Desegregated ID Execution using Genetic Algorithm

Selvakani S. Kandeeban, R.S. Rajesh

Abstract— **Intrusion Detection systems are increasingly a key part of system defense. Various approaches to Intrusion Detection are currently being used but they are relatively ineffective. Among the several soft computing paradigms, we investigated genetic algorithms and neural networks to model fast and efficient Intrusion Detection Systems. With the feature selection process proposed it is possible to reduce the number of input features significantly which is very important due to the fact that the RBF networks can effectively be prevented from over fitting. The Genetic algorithm employs only the eight most relevant features for each attack category for rule generation. The generated rules signal an attack as well as its category and it is end for training to RBF network. The optimal subset of features combined with the generated rules, can be used to analyze the attacks. Empirical results clearly show that soft computing approach could play a major role for intrusion detection. The model was verified on KDD99 demonstrating higher detection rates than those reported by the state of art while maintaining low false positive rate.**

***Index Terms*— A Genetic algorithms, Information gain, Mutual Information, Radial Basis Function Networks.**

I. INTRODUCTION

Along with bringing revolution in communication and information exchange, Internet has also provided greater opportunity for disruption and sabotage of data that was previously considered secure. While we are benefiting from the convenience that the new technology has brought us, computer systems are facing increased number of security threats that originate externally or internally. As malicious intrusions into computer systems have become a growing problem, the need for accurately detecting these intrusions has risen. Despite numerous technological innovations for information assurance, it is still very difficult to protect computer systems. Therefore intrusion detection is becoming an increasingly important technology that monitors network traffic and identifies, preferably in real time, unauthorized use, misuse and abuse of computer systems.

A number of approaches based on computing have been proposed for detecting network intrusions. The guiding principle of soft computing is exploiting the tolerance of imprecision, uncertainty, partial robustness and low solution cost. Soft computing includes many theories such as Fuzzy logic, Neural Networks, Artificial intelligence, Information and probabilistic reasoning and Genetic Algorithms. When used for intrusion detection, soft computing techniques are often used in conjunction with rule-based expert systems where the knowledge is usually in the form of if-then rules. Despite different soft computing based approaches having been proposed in recent years, the possibilities of using the techniques for intrusion detection are still under utilized.

In our approach we consider Intrusion detection as a data analysis process. Network behaviors can be categorized into normal and abnormal. Due to the sheer volume of real network traffic, both in the amount of records and in the number of features, it is extremely difficult to process all the traffic information before making decisions. We need to extract the most important data that can be used to efficiently detect network attacks. We use information theory to identify the most relevant features to be used [12]. In this work the initial point is the extraction of the most important piece of information that can be deployed for efficient detection of attacks in order to cope with this problem.

Our idea is to achieve high detection rate by introducing high level of generality when deploying the subset of the most important features of the dataset. As this also results in high false positive rate, we deploy additional set of rules in order to recheck the decision of the rule set for detecting attacks. We deploy genetic algorithm (GA) approach for offline training of the rules for classifying different types of intrusions. Genetic Algorithm field is one of the upcoming fields in computer security and has only recently been recognized as having potential in the Intrusion Detection field.

Neural Networks have been actively applied to IDS. To apply neural networks to real world problem successfully, it is very important to determine the number of hidden nodes in the given problem, because performance hinges upon the structure of the neural networks. Hence we use RBF for learning the rules. We examine the proposed method through experiments with real data and compare the results with those of other methods.

The aim here is to develop an Intrusion Detection System which adapts to the environment. Evolution and learning are the two most fundamental process of adaptation. Since learning through neural network is a complex, time consuming process, the connection between learning and evolution can be used to decrease the complexity of the problem.

The rest of the paper is organized as follows: Section 2 presents an overview of related works. Section 3 gives overview on attacks within the KDD data set and proposes the deficiencies of the data set. Section 4 gives overview on Genetic Algorithms and the benefits of its using in intrusion detection field. Section 5 discusses the detection rate of our algorithm when applied to the KDD 99 data.

Manuscript received October 9, 2001.

