show that even an MLP with a single layer of hidden neurons can generate satisfactory classification results. Because the generalization capability of the IDS is critically important, the training procedure of the neural networks is carried out using a validation method that increases the generalization capability of the final neural network [4].

Canady [2] used a three layer neural network for offline classification of connection records in normal and misuse classes. The system designed in this study was intended to work as a standalone system (not as a preliminary classifier whose result may be used in a rule-based system). The feature vector used in [2] was composed of nine features all describing the current connection and the commands used in it. A dataset of 10,000 connection records including 1,000 simulated attacks was used. The training set included 30% of the data.

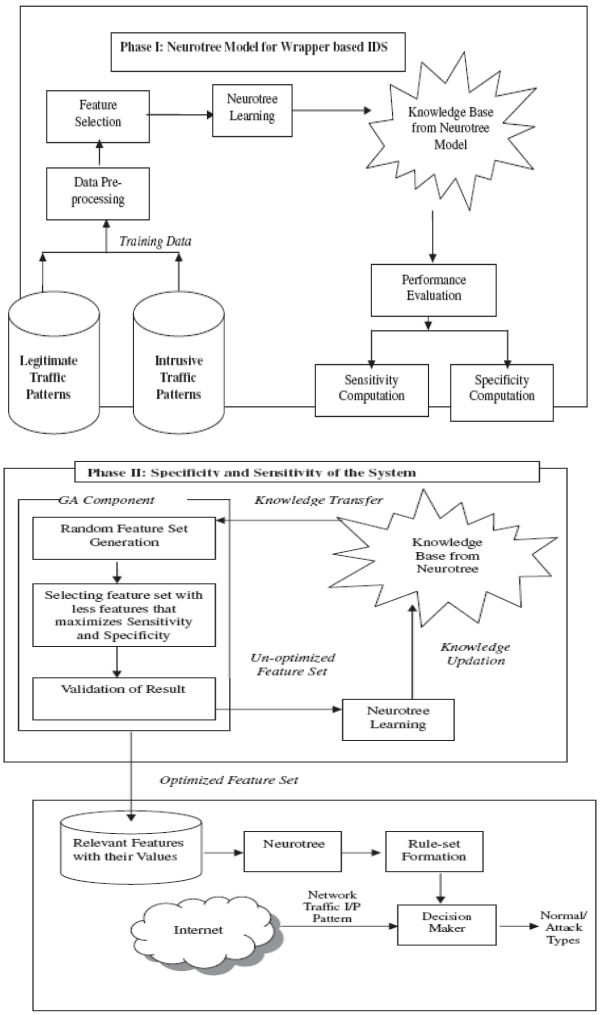
Decision tree [5] is a widely used tool for classification in various domains that need to handle large data sets. One major advantage of the decision tree is its interpretability, i.e., the decision can be represented in terms of a rule set. Each leaf node of the tree represents a class and is interpreted by the path from the root node to the leaf node in terms of a rule such as: "If A1 and A2 and A3, then class C1," where A1, A2, and A3 are the clauses involving the attributes and C1 is the class label. Thus, each class can be described by a set of rules [5].

Classification is similar to clustering in that it also partitions customer records into distinct segments called classes. But unlike clustering, classification analysis requires that the end-user/analyst know ahead of time how classes are defined. It is necessary that each record in the dataset used to build the classifier already have a value for the attribute used to define classes. As each record has a value for the attribute used to define the classes, and because the end-user decides on the attribute to use, classification is much less exploratory than clustering. The objective of a classifier is not to explore the data to discover interesting segments, but to decide how new records should be classified. Classification is used to assign examples to pre-defined categories. Machine learning software performs this task by extracting or learning discrimination rules from examples of correctly classified data. Classification models can be built using a wide variety of algorithms. Classification categorizes the data records in a predetermined set of classes used as attribute to label each record; distinguishing elements belonging to the normal or abnormal class. This technique has been popular to detect individual attacks but has to be applied with complementary fine-tuning techniques to reduce its demonstrated high false positives rate. Classifications algorithms can be classified into three types [5] extensions to linear discrimination (e.g., multilayer perceptron, logistic discrimination), decision tree and rule-based methods (e.g., C4.5, AQ, CART), and density estimators (Naïve ayes, k-nearest neighbor, LVQ).Decision trees are among the well known machine learning techniques. A decision tree is composed of three basic elements: - A decision node is specifying a test attribute. - An edge or a branch corresponding to the one of the possible attribute values which means one of the test attribute outcomes. A leaf which is also named an answer node contains the class to which the object belongs. In decision trees, two major phases should be ensured: 1.Building the tree. 2. Classification. This process will be repeated until a leaf is encountered. The instance is then being classified in the same class as the one characterizing the reached leaf. Several algorithms have been developed in order to ensure the construction of decision trees and its use for the classification task.

