Robust Preprocessing and Random Forests Technique for Network Probe Anomaly Detection

G. Sunil Kumar, C. V. K Sirisha, Kanaka Durga. R, A. Devi

Abstract- During the past few years, huge amount of network attacks have increased the requirement of efficient network intrusion détection techniques. Different classification techniques for identifying various real time network attacks have been proposed in the literature. But most of the algorithms fail to classify the new type of attacks due to lack of collaborative filtering technique and robust classifiers. In this project we propose a new collaborating filtering technique for preprocessing the probe type of attacks and implement a hybrid classifiers based on binary particle swarm optimization (BPSO) and random forests (RF) algorithm for the classification of PROBE attacks in a network. PSO is an optimization method which has a strong global search capability and is used for fine-tuning of the features whereas RF, a highly accurate classifier, is used here for Probe type of attacks classification.

Keywords: Random forest, self organizing map, intrusion detection, filtering, Normalization.

I. INTRODUCTION

With the tremendous growth of network-based services and sensitive information on networks, the number and the severity of network-based computer attacks have significantly increased. Completely preventing breaches of security is unrealistic by security technologies such as information encryption, access control, and intrusion prevention. Thus, Intrusion Detection Systems (IDSs) play a vital role in network Security Network Intrusion Detection Systems (NIDSs) detect attacks by observing various network activities, while Host-based Intrusion Detection Systems (HIDSs) detect intrusions in an individual host. There are two major intrusion detection techniques: misuse detection and anomaly detection. Misuse detection determines intrusions by patterns or signatures which can represent attacks. Thus, misuse based systems can detect known attacks like virus detection systems, but they cannot detect unknown Most of IDS products depend on misuse detection, since misuse detection usually has higher detection rate and lower false positive rate than anomaly detection. Another advantage of misuse detection is high detection speed due to low complexity of detection algorithms. Anomaly detection usually has high computational complexity, especially for unsupervised approaches such as clusters, outlier detection of the random forests algorithm, and Self- Organizing Map (SOM). Therefore, misuse detection is more suitable for on-line detection than anomaly detection. There have been many techniques for modeling anomalous and normal behaviors for intrusion detection. The signature-based and supervised anomaly detections are widely deployed and commercially available. The signature based detection extracts features from the network data. It detects intrusions by comparing the feature values to a set of attack signatures provided by human experts. However, it can only detect previously known intrusions with a Signature. The signature database has to be manually revised for each new type of discovered attacks. On the other hand, the supervised anomaly detection trains models on labeled data (i.e., data pre-classified as an attack or not) and checks how well new data fit into the model. Obviously, it cannot be quickly adapted to new types of intrusion and do not have enough labeled data available. In general, a very large amount of network data needs to be handled and classified. Hence, it is impractical to classify them manually. One of the challenges in IDSs is feature selection. Many algorithms are sensitive to the number of features. Hence, feature selection is essential for improving detection rate. The raw data format of network traffic is not suitable for detection. IDSs must construct features from raw network traffic data, and it involves a lot of computation. Thus, feature selection can help reduce the computational cost for feature construction by reducing the number of features. However, in many current data-mining based IDSs, feature selection is based on domain knowledge or intuition. We use the feature selection algorithm that can give estimates of what features are important in the classification. Another challenge of intrusion detection is imbalanced intrusion. Some intrusions such as denial of service (DoS) [2] have much more connections than others (e.g., user to root). Most of the data mining algorithms try to minimize the overall error rate, but this leads to increasing the error rate of minority intrusions. However, in real-world network environments, minority attacks are more dangerous than majority attacks. In this paper, we improve the detection performance for minority intrusions.

The KDD CUP'99 dataset that is created by MIT Lincoln Lab under contract to Defense Advanced Research Projects Agency (DARPA) is often used to examine the performance of IDS. There are 42 features and millions of connect records in the dataset.

Manuscript received December 24, 2011.

G. Sunil Kumar, Asst.Professor, Dept.of Computer Applications, Maris Stella College, Vijayawada, A.P., India (e-mail:grand1sunil@gmail.com).

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication*

However, high dimensional feature space may include many redundant or noise features which can lead to not only decreasing classification accuracy but also increasing training time and space complexity of classifier. Hence, feature selection is an efficient way to choose the essential feature space which probably can improve the quality of detecting attacks via classification.