S. Selvakani is with the Department of Computer Applications, Jaya Engineering College, Chennai, Tamilnadu, India. (Phone: 91-95530956).

R.S. Rajesh, is with Manonmanium Sundaranar University, Department of Computer Science and Engineering, Tirunelveli, Tamilnadu, India.

II. RELATED WORKS

M A. Chittur extended their idea by using GA for anomaly detection [2]. Random numbers were generated using GA. Random numbers were generated using GA. A threshold value was established and any certainty value exceeding this threshold value was classified as a malicious attack. The experimental result showed that GA successfully generated an empirical behavior model from training data. The biggest limitation of this model was the difficulty of establishing the threshold value which might lead to detect novel or unknown attacks.

J. Gomez et al.[3] proposed a linear representation scheme for evolving fuzzy rules using the concept of complete binary tree structure. GA is used to generate genetic operators for producing useful and minimal structure modification to the fuzzy expression tree represented by chromosomes. The biggest drawback of the proposed approach was that the training was time consuming.

Liao and Vemuri used the K-nearest Vector Machine [10] for profiling computer programs. The KNN classifier was employed with an interesting analogy between classifying text documents and detecting intrusion using the sequences of system calls.

Wang et al. used the evolutionary algorithm [13] for discovering neural networks for intrusion detection. The connections of the network and its weights were encoded with binary bits and evolved simultaneously. Their detection system was evaluated with www log data and showed an accuracy rate of 95%. However, in their experiment, they used their own data set rather than a public bench mark dataset.

Hofmann et al. proposed [7] the evolutionary learning of radial basis function networks for intrusion detection. They targeted a network based IDS. Their evolutionary algorithm performed two tasks simultaneously selecting the optimal feature set and learning the RBFN. The binary bits system was used to encode the 137 possible features of the network packet headers and three components of the RBFN, including the type of basis function, the number of hidden neurons, and the number of training epochs. In the experiments with the network audit data set, the RBFN optimized with the evolutionary algorithm outperformed the normal MLP and the normal RBFN.

Gonzalez et al. proposed an intrusion detection technique based on evolutionary generated fuzzy rules [5]. The conditional part of the fuzzy detection rules was encoded with binary bits and fitness was evaluated using two factors: the accuracy and the coverage of the rule. The performance was compared to the methods of different genetic algorithms and without the fuzziness of rules using two network audit datasets their own wireless dataset and the knowledge discovery and data mining cup 99 data set.

III. KDD DATA SET ISSUES AND SOLUTIONS

Learning algorithms have a training phase where they mathematically learn the patterns in the input data set. The input data set is also called the training set which should contain sufficient and representative instances of the patterns being discovered. A data set instance is composed of features.

In order to promote the comparison of advanced research in the area of intrusion detection, the Lincoln Laboratory at MIT, under DARPA sponsorship, conducted the 1998 and 1999 evaluation of intrusion detection [11]. Based on binary TCP dump data provided by DARPA evaluation, millions of connection statistics are collected and generated to form the training and test data in the classifier Learning contest organized in conjunction with the 5th ACM SIGKDD International conference on Knowledge Discovery and Data mining 1999 [11].

The data set contains 5, 000, 000 network connection records. A connection is a sequence of TCP packets starting and ending at some well defined moments of time, between which data flows from source IP to a target IP. The training portion of the dataset "kdd-10-percent" contains 494, 021 connections of which 20% are normal. The testing data set "corrected" provides a data set with a significantly different statistical distribution that the training data set and contains an additional 14 attacks. It contains 311, 029 connections of which 60, 593 are normal.

KDD data set is comprised of records. Each record in the data set consists of 41 features [8] where 38 are numeric and 3 are symbolic defined to characterize individual TCP sessions. Each record is also labeled i.e. the information whether it represents an attack or a normal connection is also provided.

As the first step in our work, in order to cope with the speed problem mentioned above, we have used the results obtained in our previous work [12] where we deployed Information Gain based Mutual Information, in order to extract the most relevant features of the data. In this way the total amount of data to be processed is highly reduced. As an important benefit of this arises the high speed of training the system thus providing high refreshing rate of the rule set.