A rule based approach using enhanced C4.5 algorithm for intrusion detection in order to detect abnormal behaviors of internal attackers through classification and decision making in networks. 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 [5].

Enhanced C4.5 algorithm is used to develop a more-robust Intrusion Detection System through the use of data-mining techniques. Signature-based intrusion-detection systems are normally known as misuse-detection systems. Misuse detection systems apply a rule-based approach that uses stored signatures of known intrusion instances to detect attacks. The attribute selection measure allowing choosing an attribute that generates partitions where objects are distributed less randomly. In other words, this measure should consider the ability of each attribute to determine training objects' classes. The measure is the gain ratio of Quinlan, based on the Shannon entropy, where for an attribute Ak and a set of objects T. The information gain measure 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].

II. THE SYSTEM OVERVIEW



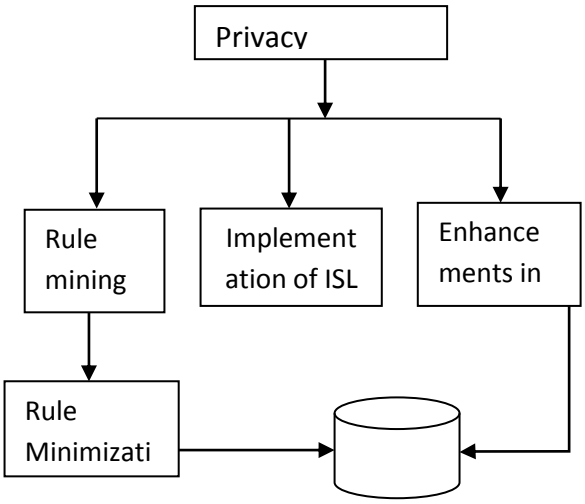


Figure: Block diagram of System Overview

III. IMPLEMENTATION DETAILS

3.1 Module 1: Preprocessing of network traffic pattern

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 and enormous records present in class like DoS. These redundant records are removed in order to improve the detection accuracy.

3.2 Module 2: Feature extraction

In order for a GA (Stein, Chen, Wu, & Hua,2005) to efficiently search optimal features from such large spaces, careful attention has to be given to both the encoding chosen and the fitness function. In this work, 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 i^{th} gene {0, 1} indicates whether or not the i^{th} 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. The advantage of this encoding is that a standard representation and a well understood GA can be used without any modification. 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. We aim to enhance the detection accuracy of the IDS which is indirectly achieved by maximizing the sensitivity and specificity of the classifier. Hence, this knowledge is imparted into the IDS through the fitness function of the GA module

IV. ALGORITHM FOR FEATURE EXTRACTION

The first algorithm hides the sensitive rules according to the first strategy. For each selected rule, it increases the support of the rule's antecedent until the rule confidence decreases below the min_conf threshold. [3]

4.1 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).

4.2 Output

A set of selected features that maximize the
Sensitivity and specificity of IDS. Begin
For each rule r in RH do
{
1. T'_{lr}= {t in D / t partially supports lr}
2. For each transaction of T'_{lr} count the number of items of lr in it.
3. Sort the transactions in T'_{lr} in descending order of the number of items of lr supported.
4. Repeat until Conf(r) < min conf
{
5. Choose the transaction t € T'_{lr} with the highest number of items of lr supported (t is the first transaction in T'_{lr}).
6. Modify t to support lr
7. Increase the support of lr by 1
8. Recomputed the confidence of r
9. Remove t from T'_{lr}
}
10. Remove r from RH
}
End

V. PROPOSED ALGORITHM

5.1 Algorithm DSR

This algorithm decreases the support of the sensitive rules until either their confidence is below the min conf threshold or their support is below the min supp threshold. [3]

5.1.1 Input

A set RH of rules to hide, the source database D, the size of the database |D|, the min conf threshold, the min supp threshold

