Proceedings of the 2001 IEEE
Workshop on Information Assurance and Security
United States Military Academy, West Point, NY, 5-6 June, 2001

A Scalable Clustering Technique for Intrusion Signature Recognition

Nong Ye, *Member, IEEE* and Xiangyang Li, *Member, IEEE*

***Abstract*-- This paper presents a data mining algorithm, namely Clustering and Classification Algorithm – Supervised (CCA-S), which we developed for detecting intrusions into computer network systems for intrusion detection. CCA-S is used to learn signature patterns of both normal and intrusive activities in the training data, and to classify the activities in the testing data as normal or intrusive based on the learned signature patterns of normal and intrusive activities. CCA-S differs from many existing data mining techniques in its ability in scalable, incremental learning. We tested CCA-S and two popular decision tree algorithms, and obtained their performance for an intrusion detection problem. CCA-S produced better intrusion detection performance than these popular decision tree algorithms.**

***Index Terms*-- Computer security, intrusion detection, signature recognition, and data mining.**

I. INTRODUCTION

COMPUTER security has become a critical issue with the rapid development of business and other transaction systems over the Internet. Intrusion detection is to detect intrusive activities while they are acting on computer network systems. One category of intrusion detection techniques, namely signature recognition (or misuse detection in some literature) relies on learning and recognition of intrusion signatures. The automatic learning of intrusion signatures is usually based on some kind of data mining algorithms. However, many existing data mining algorithms, e.g., popular decision tree algorithms as well as statistical clustering and classification techniques such as hierarchical clustering and K-means method, cannot handle large amounts of computer audit data or network traffic data that capture activities in computer network systems for intrusion detection in a scalable and incremental manner during training to learn signature patterns of intrusions. The incremental learning ability requires a data mining algorithm to update the already learned intrusion signatures whenever additional data of new intrusions are obtained. As new intrusion scenarios come out and become known, their signatures should be captured by updating the set of already learned intrusion signatures from previous data sets of intrusive activities.

CCA-S (Clustering and Classification Algorithm-Supervised) is developed in our project to overcome the problems of decision tree algorithms and statistical clustering techniques in scalability and incremental learning. Section 2 briefly reviews existing intrusion detection techniques and specifies the requirements for the scalability and the incremental learning ability of a signature recognition technique for computer intrusion detection. Section 3 describes the CCA-S algorithm. Section 4 presents the application of the CCA-S algorithm to learning and classifying normal and intrusive activities in some data sets, and shows the intrusion detection performance of CCA-S in comparison with the intrusion detection performance two popular decision tree algorithms for the same data sets. Section 5 summarizes the paper.

II. INTRUSION DETECTION AND SIGNATURE RECOGNITION

An intrusion can be defined as a series of activities aiming at compromising the security of a computer network system [1]. Intrusions may take many forms: external attacks, internal misuses, network-based attacks, information gathering, denial of service, and so on [2-3]. Intrusion detection is an important step of protecting the computer network system from intrusions. Intrusion detection tries to detect intrusive activities while they are acting on the computer network system.

There are two general categories of intrusion detection techniques: anomaly detection and signature recognition (pattern matching) [8,9]. Anomaly detection techniques learn a profile of normal activities for a subject in a computer network system, and look for intrusive activities that deviate largely from the norm profile. The subject may be a user, a host machine, or a network. Signature recognition techniques learn signature patterns of intrusions and use those signatures to classify observed activities in a computer network system as normal or intrusive. Anomaly detection techniques can detect unknown intrusions. However, anomaly detection techniques may also produce false alarms if the detected anomalies are caused by events other than intrusions. Signature recognition techniques are accurate in detecting known intrusions, but cannot detect novel intrusions. Hence anomaly detection techniques and signature recognition techniques are often used together to complement each other. This paper focuses on signature recognition techniques.

Since the seminal work on the rule-based pattern matching model by Denning [4], we have seen the rapid growth of intrusion detection systems (IDS) based on signature recognition techniques [5]. The performance comparison of some IDS can be found in [6,7].

The core of an IDS based on signature recognition is the analysis engine that learns signature patterns of intrusions either manually or automatically [8-9] and uses those signatures to classify observed activities in a computer network system as normal or intrusive. The manual learning of

ISBN 0-7803-9814-9/$10.00 © 2001 IEEE

intrusion signatures is cumbersome and impractical. Many existing signature recognition techniques such as those based on state transition analysis and Colored Petri Net lack the automatic learning capability. Considering the changing of intrusion scenarios over time, it is difficult to manually keep intrusion signatures updated for people.

The automatic learning of intrusion signatures from historic data containing examples of normal and intrusive activities is necessary. Many data mining algorithms from such fields as machine learning and statistical clustering have the potential to serve as signature recognition techniques for computer intrusion detection. The set of the learned intrusion signatures should be updated whenever additional data of normal and intrusive activities become available. There usually are large amounts of such historic data.

