2004－CSEC－27　(13)
2004／12／20

ベイジアンネットワークを用いた侵入検知システムのパフォーマンス向上の一検討

林　經正　　ファン　ステファニー　　樗林　亮介　　小林　潔　　太田　聡
ＮＴＴ未来ねっと研究所

本稿では、ネットワークに接続されたリモートホストからの未知の不正侵入攻撃を防ぐために、ベイズネットワーク（決定木の一種）を用いた侵入検知システム（IDS）を構成し、このネットワークのサイズを減らすことにより、検知処理を高速化する方式を報告する。本方式により、検知結果の正確性を損なわずに高速検知処理を実現できることを示す。また、実験結果から攻撃特徴に注目したＩＤＳ構成法を提案する。

A Performance-tuning Method in Intrusion Detection using Bayesian Networks

Tsunemasa Hayashi, Stephanie Fung[1], Ryosuke Kurebayashi, Kiyoshi Kobayashi, and Satoru Ohta
NTT Network Innovation Laboratories

We construct an anomaly-based Intrusion Detection System (IDS) using Bayesian networks to protect a computer network (or host device) from malicious traffic. This type of IDS is capable of detecting new variants of attacks. In this paper, we propose a performance-tuning method to reduce the size of Bayesian networks, and show that our method can reduce the calculation costs without degrading performance. We then propose an IDS structure to achieve high speed operation.

1. Introduction

Recent years have seen the large-scale deployment of network infrastructures for broadband access, such as ADSL (Asymmetric Digital Subscriber Line) and FTTH (Fiber to the Home). As a result, many PCs are connected to the internet all the time. Intrusion attacks from a remote host have become a serious problem on the broadband network. To detect and stop malicious network activities, Intrusion Detection Systems (IDSs) such as Snort, Dragon, and Real-secure [1]-[3] were developed. These IDSs evaluate IP packets by comparing them with a database of attack information, or “signatures”, in a manner similar to that of anti-virus tools. This type of IDS is known as a "misuse-based IDS". With a misuse-based IDS, the network administrator must update the database with new signatures as previously unknown attacks are discovered. The ever-increasing number of variant attacks has resulted in a continuously growing database of signatures that, to date, numbers to more than 2500 [1]. The trend towards reduced operation speed due to the database size, coupled with the inability of the misuse-based IDS to adapt to new variations of existing attacks, calls for a more effective approach to intrusion detection.

The anomaly-based IDS [1] tries to overcome the inadaptability issue by first creating a model of normal system behaviour, whose focus may be on users, applications, or network traffic. It can then compare this model to current activity on the network and trigger an alert if an activity deviates past a threshold. Although anomaly-based IDSs are more successful in detecting novel intrusions than misuse-based IDSs, both suffer from efficiency issues that make them potential network bottlenecks. Current attempts to improve efficiency in anomaly-based IDSs, though effective, require detailed background knowledge about the network domain.

This paper uses Bayesian networks as a tool for implementing an anomaly-based IDS. The powerful ability of algorithms to construct these networks by extracting the most relevant features and discovering causal relationships in data make the use of Bayesian networks a good choice for the task of intrusion detection. However, the advantage of being to able to detect previously new attacks comes at a price of high calculation cost. Reducing this cost without negatively

[1] visiting internship student from Simon Fraser University, School of Engineering Science in Vancouver, Canada.

affecting IDS performance is the focus of this paper.

We build upon promising research that Bayesian networks provide an accurate method of classifying data [5], and present a method to improve the efficiency of this type of IDS in such a way that does not require heavy domain knowledge. We use this technique on an IDS that we construct, then verify that it maintains its original high performance level.

The rest of this paper is structured as follows. First, we discuss related work in Section 2 and introduce Bayesian Networks in Section 3. In Section 4 we outline the concept behind our solution and provide a methodology for testing it. Section 5 describes and discusses evaluation results. In Section 6 we present our conclusions and describe future work.

2. Related Work

Although anomaly-based IDSs perform well in detecting new attack variations, when compared to misuse-based IDSs, their deficiencies include:

(1) high false alarm rates,
(2) a lack of usability, and
(3) inefficiency due to high computation costs [6].