5.1.2 Output:

The database D transformed so that the rules in RH cannot be mined

Begin
For each rule r in RH do
{
1. Tr = {t in D | t fully supports r}
2. For each transaction of Tr count the number of items in it
3. Sort the transactions in Tr in ascending order of the number of items supported
4. Repeat until (conf (r) < min conf)
{
5. Choose the transaction t in Tr with the lowest number of items
(The first transaction in Tr)
6. Choose the item j in r with the minimum impact on the on the
(|r| - 1)-item sets
7. Delete j from t
8. Decrease the support of r by 1
9. Recomputed the confidance of r
10. Remove t from Tr
}
11. Remove r from RH
}
End

5.2 Neurotree Algorithm

Procedure Neural Network (Training Set 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,

$$Error\_rate = w_1 * FP + w_2 * FN$$

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.

VI. SYSTEM DESIGN

6.1 Breakdown Structure

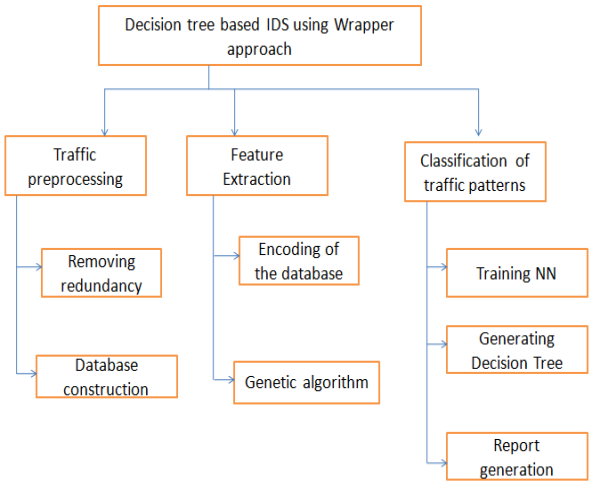


Fig 3.1. Breakdown Structure

6.2 Pre-processing of network traffic pattern

Randomly selected data points form KDD cup dataset is used to train and test the classifier.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].

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. Lincoln Labs set up an environment to acquire nine weeks of raw TCP dump data for a local-area network (LAN) simulating a typical U.S. Air Force LAN. They operated the LAN as if it were a true Air Force environment, but peppered it with multiple attacks.

The raw training data was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic. This was processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records.

A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows to and from a source IP address to a target IP address under some well defined protocol. Each connection is labeled as either normal, or as an attack, with exactly one specific attack type.

VII. ENHANCEMENTS IN EXISTING SYSTEM

Undesired side effects, e.g. non-sensitive rules falsely hidden and spurious rules falsely generated, may be produced in the rule hiding process.

In this module, I will try to develop System that strategically modifies a few transactions in the transaction database to decrease the supports or confidences of sensitive rules without producing the side effects. Since the correlation among rules can make it impossible to achieve this goal.

In this project, I propose heuristic methods for increasing the number of hidden sensitive rules and reducing the number of modified entries.

VIII. CONCLUSION

This project 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.

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them. I am highly indebted to Mr. Satyendra Vyas for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. I would like to express my gratitude towards my parents & member of I.E.T. Alwar for their kind co-operation and encouragement which help me in completion of this project. I would like to express my special gratitude and thanks to industry persons for giving me such attention and time. My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

REFERENCES

[1] Tinghuai Ma, Sainan Wang, ZhongLiu, "*Privacy Preserving Based on Association Rule Mining"* 3rd International Conference on Advanced Computer Theory and Engineering, 2010.

[2] S.-L.Wang and A. Jafari, "*Hiding informative association rule sets"* Expert Systems with Applications 33 (2007) 316-323

[3] J.Han, M. Kamber, "*Data Mining Concepts and Techniques"*, Morgan Kaufmann Publishers, 2001.

[4] V.S.Verykios, A.K. Elmagarmid, E. Bertino,Y. Saygın, and E. Dasseni, *"Association Rule Hiding*" Jan 7, 2003.
[5] Yi-Hung Wu, Chia-Ming Chiang, and Arbee L.P. Chen, Senior Member, IEEE Computer Society, "*Hiding Sensitive Association Rules with Limited Side Effects* ", Jan 2007

BIOGRAPHY

Uttam Babu Jadhav was born in Nasik, India, in 1985. He received the Bachelor in Computer Engineering degree from the University of Pune, Nasik, in 2008 and the Master in Computer Science & Engineering degree from the University of Kota, Alwar appeared both in Computer Science & Engineering. His research interests include Intrusion Detection System.
.