Existing IDS based on signature recognition focus on two kinds of activity data from a computer network system: network traffic data and computer audit data. A variety of activity attributes can be obtained from these data, producing nominal variables such as the event type, user id, process id, command, remote IP address, and numerical variables such as the time stamp, CPU time, etc. Activity data from a computer network system are huge and complex. A computer auditing facility, such as Solaris Basic Security Module (BSM), can easily produce hundreds of thousands of audit records per day, and the attributes extracted from each audit record can reach hundreds (e.g., 284 event types). As intrusive activities change over time, additional activity data must be taken into account to capture signature patterns of new intrusive activities. That is, we need a data mining algorithm that supports the scalable, incremental learning.

Decision tree algorithms are a very popular data mining technique for learning patterns from data and using these patterns for classification. In [10], we report the application of decision tree algorithms to computer intrusion detection. Some of the existing decision tree algorithms support the incremental learning [11,12]. Some other data mining algorithms support the scalability [13-15]. However, we do not have decision tree algorithms that support both scalability and incremental learning. This paper presents a data mining algorithm that supports the scalable, incremental learning for computer intrusion detection based on signature recognition. We apply this algorithm to computer audit data for intrusion detection.

III. CCA-S

In CCA-S, a data record is considered as a data point in a p-dimension space. Each dimension is either a numerical or a nominal variable, called predictor variable, representing one attribute of the data. Each data point has also a label indicating the class of the data record, called the target variable. For computer intrusion detection based on signature recognition, the target variable is a binary variable with two possible values: 0 for normal and 1 for intrusive. CCA-S clusters data points based on two criteria: the distance between data points, and the class label of data points. Only data points that are close and same in their class label can be grouped together to form a cluster. Each cluster represents a signature pattern for normal activities or intrusive activities, depending on the class label of the data points in the cluster.

Formally, each data point is a (p+1)-tuple with the attribute variable vector X containing the *p* dimensions of predictor variables and one target variable - Y. The training data set has N data points.

A. Training (supervised clustering)

It takes mainly two steps to incrementally group the N data points in the training data set into clusters.

1. Scan the training data and compute the relative importance of each prediction variable with respect to the target variable. This step calculates the coefficient of the correlation between each predictor variable Xi and the target variable Y. In addition, two dummy clusters, one for normal activities and another for intrusive activities, are created. The centroid of the dummy cluster for normal activities is denoted by the mean vector of all the data points for normal activities in the training data set. The centroid of the dummy cluster for intrusive activities is denoted by the mean vector of all the data points for intrusive activities in the training data set.
2. Incrementally group each point in the training data set into clusters. Given a data point X, we find the nearest cluster L to this data point using a distance metric weighted by the correlation coefficient of each dimension. If L has the same class label as that of X, we group X with L; otherwise, we create a new cluster with this data point as the centroid of the new cluster.

We then repeat the above steps until we process all the data points in the training data set.

B. Classification

There are two methods to classify a data point X in a testing data set.

1. Assign the data point X the class dominant in the k nearest clusters which are found using a distance metric weighted by the correlation coefficient of each dimension; or
2. Use the weighted sum of the distances of k nearest clusters to this data point to calculate a continuous value for the target variable in the range of [0, 1].

C. Incremental update

The statistics from the correlation and the clustering are stored. When new training data become available, each step of the training can be repeated for new data points to update the clusters incrementally.

An estimate of the computation cost of CCA-S is provided here. Given N data points, and the total number of the resulting clusters L, the computation cost for training is $O(p*N*L)$. And the computation cost of classifying a data point during testing is $O(p*L)$. Hence, CCA-S is scalable to even large amounts of training data.

We define various distance metrics that may be used in

CCA-S, like the weighted Euclidean distance, weighted Chi-squared distance, and weighted Canberra distance.

There are several methods to reduce the number of the resulting clusters for reducing the computation cost. For example, we can set the maximum number of clusters for a certain class label. After the number of the clusters for that class reaches the maximum number, new data points will not lead to the creation of new clusters but to the update of the existing clusters.

IV. Implementation and Result Analysis

CCA-S can be used to automatically learn signatures of normal activities and intrusive activities. Signatures are represented by the clusters that result from the training stage of CCA-S. We can assign an Intrusion Warning (IW) value to each luster based on the class label of each cluster.

As described in section 3, CCA-S classifies a data point into a class during testing. For intrusion detection, there are two classes: normal and intrusive. The predictor variables are the attributes of the p-dimensional data points. There are several distance metrics from which we may choose. In this study we use the weighted Canberra distance metric as shown below:

$$d(X,L)=\sum_{i=1}^{P}\frac{|X_i-L_i|}{X_i+L_i}C_i^2$$

where X_i is the ith attribute variable of a data point X, L_i is the ith coordinate of the centroid of a cluster L, and C_i is the correlation coefficient of the ith attribute variable and the target variable.

