



Application of decision tree classifiers to computer intrusion detection

Nong Ye & Xiangyang Li
*Department of Industrial Engineering,
Arizona State University, U. S. A.*

Abstract

There is an increasing demand for techniques to detect intrusions into a computer and network system for information security and assurance. This paper describes our research effort on the application of a data mining technique, decision trees, to automatically learn and recognize intrusion signatures for intrusion detection. In our study, decision tree classifiers are used to classify activities in a computer and network system into different states and determine the possibility of an intrusion based on the state classification. Our design of decision tree classifiers is based on an incremental tree induction algorithm. Two decision tree classifiers are developed. One decision tree classifier examines single events of activities in a computer and network system for intrusion detection (single event version). Another decision tree classifier examines a sequence of events in a moving window at a given time for intrusion detection (moving window version). We use computer audit data for training and testing the decision tree classifiers. The testing results of the two decision tree classifiers are analyzed and compared. The "moving window" version of the decision tree classifier produces better performance in intrusion detection.

1 Introduction

Information security has become a critical issue with the rapid development of business and other transactions on Internet. One basic task in preventing a computer and network system from intrusions from internal or external sources is to detect the occurrence of intrusions.

Many data mining algorithms have been used to analyze data from a computer and network system to detect intrusions. This paper describes our



effort on applying the decision tree technique to intrusion detection (ID). In our study, decision tree classifiers aim at automatically capturing signature patterns of normal and intrusive activities through learning from training examples. Decision trees of signature patterns are then used in the testing to classify an observed activity in a computer and network system into a state and to determine the likelihood of an intrusion according to the state classification.

Our application is based on the Incremental Tree Induction (ITI) algorithm proposed by Utgoff [1]. The design, key technological elements of our implementation are given after a brief introduction of the decision tree technique and intrusion detection. We apply decision tree classifiers to a stream of computer audit events from a computer and network system. The testing results are presented. In the end of the paper, some directions for future work are also discussed.

2 Decision trees

There are two kinds of variables used in a decision tree technique. One is the target variable or class; the other is a predictor variable. In the training stage, the tree induction algorithm partitions the training examples labeled by the target variable recursively until a stopping criterion is met. The partition is performed by selecting a test on some predictor variable that creates a small set of branches, passing each example down the corresponding branch, and then treating each branch as a sub-problem. After each partition, the set of training examples in a sub-problem has less inconsistency with respect to the target value. A common stopping criterion for a branch is that all examples be of the same class or target value, and then producing a leaf.

A basic purpose of a classification study can be either to produce an accurate classifier or to uncover the predictive structure of the problem [2]. For the former task, the decision tree technique is used to classify a problem region into several sub-regions with all the components in one sub-region having the same target value. For the latter purpose, the tree structure shows the relationship between the predictor variables and the target variable. A main advantage of decision tree classifiers is that they reduce the problem complexity by approximating the global decision regions using simpler local decision regions at each level of the tree. This problem is possibly multi-modal [3]. The decision tree technique uses a stepwise classification method to deal with a problem in a high dimensionality space.

3 Network intrusion and detection

An intrusion is a set of activities aimed at breaking the security of a computer and network system. Intrusions may take many forms: external attacks, internal misuses, network-based attacks, information gathering, denial of service etc. [4]

There are three major kinds of intrusion detection techniques in practical use: anomaly detection and signature recognition. Anomaly detection techniques



focus on the learning of a profile of normal activities in a computer and network system. Signature recognition techniques recognize predefined intrusion signatures from activities in a computer and network system.

Most existing signature recognition techniques for intrusion detection lack the automatic learning capability. This is a major problem for the existing signature recognition techniques, because it is difficult to manually code all intrusion signatures and update intrusion signatures as computing technology and intrusions change over time.

Computer audit data are used to capture activities in a computer and network system. Computer audit data have some special features. First, they are "simultaneous" data with a mixture of information from both normal and intrusive activities in a computer and network system. Generally we can hardly draw a line between these two kinds of information in the computer audit data. A practical technique must carefully choose an appropriate representation of data to in order to achieve accurate results. Also computer audit data are huge and complex. A large number of variables can be used to describe activities in a computer and network system. Some of them take discrete values while others take continuous values. Many of them have a wide range of values. The volume of such data easily exceeds hundreds of thousands of records per day. We have to use algorithms that are feasible both in time and storage.