Particle Swarm Optimization:

Particle Swarm Optimization is a Random global optimization technology which is based on group of intelligent. For the PSO, the solution of each optimization problems is the location of a bird in the search space, calling these birds as "particles" or "principal". Each particle has its own position and velocity, and there is a fitness value which decision by the fitness function. On the one hand, the particles have self nature. It can judge the flight speed and position by self-experience; On the other hand, it has a social nature, which can adjust the flight velocity and position by the flight of next particle, looking for the balance between personality and social. Particles search in the solution space by memorizing and following the current optimal particle. Each iteration of the process is not completely random, and if found a better solution would be to find the basis for a solution.

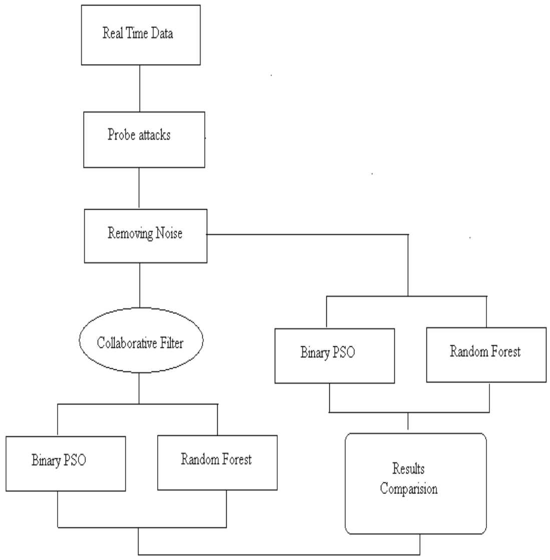
II. RELATED WORK

In a classification problem, the number of features can be quite large, many of which can be irrelevant or redundant. Since the amount of audit data that an IDS needs to examine is very large even for a small network, classification by hand is impossible. Feature reduction and feature selection improves classification by searching for the subset of features, which best classifies the training data. Some of the important features an intrusion detection system should possess include refer in Srilatha et al. [3]. Most intrusion occurs via network using the network protocols to attack their targets. Twycross [4] proposed a new paradigm in immunology, Danger Theory, to be applied in developing an intrusion detection system. Alves et al. [5] presents a classification-rule discovery algorithm integrating artificial immune systems (AIS) and fuzzy systems. For example, during a certain intrusion, a hacker follows fixed steps to achieve his intention, first sets up a connection between a source IP address to a target IP, and sends data to attack the target. Generally, there are four categories of attacks. They are: 1) DoS (denial-of-service), for example ping-of-death, teardrop, smurf, SYN flood, and the like. 2) R2L: unauthorized access from a remote machine, for example guessing password, 3) U2R : unauthorized access to local super user (root) privileges, for example, various "buffer overflow" attacks, 4) PROBING: surveillance and other probing, for example, port-scan, ping-sweep, etc. Some of the attacks (such as DoS, and PROBING) may use hundreds of network packets or connections, while on the other hand attacks like U2R and R2L typically use only one or a few Connections.

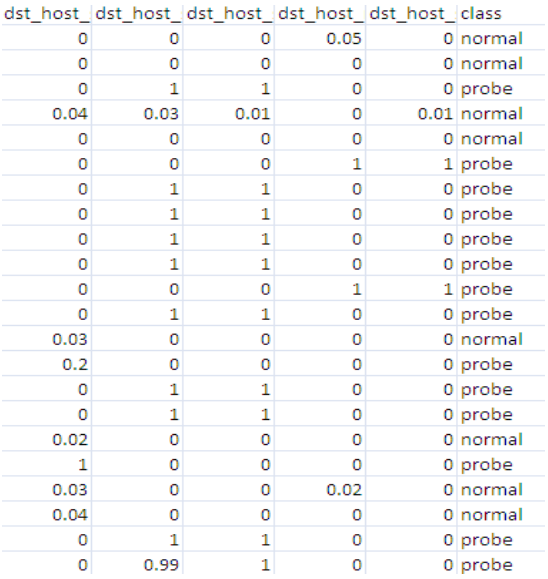
III. OVERVIEW OF THE FRAMEWORK

The proposed framework applies collaborating filters along with robust classifies to detect the intrusions. The framework is shown in Figure . The NIDS captures the network traffic and constructs dataset by pre-processing. After that, the random forest and binary PSO algorithms are used to build the service-based patterns. Proposed approach includes the collaboration filter to the probe attacks after filtering is applied result is applied to both classifiers and then results are compared with existing approach results. The system will effectively classifies the probe type of attacks.