For intrusion detection, the IW level of a data point in the testing data set is determined using the second method of classification to obtain a value between 0-1. The Receiver Operating Characteristic (ROC) analysis of the detection performance of CCA-S is performed to determine the hit rates and the false alarm rates using varying signal thresholds. If the IW value of the data point is greater than a signal threshold, the data point is classified as intrusive, and an alarm signal is produced. The false alarm rate and the hit rate for a given signal threshold are calculated based on these alarm signals and true class labels of the data points in the testing data set. The hit rates and false alarm rates for various signal thresholds can be plotted as a ROC curve. The closer a ROC curve to the top-left corner representing the 0% false alarm rate and the 100% hit rate, the better the detection performance of an intrusion detection technique.

We use computer audit data from the Basic Security Module (BSM) in the Solaris operating system in this study. We have two data sets, one for training, and another for testing. Each data set consists of two parts. The normal event part is from the 1998 DARPA – the MIT Lincoln laboratory evaluation data. The intrusive event part is obtained by simulating a number of intrusion scenarios. In the training data set, there are 1613 normal events and 526 intrusive events. In the testing data set, there are 1406 normal events and 1225 intrusive events. For the training data, the class label for an intrusive event is 1, and the class label for a normal event is 0.

Each data set consists of audit events in sequence. Each audit event in this event stream has several attributes. These attributes are the variables that contain information about the activities. The attributes of an audit event include the event type, user ID, process ID, command, time, remote IP address, and so on.

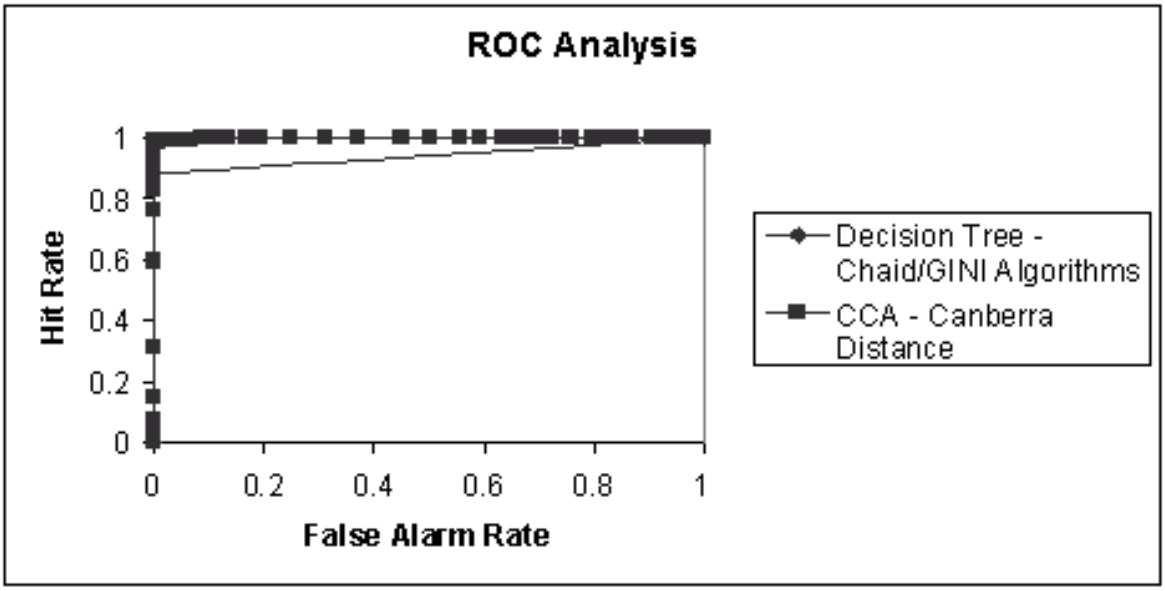

Fig. 1. ROC comparisons on small intrusion detection data

We use only the information of the event type to form our predictor variables. There are 284 different event types in Solaris, but only 30 event types appear in the training data set. Hence, we have 30 predictor variables for 30 event types respectively. Each predictor variable denotes the occurrence frequency of the corresponding event type in the recent past of the event stream. We use the Exponentially Weighted Moving Average (EWMA) technique to compute the observation values of the 30 predictor variables for an audit event *t* in the event stream as follows:

ISBN 0-7803-9814-9/$10.00 © 20[illegible] IEEE

$X_i(t) = \lambda * 1 + (1-\lambda) * X_i(t-1)$ if the audit event t belongs to the ith event type

$X_i(t) = \lambda * 0 + (1-\lambda) * X_i(t-1)$ if the audit event t is different from the ith event type

where $X_i(t)$ is the smoothed observation value of the ith variable for the audit event t, λ is a smoothing constant which determines the decay rate. The decay rate allows us to introduce aging while obtaining the observation values of X for the current event. More weight is given to events in the more recent past of the current event.