Decision trees can be used as a signature recognition technique to automatically learn and recognize intrusion signatures for intrusion detection. The decision tree technique requires examples of both normal and intrusive activities in the training. In the testing, the decision tree technique classifies an observed activity into normal or intrusive. The application of the decision tree technique to intrusion detection is presented as follows.

- 1) The decision tree technique is used to learn signatures of normal activities and intrusions automatically. Signatures are stored in the tree structure inducted from the historic data of a computer and network system. As a side product, we can provide an intrusion rating (an Indication and Warning value – an IW value) for an observed activity.

- 2) Since each leaf in a decision tree structure represents a specific status of activities with a certain IW value, it can be regarded as a special state of the system. Then the decision tree technique can classify system activities into different states. Such aggregate state information is very useful for other techniques to perform aggregate intrusion analysis.

There are several requirements for the decision tree technique when it is applied to intrusion detection.

- 1) Incremental training. Due to a huge amount of computer audit data, it is impossible to build a decision tree on a huge set of training examples in one step. It is necessary to have the ability to incrementally incorporate new training examples into an existing tree.

- 2) Inconsistent training examples. It is very common in the practical data to find the examples with different target values but the same predictor attributes - inconsistent training examples. This demands the ability of a tree induction algorithm in accommodating all these examples in one leaf of a decision tree.



3) Avoiding data over-fitting. When there is noise in the training data and there is an overlap between partitions, it is very possible to over-fit data so that the resulting classifier fits the training data much better than the population of data including the testing data. Tree pruning can be used to avoid this problem. Choice of predictor variables and the training sample also affect it a lot.

4 Design of decision tree classifiers

For intrusion detection, the decision tree technique implements the following functions.

1) In the training stage, the tree induction algorithm builds a decision tree through learning the normal and intrusive examples in the training data, with different states of activities represented by different leave.

2) In the testing or detecting stage, the decision tree is used to determine the state of an observed activity and provide an IW value.

Computer audit data consist of audit events. The training data for our decision tree classifiers are audit event streams of intrusive activities and normal activities. A different set of audit event stream for intrusive activities and normal activities is used as the testing data. In this paper, "event" means the activity in a computer and network system at a given time stamp and the information associate with that activity at that point. "Attributes" are the variables of the event whose values represent certain information of such an activity.

To meet the requirements of intrusion detection, our decision tree classifiers are based on an Incremental Tree Induction (ITI) algorithm developed at the University of Massachusetts. It has many advantages to be used for intrusion detection. It has multiple training modes including the error-correction mode and the batch mode. The error-correction mode is for incremental training. It is capable of processing inconsistent training examples to get the states with an averaged IW value falling between 0 and 1. The induction algorithm is based on the information entropy. A gain-ratio metric is used in a decision test to avoid the selection of attributes with many uniformly distributed values. It uses an efficient data structure to deal with a large data set. However, since it has a binary tree structure, it tends to grow the tree with many levels. The resulting tree can become unwieldy and not represent results efficiently.

We have modified the ITI algorithm to meet the requirements of intrusion detection. We want to get the IW value for a certain state so we design an additional function to record the IW information of training examples in each leaf and calculate the average IW value for that state. In the meantime, we have embedded the ITI algorithm into our research framework, and have solved the memory management problem.

4.1 Single event version and moving window version

In our study, the training and testing data take the form of an event stream. This event stream consists of a set of audit events over time. Each event has several attributes. We can use some of these attributes as the predictor variables and



another one as the target variable to construct a decision tree classifier. An event in the training data is labeled as 1 if the event is a part of an intrusion and 0 otherwise. Since in our study a decision tree classifier tries to separate intrusions from normal activities, the label of the event is used as the target variable. We use the “event type” as the only predictor variable. A direct method is to use this attribute as the predictor variable.

The induction algorithm processes events in the training data one at a time to grow the tree. We call this version of decision tree classifiers as a “single event” classifier.