The third issue is addressed indirectly by controlling overload in an IDS, as demonstrated in the PacketScore approach [7]. In this solution, when the network is under heavy load, the IDS selectively evaluates high-risk packets. The risk of each packet is determined by a complex scoring system that compares the packet's attributes to their corresponding normal profiles which vary according to the time of day. This solution is effective in eluding overload attacks, whereby an attacker tries to bombard an IDS with so much malicious traffic that it is unable detect the attacker's intended intrusion while it is overwhelmed [8]. However, for PacketScore to be effective, extensive traffic profiling must be done prior to deployment. Packet discarding also has the drawback of reducing the ability to detect attacks with low traffic volume. Our approach does not compromise the detection rate of these attacks and focuses directly on improving calculation speed rather than on overload control as a method to alleviate inefficiency in the anomaly-based IDS.

Favourable results have also been achieved with the introduction of cost-sensitive modeling into the IDS. In cost-sensitive modeling [8], operational cost is one type of cost considered. Operational cost can be modeled by categorizing each predictive feature according to the amount of resources needed to compute it. Combining knowledge of operational cost with consequential costs, an IDS can be implemented so as to not respond to intrusions deemed to be low cost. Alternatively, the IDS can be broken down into multiple IDSs each with a larger subset of predictive features and greater computational cost. A prediction can then be made using the IDS appropriate for the amount of resources available.

Though wide in solution scope, this method requires that various types of costs be carefully determined and incorporated into the model. Unlike the technique we present in this paper, cost-sensitive modeling requires expert knowledge for determining costs and is specific to the environment of each IDS [9]. We also make the assumption that the availability of all required predictive features does not impose constraints on the speed of prediction.

3. Bayesian Networks

We now introduce a tool used for data classification in the field of data mining, which will form the basis of our IDS. This tool, known as a Bayesian network, has been found to achieve results superior to other methods in the construction of an anomaly-based IDS [5].

3.1 Overview

A Bayesian network [10] graphically represents a domain in which there is uncertainty. In a Bayesian network, each random variable in the domain is represented by a node that is part of a directed acyclic graph. A directed edge between a parent and child node represents a relationship in which the parent has a

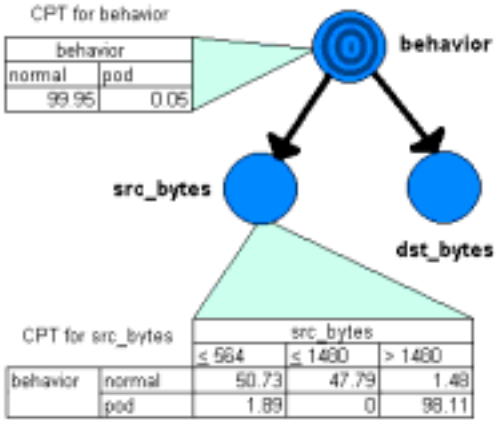

Fig.1: A Bayesian Network and its Conditional Probability Tables (CPTs).

causal effect on the child, while two unconnected nodes are modeled as independent.

Each node contains all the possible states of the random variable that it represents, alongside a

conditional probability table (CPT). The CPT lists the probabilities of the node being in a specific state, given the state of its parents. Figure 1 illustrates a Bayesian network and CPTs for two of the nodes.

3.2 Learning and Data Classification

Given a dataset that is representative of the domain to be modeled, its Bayesian network representation can be constructed using various machine-learning algorithms that learn the probabilities and discover the causal relationships between the nodes. Once constructed, the network can be used to predict the probability of a certain event given the value of other variables. It is often the case that all variables are known except one target variable whose value we want to determine.

Referring to Figure 1, we can infer the probabilities of each of the states (normal, pod) in the target variable "behavior", given the state of its children "src_bytes" and "dst_bytes". By using statistical inference, we can combine our Bayesian network model with information about current variable states in order to classify events based on their most probable outcomes.