Dataset and Preprocessing

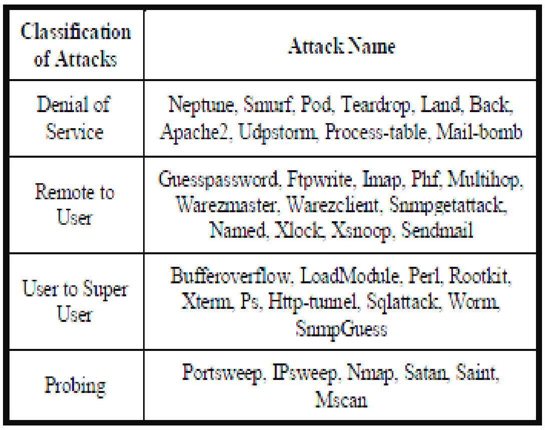


The DARPA dataset is commonly to test most of IDSs. The KDD’99 dataset is a subset of the DARPA dataset prepared by Sal Stofo and Wenke Lee. This dataset is a preprocessed dataset consisting of 41 features (e.g., protocol type, service, and flag) extracted from the tcp dump data in the 1998 DARPA dataset. This dataset can be used without further time-consuming preprocessing and different IDSs can be compared with each other by using the same dataset.A complete KDD ’99 dataset containing 4,898,431 connections with attacks is used for experimentation.



dst_host_	dst_host_	dst_host_	dst_host_	dst_host_	class
0	0	0	0.05	0	normal
0	0	0	0	0	normal
0	1	1	0	0	probe
0.04	0.03	0.01	0	0.01	normal
0	0	0	0	0	normal
0	0	0	1	1	probe
0	1	1	0	0	probe
0	1	1	0	0	probe
0	1	1	0	0	probe
0	1	1	0	0	probe
0	0	0	1	1	probe
0	1	1	0	0	probe
0.03	0	0	0	0	normal
0.2	0	0	0	0	probe
0	1	1	0	0	probe
0	1	1	0	0	probe
0.02	0	0	0	0	normal
1	0	0	0	0	probe
0.03	0	0	0.02	0	normal
0.04	0	0	0	0	normal
0	1	1	0	0	probe
0	0.99	1	0	0	probe

ATTACKS IN KDD99'S DATABASE



Classification of Attacks	Attack Name
Denial of Service	Neptune, Smurf, Pod, Teardrop, Land, Back, Apache2, Udpstorm, Process-table, Mail-bomb
Remote to User	Guesspassword, Ftpwrite, Imap, Phf, Multihop, Warezmaster, Warezclient, Snmpgetattack, Named, Xlock, Xsnoop, Sendmail
User to Super User	Bufferoverflow, LoadModule, Perl, Rootkit, Xterm, Ps, Http-tunnel, Sqlattack, Worm, SnmpGuess
Probing	Portsweep, IPsweep, Nmap, Satan, Saint, Mscan

IV. PROPOSED ALGORITHMS

The traditional filtering algorithms are not adaptive to the situations when kdd99 dataset is large. This may result in the false recommendations. In this paper, a new proposed collaborating filter is used in order to get active probe attacks dynamically according to the network feature changing by using the dynamic similarity. It can detect the target probe type of attacks based on the network remaining features. The procedure for the proposed recommendation is as follows.