Generally, an intruder launches a series of actions in an intrusion scenario to achieve an objective. These actions result in an event sequence ordered by time stamp. Such a event sequence contains a temporal characteristic of intrusive activities. If this kind of temporal information can be incorporated into the predictor variables, better results may turn out.

Hence, we also use a moving window to performance a data transformation as shown in figure 1. Assuming that we snapshot a segment of an event stream through a moving window. After this moving window slides across the original data stream, a set of new data is produced from the sequential records of such snapshots. Each snapshot in the moving window records a sequence of events. We call the decision tree classifier using this transformed data as the “moving window” classifier.

How to choose the moving window size affects the decision tree a lot. In general every type of attack scenario has its unique event set length around the mean value of certain distribution. If the moving window size is much greater than this length, the information retrieved from original data is not fit for the decision tree to catch such attack. On the other hand, if the moving size is much less, we get partial poor information of this attack in our new data unit. According to this consideration, maybe we should use multiple moving window sizes for different attack scenarios to get better transfer and result. But that is too difficult to realize. In our research, we just make this size 100, that is the approximate average length of all kinds of attack scenarios.

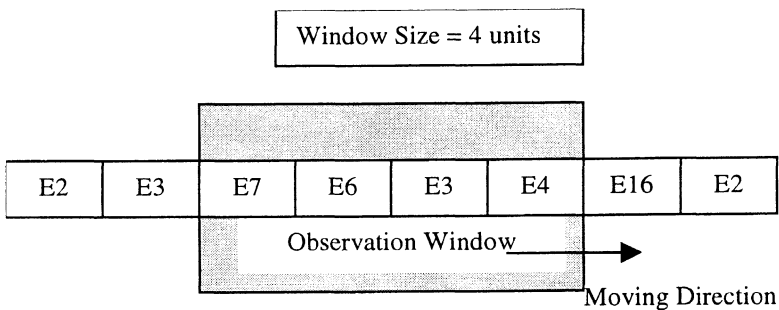


Figure 1: A moving window.



4.2 Two mapping methods for moving window transfer

For the moving window classifier, there are two methods to represent transformed data. We assume the moving window size is A and there are A positions, and the value range for the event of the original data is B . In event stream, this is the number of possible event types. There are about 284 system event types defined in Solaris 2.6, and thus B equals to 284. One method of representation is to record the sequence of event types mapped to the positions of the moving window. Each of the A positions in the moving window is an attribute variable in the new data unit and a predictor variable for the decision tree classifier. The value of a variable can be any event of the B possible event types. The complexity to deal with the new data is B^A .

In the other method we use each possible event type in event stream as an attribute variable for new data, thus creating B attributes. There are two ways to set the values of these attributes. In the "count" form, we set the value to be the count number that the related event type appears in the observation window. For the second one it just records if this event type exists in this window - 0 or 1 respectively. This form is called "existence" form. For the "count" form the possible complexity is A^B . For the "existence" form, it is A^2 .

The advantage of the first mapping method is that the new data retain the position information in the raw event stream. We comfortably regard it more useful and understandable. However, since the value range is usually much larger with regard to a given window size in many cases, i.e. B is much greater than A , we have to choose the second method to obtain a feasible computation complexity.

5 Implementation and results

There are two different sets of data, one set for training and another set for testing. Each set of event stream has two parts of data. One part is from normal activities produced by the Lincoln Lab at MIT in 1998. Each event has a label of 0. The other part is from intrusion scenarios simulated in our lab. The IW label of each event is 1. We use the same training set to build the single event classifier and the moving window classifier. Then both of them are applied to the testing set of event stream, and produce two sets of testing results. The measurement of decision tree performance may cover many aspects such as accuracy, leave or tree size, and CPU cost etc. Among these we focus on the accuracy performance. The predicted IW value for an event in the testing data is compared with the original IW label for that event.

5.1 Single event classifier

For the single event classifier we use only the event type as the predictor variable. Figure 2 shows the Receiver Operating Characteristic (ROC) curve for the single event classifier in terms of false alarm rates and detection rates when using various decision thresholds.