4. System Concept and Methodology

A network connection [2] has many possible characteristics (attributes) that can potentially be represented by nodes in a Bayesian network. In addition to describing the connection itself, attributes may also describe a set of connections or traffic on the network. In differentiating a malicious network connection from a normal one, the presence of specific combinations of attribute values signifies an increased likelihood that it is part of an attack. The number of possible attributes that can be used for prediction, however, is endless, and limits to the availability of computer resources dictate the need for lowering the calculation cost of a Bayesian-network-implemented IDS.

A Bayesian network's calculation cost is dependent on the number of nodes and edges it has. The number of edges influences the size of the CPT, thereby influencing the calculation time. When we construct Bayesian networks, the Augmented Markov Blanket algorithm chooses only those nodes pertinent to the prediction outcome [11]. Software implementing this algorithm takes care of choosing the attributes to model, but we would like to further narrow down this subset.

In this section we introduce a metric called information gain, which helps us choose these attributes based on how informative their values are in determining the behaviour of the network connection.

We then present our method of discovering the best choice for maximizing performance.

4.1 Information Gain

Information gain is an evaluation metric of interest in Bayesian networks because it quantifies each attribute's "usefulness" in determining probabilities for the target variable, the attribute we wish to classify. Information gain relates the values of entropy, which measure the level of impurity in a set of samples. When there is at most one class of values present, entropy is at its lowest, and a conditional probability table provides the greatest amount of useful information. Conversely, when entropy is highest, the proportions of all present classes are equal, and the knowledge of that attribute makes no difference in determining the classification of the target variable. If the target value can take on c different values, then the entropy of a set of samples S relative to this c-wise classification is defined as

$$Entropy(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

where p_i is the proportion of S belonging to class i..

The information gain $Gain(S, A)$ of an attribute A relative to S, is defined as

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} - \frac{|S_v|}{|S|} Entropy(S_v)$$

where $Values(A)$ is the set of all possible values for attribute A, and S_v is the subset of S for which attribute A has value v [12].

4.2 Node Reduction Strategy

The use of machine-learning algorithms alone to construct Bayesian network does not provide us with a sufficient means of enhancing the performance of a Bayesian network IDS. Thus, we have developed a technique for reducing the cost of calculation in such an IDS.

Motivation for determining the information gain in an IDS follows naturally from its definition. By selectively removing nodes that provide little or no information for classifying a packet, we can improve the evaluation speed of the IDS. We wish to verify that removing these nodes does not noticeably affect the performance of the IDS and, more generally, that information gain is well-related to the good performance of an IDS.

We investigate these ideas with the following steps:

(1) Construct a Bayesian network from a dataset consisting of multiple attributes and correctly

[2] In this paper, we use "network connection" to describe a TCP connection, UDP stream, or ICMP packet.

labeled data containing both attacks and normal traffic.

(2) Calculate the information gain of each node with respect to the target variable (attack name)

(3) Measure the performance of the IDS (details to be covered in Section 5.2) and remove the least informative node. In some cases, the conditional probability tables must be relearned if removing a causal relationship results in probability tables that are no longer normalized.

(4) With the resulting network, repeat Step 3 until there are no nodes left.

In order to investigate how the order of node removal affects the performance of the IDS, we perform the following variations on the procedure described above, evaluating two extreme cases and a control case:

Case_Low: Remove the node with the *lowest* information gain first.

Case_High: Remove the node with the *highest* information gain first.

Case_Rnd: Remove a random node.

The evaluation speed for each network is also observed in order to consider the optimal balance between speed and correctness.

5. Evaluation

In this section we will describe how the IDS was constructed, investigate the ideas presented above, and evaluate the performance of IDSs utilizing these performance-enhancing techniques.

5.1 Evaluation Environment

To construct and evaluate the performance of our IDS, we used the software BayesiaLab [13] in combination with its Java API for performing inferences.

Using data from the KDD Cup 1999 data mining competition [14], we constructed an IDS using 18 predictive features and one target variable. This variable, "behavior", represents the conditional probability table used to predict whether a connection is normal or it is one of ten different attacks.

Table 1 shows the composition of the data set used for constructing the Bayesian network, or "training data"[3].