Step1. Calculating the weight W(u,i) This step is used to get the best attributes for probe type of attacks detection using W(u,i)=best feature selection method(data)

Step2. Using approved Pearson’s correlation to calculate the similarity Pearson’s correlation, as following, measures the linear correlation between two vectors of ratings.

$$sim(i,j)=\frac{\sum_{c\in I_{i,j}}(R_{i,c}-A_i)(R_{j,c}-A_j)}{\sqrt{\sum_{c\in I_{i,j}}(R_{i,c}-A_i)^2*\sum_{c\in I_{i,j}}(R_{j,c}-A_j)^2}}$$

Where Ri,c is the rating of the probe type of attack c by network protocol i, Ai is the average rating of network protocol i for all the co-rated network features, and Ii,j is the probe attack set both rating by network protocol i and protocol j. We approved the Pearson’s correlation using W(u,i) as follows.

$$sim(i.j)=\frac{\sum_{c\in I_{i,j}}(W(i.c)*R_{i,c}-A_i)(W(j.c)*R_{j,c}-A_j)}{\sqrt{\sum_{c\in I_{i,j}}(W(i.c)*R_{i,c}-A_i)^2*\sum_{c\in I_{i,j}}(W(j.c)*R_{j,c}-A_j)^2}}$$

Step3. Neighbor Selection

A set of K probe attacks is found, which is formed according to the degree of similarity between each of the network attacks with the target probe attack.

Step4. Prediction

To generate prediction of a probe attack rating prediction formula is used. Since we have got the probe attack features based on the protocol and size of the src bytes, we can calculate the weighted average of probe attacks rating. The producing prediction formula as following:

$$P_{ui}=A_u+\frac{\sum_{m=1}^{n}(R_{mi}-A_m)*sim(u,m)}{\sum_{m=1}^{c}sim(u,m)}$$

Ai is the average rating of network protocol i for all the co-rated network features, Rmi: the rating of the probe attacks to the attack i, Am: average ratings of the probe attacks m to the protocols, sim(u, m): the similarity of the probe attack and the network attacks m, n: the number of the closeness of the attack similarity.

Random Forests:

The random forests are an ensemble of unpruned classification or regression trees whose literature in relevance to intrusion detection. Random forest generates many classification trees. A tree classification algorithm is used to construct a tree with different bootstrap sample from the original data. When the formation of forest is completed, a new object which is to be classified is taken from each of the tree in the forest. A vote is given by each tree which indicates the decision of the tree decision about the class of the object. The forest selects the class with the most votes for the object. The main features of random forests algorithm are listed as follows:

1. It is unsurpassable in accuracy among the current data mining algorithms.
2. It shows efficient performance on large data sets with many features.
3. It can give the estimate of what features are important.
4. It has no nominal data problem and does not overfit.
5. It can handle unbalanced data sets.

In random forests, there is no necessity for cross validation or any test set to get an unbiased estimate of the test error. Since each tree is constructed using the bootstrap sample, approximately one-third of the total cases are omitted out of the bootstrap samples and they do not appear in the training.

The working of Random Forests is as follows:

1. Choose T number of trees to grow
2. Choose m number of variables used to split each node. m<<M, where M is the number of input variables.
3. Grow trees, while growing each tree do the following:

(a) Construct a sample of size N from N training cases with replacement and grow a tree from this new sample.

(b) When growing a tree at each node select m variables at random from M and use them to find the best split.

(c) Grow the tree to a maximal extent. There is no pruning.

4. To classify point X collect votes from every tree in the forest and then use majority voting to decide on the class label.

Binary pso:

Particle swarm optimizer is a population-based optimization algorithm using multiple candidate solutions to find the global optimum of a search space. It is inspired mainly by social behavior of flock organisms, such as swarms of birds or schools of fishes. The population is called a swarm and an individual is called a particle. A particle moves with an adaptive speed with an attempt to find the global optimum through cooperating and competing with other particles. When a specific particle finds the best solution, other particles move closer to it. Each particle represents a candidate solution to the optimization problem. They collaborate in an attempt to uncover ever-better solutions. Each particle in the swarm has two associated characteristics, a current position and a velocity. The position of a particle is influenced by the best position visited by itself (Pbest) and the position of the best particle in its neighborhood. When the neighborhood of a particle is the entire swarm, the best position in the neighborhood is referred to as the global best (gbest) particle.

When smaller neighborhoods are used, the algorithm is generally referred to as a local (lbest) PSO velocity in the previous iteration of the algorithm and the location of a particle relative to its Pbest and gbest (or hest). Therefore, at each step, the size and direction of each particle's move is a function of its own history and the social influence of its peer group. The following provide a description of a canonical continuous version of the algorithm.