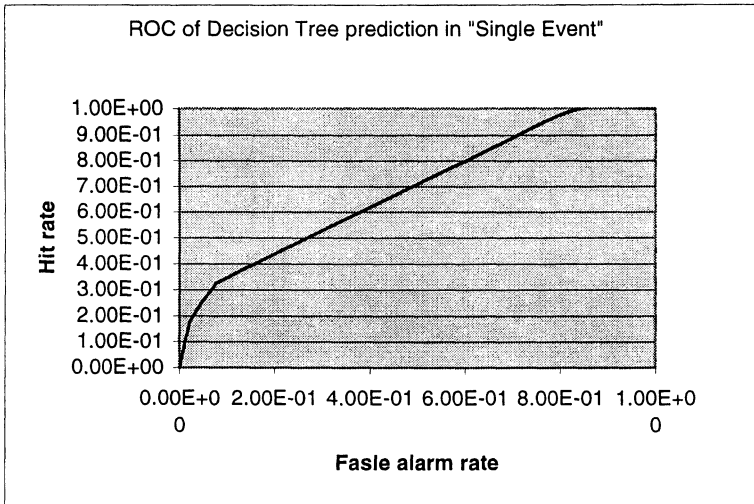


Figure 2: ROC curve for the single event classifier.

This curve seems only a little above the reverse diagonal line. The problem is due to the shortcoming of using the single event representation of data. We recall that the main advantage of decision tree classifiers is that it approximates the global decision regions by using simpler local decision regions at each level of the tree. This implies that decision tree classifiers are suitable to multivariate problems. In the “single event” classifier, the classifier simply enumerates the frequency of normal or intrusive categories for each event type with mixed IW classes. The decision tree classifier cannot provide much benefit on such a single-variable task.

5.2 Moving window classifier

For the moving window classifier, we choose the second mapping method – using each event type as an attribute variable for the transformed data. Although not all event types appear in the training data, the number of attributes is set to 284. To further reduce the complexity, the “existence” value is used.

Figure 3 shows the ROC curve for the testing results of the moving window classifier. It has much better performance than the single event classifier. In fact, for a wide choice range of threshold values, the detection rates remain at about 0.9 while false alarm at a little over 0.1.

Figure 4 and 5 show the original labeled values and the predicted IW values respectively along the event stream. We can see that except for some events near the central part of the event stream, these two kinds of values almost exactly match each other. Although this classifier is produced from a small data sample, it provides the evidence that the decision tree technique can be used effectively for signature recognition of intrusions.

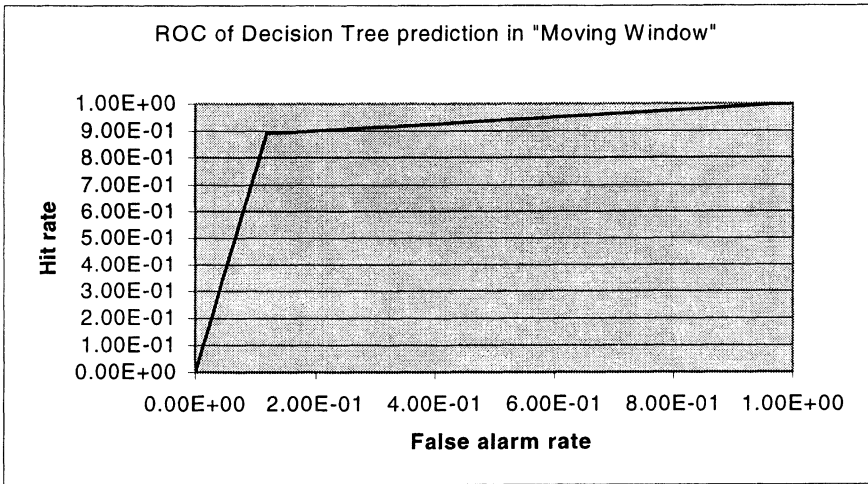


Figure 3: ROC curve for the testing result of moving window classifier.