Behavior	Training Data Number of instances	% of total
Back	2203	0.45%
Ipsweep	1247	0.25%
Land	21	0.00%
Neptune	107201	21.75%
Nmap	231	0.05%
Normal	97277	19.74%
Pod	264	0.05%
Portsweep	1040	0.21%
Satan	1589	0.32%
Smurf	280790	56.97%
Teardrop	979	0.20%
TOTAL	492842	100.00%

Table.1 Composition of the training data set.

Each instance in the dataset represents one network connection and is comprised of a set of pre-calculated "attributes" (predictive features) which quantify various properties of the connection.

The network structure is then learned from the dataset using the Augmented Markov Blanket algorithm, which constructs Bayesian networks that perform well in regards to both speed and accuracy [11]. Unlike the popular Naïve Bayes algorithm, this algorithm does not assume independence among attributes, which lends itself well to modeling the network domain and providing more accurate predictions. Additionally, after learning the relationships between attributes, it retains only those attributes that contribute to the prediction of the target variable. In our case, the algorithm extracts from the original 19 attributes 13 that match this description.

The resulting Bayesian network is depicted in Figure 2.

5.2 Performance Metrics and Testing Data

The following measures will be used to compare IDS performance:

(1) Correctness – the percentage of correct predictions

(2) False alarm rate – the percentage of predictions incorrectly characterized as attacks

(3) Detection rate – the percentage of attacks that are detected, including those that are classified incorrectly as a different attack

(4) Throughput – the number of predictions performed per second

The data set used for evaluation, or "testing data", is completely independent of the training data set. It has been modified from the original to include only those

[3] This dataset has been modified from the original 10% subset of KDD Cup 1999 training data to include only DoS and Probe type attacks.

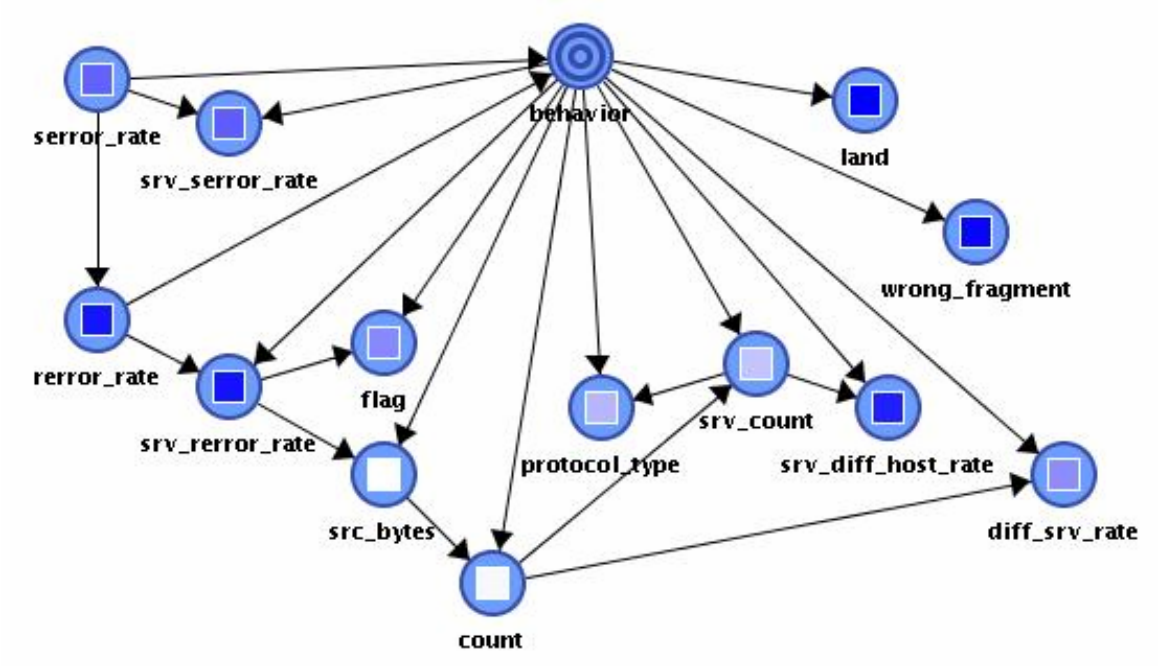

Fig.2: A Bayesian Network IDS capable of detecting 10 types of attacks.