Subsequently, these features are used to form rules for detecting various types of intrusions. This permits the introduction of higher level of generality and thus higher detection rates. The problem that arises with discarding features is a certain increase of false alarm rate

IV. UR METHODOLOGY

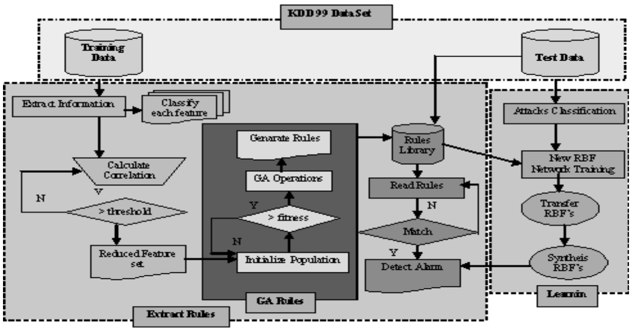


Fig 1 Our Methodology

Our approach shown in fig 1aims at developing an automated approach for building the IDS using feature extraction, Genetic Algorithm and Neural networks. Feature extraction is essential because it is extremely difficult to process in real time the large amount of network traffic to detect network attacks. To process the network data in real time, we need to extract the most important data that can be used to efficiently detect network attacks. Here the information theory concept is used to identify the most relevant features

A. *Feature Selection*

Feature selection is a fundamental problem to select our relevant features and cast away irrelevant and redundant features from an original feature set. If a feature subset satisfies the evaluation measure and has the minimal size, it is regarded as the optimal feature subset. Mutual Information is an important information measure for feature subset. It has been taken as an evaluative measure, where the high valued features are selected and the low valued features are simply discarded. That often reserves redundant features and deletes relevant features.

```
1. Generate feature set R from the ranked list of features
2. For each feature for each type of attack, calculate the
   mutual information between the feature $X_i$ and the
   decision Y, $I(Y;X_i)$
3. Updating relevant features set R by comparing the
   mutual information $I(Y_i;X_i)$
      if $I(Y;X_i) \geq \delta x$ then $R \leftarrow R + \{ X_i \}$
      where $\delta x$ is the threshold which is user defined
4. Create working Set W by copying R
5. Set goal Set G = null
6. While $e(G) < \delta_2$ do
      If W = null then break
      Choose $X_k$ Є W that subjects to
         (i)   Mutual information where
               $I(Y;X_k) \geq I(Y;X_l)$ for all $l \neq k$, $X_l$ Є W
         (ii)  Correlation Measure
               $Q_y(X_k, X_n) \leq Q_y(X_m, X_n)$ for all
                    $m \neq k$, $X_n$ Є G
                    $W \leftarrow W - \{X_k\}$
                    $G \leftarrow G + \{X_k\}$
   End Loop
7. Obtain a feature subset from the intersection of all the
   attacks subset
```

The goal of the feature selection algorithm is to select the minimum set of features that are strongly related to the desired variable and have least redundancy among them. It consists of two functional modules. The first one focuses on removing irrelevance. It depends on a user defined threshold δ, to determine which feature is relevant to the final decision. In this part of the algorithm, irrelevant features are removed from the original feature set. The second part focuses on eliminating redundancy from the features to be selected.

Finally the stopping criterion is based on the evaluation metric e(S), from the user defined threshold δ_2. For each pass, the feature X_k is chosen which satisfies two conditions simultaneously. The first one is that the feature X_k should be the most relevant one compared with the rest of features in the working set. The second one is that feature X_k should have the least correlation with all the features in the working set W.

The main computational part of the algorithm involves computing the mutual information values for $Q_y(X_i, X_j)$ and e(S), which has linear complexity O(N) in terms of the number of instances(N). The complexity of the algorithm that deals with determining the relevant features that is, the algorithm has linear complexity to determine the feature Set from the relevant ones.