In our study, we initialize $X_i(0)$ to 0 for i = 1,…,30. We let λ be 0.3 – a typical value for the smoothing constant [14], which corresponds to a 15-event observation window.

Therefore, for each event in the training data and the testing data, we obtain a vector of (X_1,…, X_{30}) - a data point in a 30-dimensional space. Figure 1 shows the ROC curves of CCA-S and two popular decision tree algorithms: CHAID and CART based on the GINI split index available in a commercial data mining package - AnswerTree from SPSS. The detection performance of CCA-S is better than that of the decision tree algorithms. In fact, CCA-S achieves the 100% hit rate at the 0% false alarm rate for this testing data set.

V. Conclusion

Many data mining techniques such as decision tree algorithms cannot deal with large data sets in a both incremental and scalable manner. We develop the CCA-S algorithm to overcome these problems. The application of CCA to computer intrusion detection based on signature recognition demonstrates the better detection performance of CCA-S than that of popular decision tree algorithms.

Acknowledgment

This work is sponsored in part by the Air Force Office of Scientific Research (AFOSR) under grant number F49620-99-1-001, and the Defense Advanced Research Project Agency (DARPA) /Air Force Research Laboratory – Rome (AFRL-Rome) under grant number F30602-99-1-0506. The U.S. government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of, AFOSR, DARPA/AFRL-Rome, or the U.S. Government.

References

[1] Heady, R., Luger, G., Maccabe, A., and Servilla, M., The Architecture of a Network Level Intrusion Detection System, Technical Report CS90-20, Department of Computer Science, University of New Mexico, August 1990.

[2] Graham, R., FAQ: Network Intrusion Detection Systems, http://www.robertgraham.com/pubs/network-intrusion-detection.html, 2001.

[3] Escamilla, T. *Intrusion Detection: Network Security beyond the Firewall*. John Wiley & Sons, New York, 1998.

[4] Denning, D., An Intrusion-detection Model, *IEEE Transactions on Software Engineering*, 13(2), pp. 222-232, February 1987.

[5] Frincke, D. A., and Huang, M., Recent Advances in Intrusion Detection Systems, *Computer Networks*, 34, pp. 541-545, 2001.

[6] Lippmann, R., Fried, D., Graf, I., Haines, J., Kendall, K., McClung, D., Weber, D., Webster, S., Wyschogrod, D., Cunningham, R., and Zissman, M., Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation, *Proceedings of the DARPA Information Survivability Conference and Exposion,* Los Alamitos, IEEE Computer Society, pp. 12-26, 2000.

[7] Lippmann, R., Haines, J. W., Fried, D., Graf, I., Korba, J., Das, K., The 1999 DARPA off-line intrusion detection evaluation, *Computer Networks*, No. 34, pp. 579-595, 2000.

[8] Axelsson, S. Intrusion Detection Systems: A Survey and Taxonom, Report, Department of Computer Engineering, Chalmers University of Technology, Goteborg, Sweden, 2000.

[9] Debar, H., Dacier, M., and Wespi, A., Towards a taxonomy of intrusion-detection systems, *Computer Networks*, 31, pp. 805-822, 1999.

[10] Sinclair, C., Pierce, L., and Matzner, *S.*, An application of machine learning to network intrusion detection, *Proceedings of 15th Annual Computer Security Applications Conference (ACSAC '99)*, pp. 371-377, 1999.

[11] Utgoff, P. E., Berkman, N. C. and Clouse, J. A. Decision Tree Induction Based on Efficient Tree Restructuring, *Machine Learning Journal*, 10, pp. 5-44, 1997.

[12] Crawford S. L., Extensions to the CART Algorithm, *International Journal of Man-Machine Studies,* 31, pp. 197-217, 1989.

[13] Goil, S., and Choudhary, A., A parallel scalable infrastructure for OLAP and data mining, *International Symposium Proceedings on Database Engineering and Applications*, IEEE, pp. 178 –186, 1999.

[14] Chaudhuri, S., Fayyad, U., and Bernhardt, J., Scalable classification over SQL databases. *15th International Conference on Data Engineering*, IEEE, pp. 470-479, 1999.

[15] Shafer, J., Agrawal, R., and Mehta, M., SPRINT: A Scalable Parallel Classifier for Data Mining, *Proceedings of the 22nd VLDB Conference*, Mumbai, India, 1996.