attacks that were "learned" from the training data. The composition of the testing data set is shown in Table 2.

behavior	Testing Data Number of instances	% of total
Back	1098	0.38%
Ipsweep	306	0.11%
Land	9	0.00%
Neptune	58001	20.26%
Nmap	84	0.03%
Normal	60593	21.17%
Pod	87	0.03%
Portsweep	354	0.12%
Satan	1633	0.57%
Smurf	164091	57.32%
Teardrop	12	0.00%
TOTAL	286268	100.00%

Table.2 Composition of the testing data set.

5.3 Results

5.3.1 Correctness

For each of the three cases, the correctness is plotted with respect to the number of remaining nodes in the Bayesian Network, as shown in Figure 3.

In Case_Low, where nodes with low information gain are removed first, correctness remains relatively high even with only four nodes remaining. With four nodes left, correctness drops by 0.39% to 99.34% from an original 99.73% when the IDS is unmodified. In contrast, correctness markedly deteriorates by the removal of the fourth node in Case_High, where nodes with highest information gain are consecutively removed. The case in which nodes are randomly removed, Case_Rnd, is bounded by the two extremes.

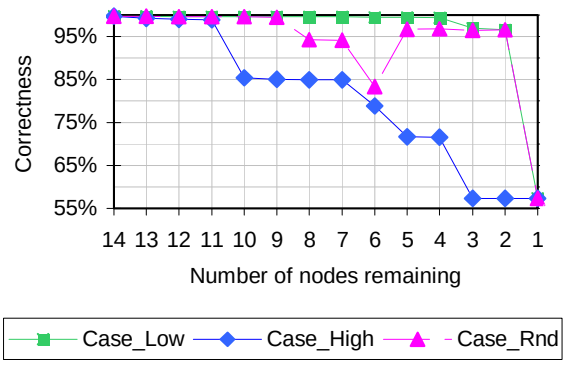

Fig.3 Correctness vs. number of remaining nodes

5.3.2 False Alarm Rate

Figure 4 plots the false alarm rate in relation to the number of remaining nodes.

Again, Case_High exhibits marked deterioration in perform ance beginning at the removal of the fourth node. Case_Low and Case_Rnd are nearly identical and the original false alarm rate decreases from 0.58% to 1.80% when two nodes are left.

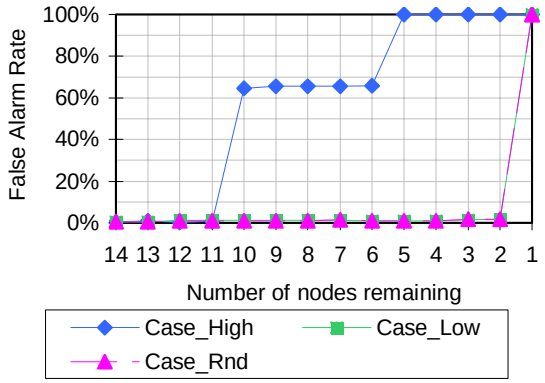

Fig.4 False alarm rate vs. number of remaining nodes

5.3.3 Detection Rate

Depicted in Figure 5 is a graph relating the detection rate and the number of remaining nodes.

In Case_High, the detection rate slowly improves as more nodes are removed, while Case_Low is the opposite. From a detection rate of 99.92%, the detection rate is reduced to 99.60% with 3 nodes remaining.

The random case exhibits sporadic performance, with a sharp decrease in performance for the six-node IDS. This result suggests that the 8th node removed, indicating the error or normal status of the network connection, is of particular importance to the IDS in Case_Rnd. At this point, we note the possibility that performance depends heavily on the network structure.