TABLE I LIST OF FEATURES WHICH IS MORE RELEVANT

Attack	Relevant Features
Normal	1, 6, 12, 15, 16, 17, 18, 19, 32, 37
Smurf	2, 3, 5, 23, 24, 29, 33, 34, 35, 36, 40, 41
Neptune	4, 23, 24, 25, 26, 29, 33, 34, 35, 38, 39
Land	7

Table I details the most relevant features for some attacks. For majority of the features (31 over 41), normal, smurf, and Neptune are the most discriminative features. There are nine features with very small maximum Information Gain which is smaller than 0.001 which contribute very little to intrusion detection. Moreover features 20 and 21 do not show any variations in the training set therefore they have no relevance to intrusion detection.

B. *GA Approach*

Genetic Algorithms GA are search algorithms based on the principles of natural selection and genetics. GA evolves a population of initial individuals to a population of high quality individuals, where each individual represents a solution to the problem to be solved. Each individual is called chromosome and is composed of predetermined number of genes. The quality of each rule is measured by a fitness function as the quantitative representation of each rule’s adaptation.

The procedure starts from an initial population of randomly generated individuals. Then the population is evolved for a number of generations while gradually improving the qualities of the individuals in the sense of increasing the fitness value as the measure of quality.

During each generation, three basic genetic operators are sequentially applied to each individual i.e. Selection, Cross over and Mutation. Those chromosomes with a higher fitness value are more likely to reproduce offspring. If the new generation contains a solution that produces an output that is close enough or equal to the desired answer then the problem has been solved. If this is not the case, the new generation will go through the same process. This will continue until a solution is reached.

The sample rules are the following ones:

Rule 1:

```
If protocol_type=tcp then
  If service=http then back
 Else if service= private then neptune
  else  if service =finger | telnet then
        if count=1 then land
         if  count >=1&& <=302 then Neptune
```

rule 2:

```
If protocol_type=tcp then
  If service=http then
     If logged_in=1 then
        If dst_host_count =255 then
             If dst_host_same=0 then back
Default: normal
```

C. Knowledge Insertion

After the creation of rules, in order to generalize the profile created neural network methods has been introduced to train the system obey to that fixed IDS rules so the training procedure of the neural network is carried out using a Radial Basis Function network that increases the generalization capability.

Then the symbolic knowledge can be inserted into an RBF network, where no training data exists but a domain expert may be able to formulate IF-THEN rules. The extraction of knowledge in the form of rules has been successfully explored before an RBF networks. The objective converting from rules to RBF networks is to have the knowledge in the consistent format. Hence a full fledged Intrusion Detection System is thus obtained.

One of the main advantages of RBF networks over MLPs is the possibility of setting the free parameters of the hidden units and the possibility of choosing suitable parameters without non-linear optimization. The advantages of RBF models are that the centers can be positioned to reflect domain knowledge and the optimization is fast and accurate.

V. EVALUATION AND RESULTS

We have used an open source machine learning framework WEKA [Waikato Environment for Knowledge Analysis] written at University of Waikato, Newzealand. The algorithms can either be applied directly to a data set or called from our own JAVA code. The input data for weka classifiers is represented in.ARFF [Attribute Relation Function Format], consisting of the list of all instances with the values for each instance separated by commas. As a result of data set training and testing, a confusion matrix will be generated showing the number of instances of each class that has been assigned.

Experiments were conducted to verify the performance of intrusion detection approach based on the above discussion. All the experimental data is available from the corrected data set of KDD cup 1999. Important features based on correlation Measure and Information gain was identified. There were 21 types of intrusions in the test set but only 9 of them were chosen in the framing set. Therefore the selected data also challenged the ability to detect the unknown intrusions.

The main concern of features reduction is one of false alarms and missed intrusion detection. In this work, we attempted to reduce the feature that may be effectively used for intrusion detection without compromising security. We have specially focused on statistical techniques [SPSS] to test individual significance and mutual significance.

In this KDD Data set each sample is unique with 34 numerical features and 7 symbolic features. In the preprocessing task, we map symbolic attributes to numeric valued attributes. Symbolic features like protocol_type(3 different symbols – tcp, udp, icmp), Service(6 different symbols) and flag(11 different symbols) were mapped to integer values ranging from 1 to N where N is the number of symbols.