Pseudo-code of the PSO algorithm

```
Initialize Population
2. WHILE (Stopping criterion is not met)
3. FOR p=1 to number of particles Select attributes
   Separate Training data and Test data using K-fold Cross-validation Train on Training data Classify using Test data Store Detection rate in an array
4. NEXT p
5. Update particle's velocity and position
6. NEXT generation until stopping criterion
```

The particle position for a particular dimension is updated as:

$$V_{pd}^{new} = w * V_{pd}^{old} + c1 * rand1(pbest_{pd} - X_{pd}^{old}) + c2 * rand2(gbest_{pd} - X_{pd}^{old}) \quad (1)$$

$$S(V_{pd}^{new}) = 1/(1 + e^{V_{pd}^{new}}) \quad (2)$$

$$if(rand < S(V_{pd}^{new}))than(X_{pd}^{new} = 1)$$

$$else(X_{pd}^{new} = 0)$$

Here, sigmoid function is used to calculate the presence of a particular attribute in the attribute set. If the value of $S(V_{pd}^{new})$ is greater than a randomly generated number between (0, 1), then it is set to 1, which means that this attribute is selected and if the value of $S(V_{pd}^{new})$ is less than the randomly generated number then it is set to 0 which means that this attribute is not selected for the next generation.

V. EXPERIMENTAL RESULTS

Performance of our system is calculated on the basis of the number of trees constructed during the training phase. More the number of trees constructed more the amount of accuracy with only a small reduction in the performance. Figure 2 shows the comparison of Random Forests with other algorithm. As can be seen, with the increase in the number of trees used in the forest, the false positive rate decreases while determining attacks.

Figure 4 shows the comparison of Random Forests algorithms with several different models. It shows that the execution times for different models do not vary very significantly. As the number of trees increases, the execution time for a given test set increases. This reduction in performance is negligible upon consideration of reduction in the rate of false positives.

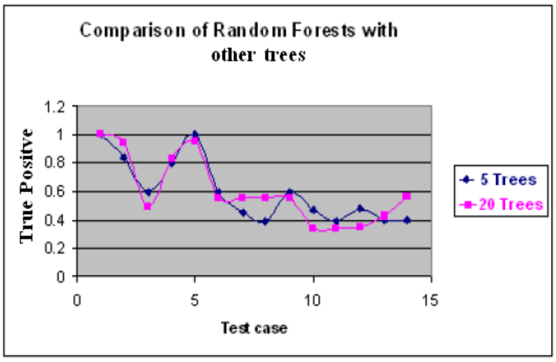
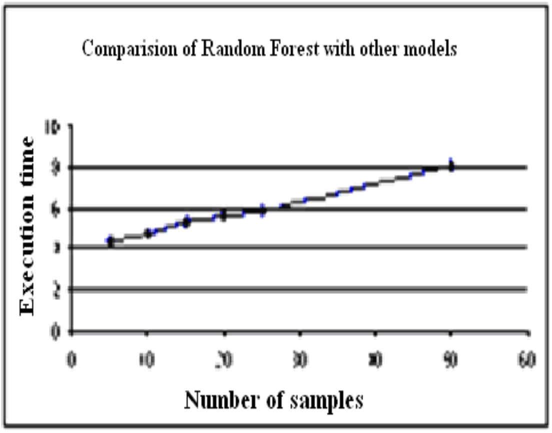
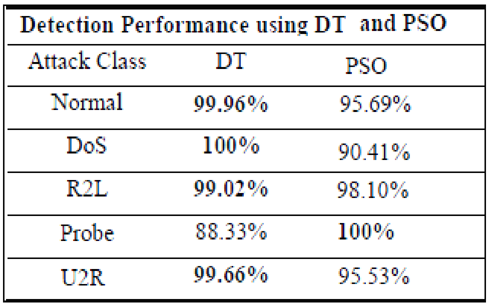