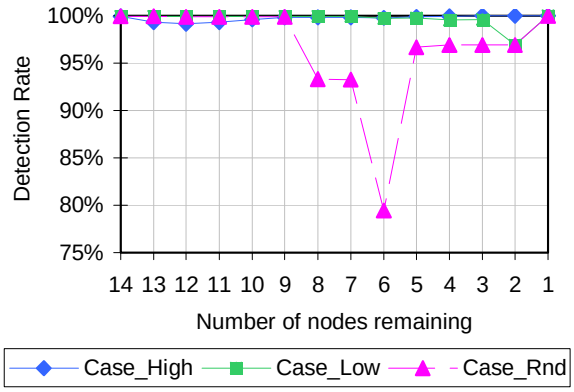

Fig.5 Detection rate vs. number of remaining nodes

5.3.4 Calculation Time and Throughput

Figure 6 shows the relationship between calculation time and the remaining number of nodes. Although results were obtained by removing nodes as in Case_Low, the slight variation in network structure among the various cases should have little effect on calculation time.

Throughput is also plotted in Figure 6.

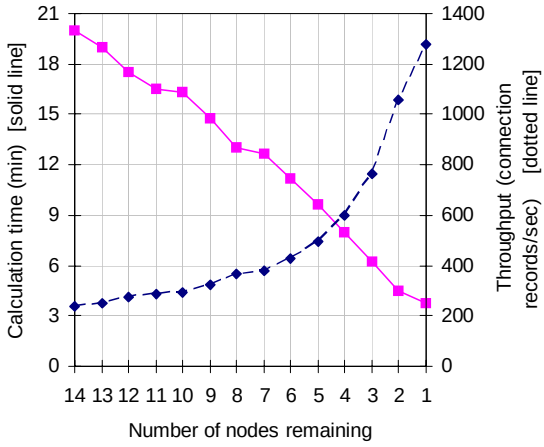

Fig.6 Calculation and throughput vs. number of remaining nodes

As indicated by the solid line, calculation time decreases linearly as the number of nodes decreases. With five nodes remaining, the throughput almost doubles, increasing from 239 to 498 connections per second. Assuming that we deal exclusively with short HTTP connections transferring 2300 bytes per connection, the corresponding bandwidth increases from 4.4 Mbps to 9.1 Mbps. These calculation times were obtained from a PC with the following specifications:

- 1GHz Pentium III processor,
- LI Cache size 16KB,
- L2 Cache size 256KB,
- 384MB RAM, and
- running RedHat Linux 9.

5.4 Considerations

With the potential to double throughput by removing nine nodes, we now consider a general strategy for improving the performance of a Bayesian network-implemented IDS.

We observe that the removal of high information gain nodes have a marked negative effect on the correctness and false alarm rate. Removal of nodes with the lowest information gain has the least detrimental effect on the performance of the IDS, as measured by correctness, false alarm and detection rates.

By removing nodes that do not contribute much to improving the performance of the IDS, we can improve the calculation time. Figure 7 illustrates the tradeoff between correctness and calculation time.

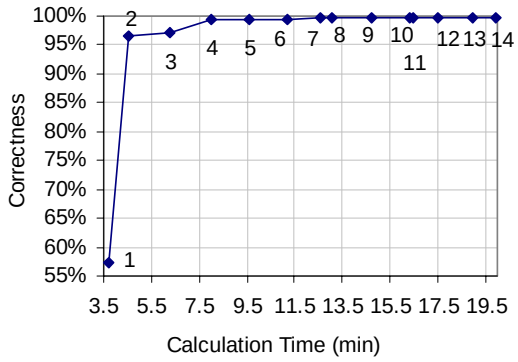

Fig.7 Correctness vs. calculation time. Data labels indicate the number of nodes in the IDS.

As can be derived from the graph, a lower limit of 8 minutes (366 connections per second) is required to maintain a correctness rate above 99%.

In the case where low information gain nodes are removed first, despite a high overall correctness maintained throughout the removal of 10 nodes, one concern is that the removal of a particular node is especially detrimental to the performance of an attack that depends heavily on that node. We analyze this in Figure 8 by plotting the correctness for each attack type as the number of nodes decreases.