In the normalization step, we linearly scale each of these features to the range [0.0, 1.0]. Features having smaller integer value ranges like duration [0, 58329], num_compromised [0, 255] were scaled linearly to the range [0.0, 1.0]. Two features spanned over a very large integer range, namely src_bytes [0, 693375640] and dst_bytes [0, 5203179] were scaled by logarithmic scaling to the range [0.0, 20.4] and [0, 15.5]. For Boolean features having values (0 or 1), they were left unchanged.

There are nine features with very small information gain which contribute very little to intrusion detection. Two features do not show any variations in the training set. Finally for each type of attack appropriate reduced feature subset was selected.

It should be noted that the test data is not from the same probability distribution as the training data. Moreover the test data includes a novel attack type that has not been appeared in the training data.

In the second stage, from the reduced feature subset, rules are formed using the genetic algorithm from the KDD data set and tested on the KDD training set to observe their performance with respect to detection, false alarm rate and missed alarm rate. The population of GA is done by encoding the DNA Signature. The only drawback in this is the rules are biased to training data set. The genetic algorithm in the proposed design evaluates the rules and discards the bad rules while generating more rules to reduce the false alarm rate and to increase the intrusion detection. The GA thus continues to detect intrusions and produce new rules, storing the good rules and discard the bad ones.

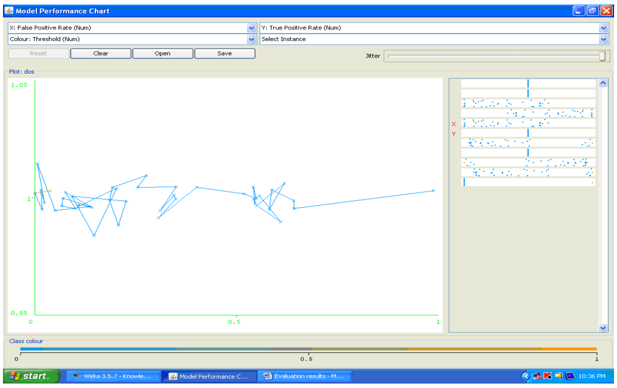
In the third stage, the neural network RBF network focused to train the system by using the training set rules and without actual data it was able to rely on input to output mappings defined through expert knowledge. RBF can model any nonlinear function using a single hidden layer, which eliminates considerations of determining the number of hidden layers and nodes.

The summary of the results after RBF training is given as follows:

```
Correctly Classified Instances      916          99.7821
Incorrectly Classified Instances      2           0.2179 %
Kappa statistic                   0.9959
Mean absolute error               0.0008
Root mean squared error           0.0269
Relative absolute error           0.4262 %
Root relative squared error       9.0025 %
Total Number of Instances         918
```

The detection and false positive rates did not change too much when the threshold changed. The reason is that the Gaussian activation function of RBF assures that the same classes are clustered together and magnifies the output difference if the instances belong to different classes. The original RBF network had an overall tendency to misclassify most examples as “OK” but now the domain rules enabled tighter bounds.

The time complexity is quite low. It requires m*(n^2-n)/2 operations for computing the pair wise feature correlation matrix, where m is the number of instances and n is the initial number of features. The feature selection requires (n^2-n)/w operations for a forward selection and a backward elimination. The hill climbing search is purely exhaustive,



X: False Positive Rate (Num)
Y: True Positive Rate (Num)
Colour: Threshold (Num)
Select Instance

Fig 2: ROC Curve showing Performance

but the use of the stopping criterion makes the probability of exploring the entire search space small.

VI. Conclusion

In this work a combination of GA based ID for detecting different types of attacks was introduced. As we use only nine features to describe the data, its time of training is considerably reduced, thus providing high refreshing rate of the rule set. However the detection rate is not good for some runs because of the selection of cross over and mutation points in corresponding operations is random. The linear structure of the rule makes the detection process efficient in real time processing of the traffic data. The evaluation of our approach showed that the hybrid method of using discrete and continuous features can achieve a better detection rate of network attacks. In order to increase the detection rate, we select the features that are appropriate for each type of network attacks. That is also an added advantage.