Figure 2 : True positve rate in Random Forest



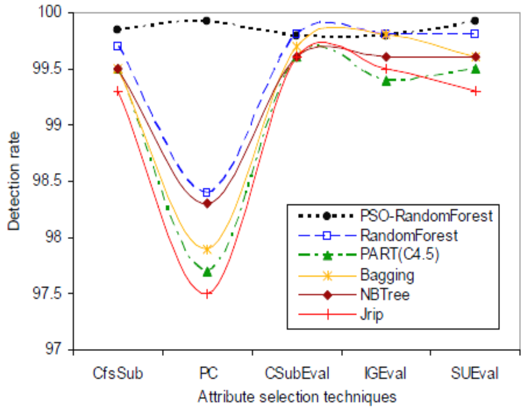
In Table 1 we compare the performance of Intrusion Detection System (IDS) Classifiers using three feature reduction techniques. As we can see from the table, DT outperformed Normal, DOS, and Probe attacks and also in R2L, and DT shows almost the same results. PSO has only higher performance in probe type of attacks.

TABLE 1 DETECTION PERFORMANCE USING DT AND PSO METHODS



Detection Performance using DT and PSO		
Attack Class	DT	PSO
Normal	99.96%	95.69%
DoS	100%	90.41%
R2L	99.02%	98.10%
Probe	88.33%	100%
U2R	99.66%	95.53%

The following figure shows the Intrusion Detection rate and False Positive rate achieved by six best classifiers.



Classification accuracy for PROBE attacks

VI. CONCLUSION AND FUTURE WORK

In this paper we apply binary particle swarm optimization and Random forest methods to intrusion detection to avoid a hard definition between normal class and certain intrusion class and could be considered to be in more than one category .We introduce the current status of intrusion detection systems (IDS) and BPSO based feature selection heuristics, and present some possible data mining random forest technique for solving problems. BPSO based method with data reduction for network securities are discussed. As can be seen, with the increase in the number of trees used in the forest, the false positive rate decreases while determining attacks. The Collaborative filtering technique and random forests algorithm has been successfully applied to find patterns that are suitable for prediction in large volumes of data. Basically, in intrusion prediction, we can predict a specific intrusion based on symptoms. Improvements can be made on the collaborative filtering algorithm in the subsequent researches in order to make sure the precision of data source and improve the mining efficiency.

REFERENCES

1. D.S Bauer, M.E Koblentz,. NIDX- an expert system for real-time network intrusion detection, Proceedings of the Computer Networking Symposium, 1988. pp. 98-106.
2. Herv Debar, Marc Dacier, and Andreas Wespi, "Towards a Taxonomy of Intrusion Detection Systems", IBM Technical Paper, Computer Networks, Vol.39, Issue 9, pp. 805-822, April 1999.
3. S. Chebrolu, A. Abraham, J. P. Thomas, Feature Deduction and Ensemble Design of Intrusion Detection Systems, Computer & Security, 2004.
4. A.Sundaram, "An introduction to intrusion detection, Crossroads": The ACM student magazine, 2(4), April 1996.
5. D. Denning, "An intrusion-detection model", In IEEE computer society symposium on research in security and privacy, 1986, pp. 118- 131.
6. Zhang Jianpei, Liu Jiandong, Yang Jing. Data Preprocessing Method Research for Web Usage Mining [J]. Computer Engineering and Applications, 2003, (10):191-193(In Chinese).
7. Yu kai,Xu Xiao-wei,Martin Ester,et al.Collaborative Filtering and Algorithms:Selecting Relevant Instances for Efficient and Accurate Collaborative Filtering[C]//Proceedings of the Tenth International Conference on Information and Knowledge Management. 2001:239-246.
8. Huang Z,Chen H,Zeng D.Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering [J].ACM Transactions on Information Systems, 2004, 22(1):116-142.
9. Herlocker J,Konstan J,Terveen L,et al.Evaluating Collaborative Filtering Recommender Systems.ACM Trans.on Information Systems(TOIS),2004,22(1):5-53
10. KDD'99 datasets,The UCI KDD Archive, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, Irvine, CA, USA, 1999.

AUTHORS PROFILE

G. Sunil kumar, received his M.C.A from JNTU Kakinada. He is currently working as Lecturer in department of Computer Applications in Maris Stella College and research project developer in GSK research and development solutions. His research areas include Data Mining applications in network environment, Computer Networks, Image Processing, Network Security attacks and its countermeasures. He completed computer science, mathematics, statistics research projects for different universities research scholars.