It can be observed that the Land attack immediately decreases in correctness after the removal of the first node, also called "land". This Boolean node takes on a truth value whether the source and destination IP are identical, completely characterizing a Land attack. [14].

It should be noted that the low information gain that results in the removal of the land node does not contradict our findings. Since information gain accounts for the relative number of instances of the various attacks, the land node's low information gain can be attributed to the disproportionately small amount of land data used in training the IDS. This result points to a possibility of a performance-predictive metric more useful than information gain, in which accuracy of predictions contributes to scaling each node's information gain.

We also observe from Figure 8 that by reducing the IDS to one with seven nodes, we sacrifice the correctness of the Land and Pod attacks. Due to the simplicity of these attacks, we propose the following alternative methods for detecting them:

(1) Land – hardware pre-filtering of packets in which the source and destination addresses are identical.

(2) Pod – discard fragmented and oversized ICMP packets.

We propose this technique as a general strategy for enhancing the performance of a Bayesian network implemented IDS. If it is observed that some attack can

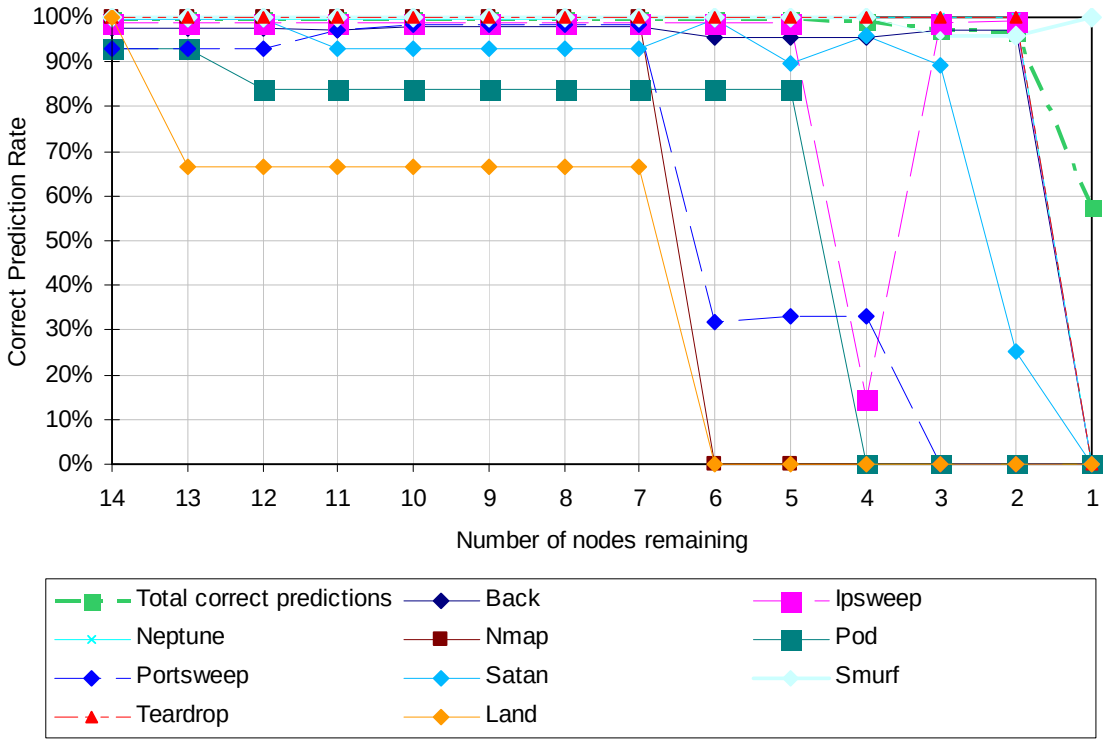

Fig.8: Correctness vs. number of remaining nodes for individual attacks.

be characterized deterministically using a subset of nodes, we may extract those nodes from the IDS provided that they are used exclusively for the prediction of that attack. We can then move the detection of this attack into a firewall implemented in hardware. Shifting the responsibility for detection in this way allows for improvement in performance by node removal.