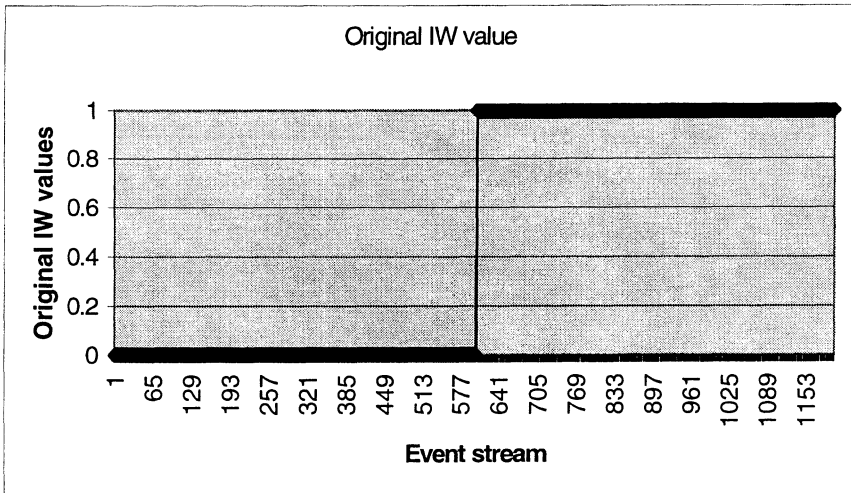


Figure 4: Original IW value.

6 Conclusion and future work

In this paper we show the application of the decision tree technique to intrusion detection and the testing results. We demonstrate that decision tree

classifiers have a reasonable ability in classifying intrusions. In the future there are several issues to be considered to improve the application of the decision tree technique to intrusion detection. An obvious consideration is the design of the target and predictor variables, i.e. how to filter original information to capture essential information better. A comprehensive knowledge of the target computer and network system is very important. Another important issue is to improve the decision tree algorithms to meet the requirements of intrusion detection. For example, statistics-based CART trees have the strength to process an overlap between partitions, which is common in complex data from a computer and network system. Such decision tree algorithms should have the ability to allow multiple tests or combine tests at a single decision node to reduce the number of leave and states. For practical intrusion detection systems, the computation cost that is feasible for a large volume of data is always an importance issue.

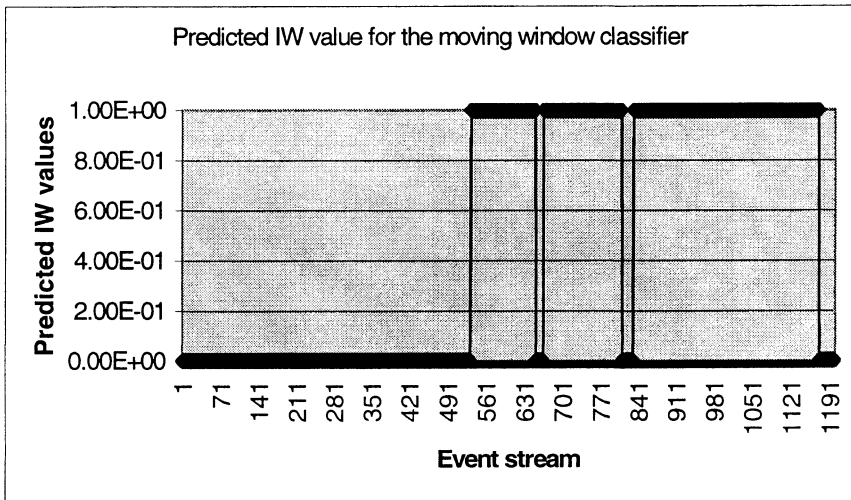


Figure 5: Predicted IW values for moving window classifier.

References

- [1] 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.
- [2] Breiman, L., Friedman, J., Olshen, R. and Stone, C. *Classification and Regression Trees*, Wadsworth Inc., 1984.
- [3] Safavian, S. R. and Landgrebe, D. "A Survey of Decision Tree Classifier Methodology," *IEEE Transactions on Systems, Man and Cybernetics*, **21**, pp. 660-674, 1991.



- [4] Ye, N., Giordano, J., Feldman, J. and Zhong, Q. Information Fusion Techniques for System-Level Intrusion Detection, Proceedings of the 1998 Information Technology Conference, pp. 117-120, 1998.