The modification of existing RBF networks using heuristic rules has obvious benefits when used in certain situations. The use of knowledge synthesis only makes sense when the available data is insufficient to build a reliable classifier. In such a situation it is advantageous to use heuristic rules to modify an existing RBF network to detect infrequently encountered input vectors that would otherwise be misclassified. However, care must be taken when applying the domain rules.

Further enhancements should be made by the rule learning technique using Radial Basis Network for detecting any unknown attacks.

References

[1] Bouzida.Y and Cuppens.F, "Detecting known an novel network intrusion", "IFIP/SEC 2006, Int. Information Security Conference Karlstad University, Sweeden, PP.123-129, May 2006.

[2] Chittur.A, "Model Generation for an Intrusion Detection System using Genetic Algorithms", High school Honors Thesis, http://www1.cs.columbia.edu/ids/publications/gaids-thesis01.pdf, accessed in 2006.

[3] Gomez.J, Dasgupta.D, Nasaroui.D and Gonzalez.F, "Complete expression Trees for evolving Fuzzy classifiers system with Genetic Algorithms and Applications to Network Intrusion Detection". In proceedings of NAFIPS-FLINT Joint Conference, New Orleans, LA, PP.469-474, June 2002.

[4] Gong.R.H, Zulkernine.M, Anolmaesumi.P, "A software Implementation of a Genetic Algorithm Based approach to Network Intrusion Detection", Proceedings of the SNPD/SAWN' 05, PP.19-27, Aug 2005.

[5] Gonzalez.F, Gomez.J, Kaniganti.M, and Dasgupta.D, "An evolutionary approach to generate fuzzy anomaly signatures", In proceedings 4th Annual IEEE Information Assurance workshop, West point, NY, PP.251-259, June 2003.

[6] Grosan.C, Abraham.A, Chis.M, "Computational Intelligence for light weight Intrusion Detection systems", Int.Conference on Applied computing, IADIS'06, San Sebastian, Spain, PP.538-542, May 2006.

[7] Hofmann.A and Sick.B, "Evolutionary optimization of radial basis function networks for Intrusion Detection", In proceedings of Int.Joint.Conf. Neural Networks, Portland, OR, Vol 1, PP.415-420, July 2003.

[8] KDD cup 1999 data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[9] Lascov.P, Dussel.P, Schafer.C and Riek.K, "Learning Intrusion Detection: Supervised or Unsupervised? ", CIAP: International conference on image analysis and processing No.13, cagliari, Italy, Vol.3617, PP.50-57, Sep 2005.

[10] Liao.Y and Vemuri.V.R, "use of k-nearest neighbour classifier for intrusion detection", Computer Security, Vol.21, No.5, PP.439-448, Oct 2002.

[11] MIT Lincoln Laboratory DARPA Intrusion Detection Evaluation [Online]. Available: http://www.ll.mit.edu/IST/ideval/index.html.

[12] Selvakani.S, Rajesh.R.S, "Improving ID performance using GA and NN", International Journal of Computer Aided Engineering and Technology", Vol.13, N0.1/2/3, 2008.

[13] Wang.L, Yu.G, Wang.G and Wang D, "Method of evolutionary neural network based intrusion detection", In proceeding of Int.Conf.Info-tech and Info-net, Beijing, China, Vol 5, PP.13-18, Oct 2001.

Selvakani S Kandeeban is an Assistant Professor of MCA Department at Jaya Engineering College, Chennai. She received her MCA degree from Manonmanium Sundaranar University and M.Phil degree from Madurai Kamaraj University. She has presented 4 papers in National Conference and 1 paper in international conference. She has published 1 paper in National journal and 7 papers in International Journal. She is currently pursuing her Ph.D degree in Network Security.

Dr. R. S Rajesh received his B.E and M.E degrees in Electronics and Communication Engineering from Madurai Kamaraj University, Madurai, India in the year 1988 and 1989 respectively, and completed his Ph.D in Computer Science and Engineering from Manonmaniam Sundaranar University in the year 2004.