With the removal of two more nodes to obtain a five-node IDS, we sacrifice the correctness of two more attacks, Nmap and Portsweep. We notice that removing the "flag" node in the seven-node IDS causes a sharp decrease in their correctness. The "flag" attribute represents the error state of the network connection, and has direct relevance to the Nmap and Portsweep attacks since both try to gather information about networks by listening to the responses they receive from live hosts. The "flag" node's early removal as a result of low information gain is also due to the weak presence of the Nmap and Portsweep attacks in the training data. Further investigation is required before we can determine an alternative strategy for detecting these attacks while maintaining high performance.

Finally, a limitation of the current IDS is the use of per-connection data as opposed to per-packet data in our analysis. Inability to make a prediction until a connection is closed can hinder the ability to stop the intrusion while it is in progress. For this reason, a packet-based approach would be of benefit.

6. Conclusions and Future Work

In this paper we have proposed a technique for removing nodes in a Bayesian network to produce high-speed detection of malicious traffic without great impact on the performance of an IDS.

We found that the strong relationship between information gain and correctness allows us to remove nodes with low information gain and minimally affect the overall performance of the IDS, as measured by correctness, false alarm rate, and detection rate.

In future work, we aim to move away from the connection-based data, as presented in this paper, to a packet-based approach which could allow for earlier detection along with better control of intrusions once they have been detected.

Finally, there is a possibility that a more informative performance predictor than information gain can be found if we factor in the correctness of each type of attack.

Acknowledgement We thank Hitoshi Uematsu, Masahiro Morikura, Osamu Ishida, and Haruhisa Ichikawa of NTT Network Innovation Laboratories for their support of the research described in this paper. We would also like to acknowledge Yasuji Matsuge and Masaki Hiratsuka for their help and invaluable discussion input, and Silvia Yuen for her contributions in the early stages of this research.

References

[1] Snort. http://www.snort.org.
[2] Dragon. http://www.enterasys.com/products/ids.
[3] RealSecure. http://www.iss.net/products_services/enterprise_protection.
[4] Werrett, J., 2003, "Review of Anomaly-based Network Intrusion Detection", http://www.csse.uwa.edu.au/+werrej01/docs/review.pdf.
[5] Amor, N.B., Benferhat, S., Elouedi, Z., 2004, "Naïve Bayes vs Decision Trees in Intrusion Detection Systems", 2004 ACM Symposium on Applied Computing, 1-58113-812-1/03/2004.
[6] Lee, W., et. al, 2001, "Real Time Data Mining-based Intrusion Detection", Proc. DARPA Information Survivability Conference and Exposition, 1(12—14), pp. 89—100.
[7] Kim, Y., et. al, 2004, "PacketScore: Statistics-based Overload Control against Distributed Denial-of-Service Attacks", *IEEE INFOCOM 2004*, 0-7803-8356-7/04.
[8] Lee, W., et. al, 2002, "Toward Cost-Sensitive Modelling for Intrusion Detection and Response", *Journal of Computer Security*, 10(1,2).
[9] Stolfo, S. J., et. al, 2001, "Data Mining-based Intrusion Detectors: An Overview of the Columbia IDS Project", *SIGMOD Record*, 30(4), pp. 5—14.
[10] Kruegel, C., et. al, 2003, "Bayesian Event Classification for Intrusion Detection", *Proc. 19th Annual Computer Security Applications Conference*, pp. 14—23.
[11] Madden, M. G., 2002, "Evaluation of the Performance of the Markov Blanket Bayesian Classifier Algorithm", Technical Report No. NUIG-IT-0011002, *National University of Ireland, Department of Information Technology*, Galway.
[12] Bao, H.T., "Constructing Decision Trees". ttp://www.netnam.vn/unescocourse/knowlegde/3-2.htm.
[13] BayesiaLab, http://www.bayesia.com.
[14] Hettich, S. and Bay, S. D., 1999, KDD Cup 1999 Data, The UCI KDD Archive, *University of California, Department of Information and Computer Science*, Irvine, CA. http://kdd.ics.uci.edu