Towards fast detecting intrusions: using key attributes of network traffic

Wei Wang[1, 2], Sylvain Gombault[3], Thomas Guyet[1],

[1]Dream Team, IRISA, France
[2]AxIS Team, NRIA Sophia Antipolis, France
[3]Department of Networks, Security and Multimedia, TELECOM Bretagne, France

{wei.wang, thomas.guyet}@irisa.fr, sylvain.gombault@telecom-bretagne.eu

***Abstract*—Extracting attributes from network traffic is the first step of network intrusion detection. However, the question of which or what attributes are most effective for the detection still remains. In this paper, we employed information gain, wrapper with Bayesian Networks (BN) and Decision trees (C4.5) respectively to select key subsets of attributes for network intrusion detection based on KDD Cup 1999 data. We then used the selected 10 attributes to detect DDoS attacks in the real environments. The empirical results based on DDoS attack data collected in the real world as well as KDD Cup 1999 data show that only using the 10 attributes, the detection accuracy almost remains the same or even becomes better compared with using all the 41 attributes with both BN and C4.5 classifiers. Using a small subset of attributes also improves the efficiency in terms of attribute forming, models training as well as intrusion detection.**

Keywords-Intrusion detection; DDoS attack detection; attribute selection; Bayesian networks; C4.5; information gain

I. INTRODUCTION

Network-borne attacks are currently major threats to information security. As an important technique in the defense-in-depth network security framework, intrusion detection has become a widely studied topic in computer networks in recent years [1]. In general, the techniques for intrusion detection fall into two major categories: signature-based detection and anomaly detection. Signature-based detection (e.g., Snort [2] and IDIOT [3]) identifies malicious behavior by matching it against pre-defined description of attacks. Anomaly detection [4], on the other hand, defines a profile of a subject's normal activities and attempts to identify any unacceptable deviation as possibly the result of an attack.

Intrusion Detection Systems (IDS) can also be categorized as host-based IDSs and network-based IDSs according to the target environment for detection. Host-based IDSs usually monitor the host system behavior by examining the information of the system, such as CPU time, system calls and command sequences. Examples are [5-8]. Network-based IDSs, on the other hand, monitor network behavior usually by examining the content (e.g., payload) as well as some statistical attributes of network traffic. In 1999, Lee et al. [9-10] constructed 41 attributes from raw traffic data (i.e., tcpdump files) to build classification models for network based intrusion detection. The raw traffic data was collected at MIT Lincoln Laboratory for the 1998 DARPA Intrusion Detection Evaluation program [11]. The 41 attributes have been shown effective for network intrusion detection [9-10] and the attribute sets of the network traffic have also been used as KDD Cup 1999 data (The 1999 Knowledge Discovery and Data Mining Tools Competition) [12]. Lee et al. [9-10] used *Ripper* to mine some detection rules from the attribute sets and to build misuse detection models. Eskin et al. [13] used unsupervised methods, namely, cluster based estimation, k-Nearest Neighbor (kNN) and one class Support Vector Machines (SVM) for network intrusion detection. Jin et al. [14] utilized the covariance matrices of sequential samples to detect multiple network attacks. Katos [15] evaluated cluster, discriminant and logit analysis on the same KDD Cup 1999 data for network intrusion detection. Shyu et al. [16] proposed a Principal Component Classifier (PCC) for network intrusion detection. They measured the Mahalanobis distance of each observation from the center of the data for anomaly detection. S. Mukkamala et al. [17] evaluated performance of Artificial Neural Networks (ANNs), SVM and Multivariate Adaptive Regression Splines (MARS) on KDD Cup 1999 data for network intrusion detection. In our previous work [18-20], we used Principal Component Analysis (PCA) and kNN for network intrusion detection and identification. Bouzida et al. [21-22] also employed PCA, neural networks and enhanced decision trees for network anomaly intrusion detection based on the same KDD Cup 1999 data set.

Data involved in current computer networks increases very fast and is naturally massive. In experiments carried out by MIT Lincoln Lab for the 1998 DARPA evaluation [11], for example, network traffic over 7 weeks contains four gigabytes of compressed binary tcpdump data that were processed into about five million connection records. A practical IDS, therefore, should have the capacity of fast processing large amounts of network data so that actions for response can be taken as soon as possible before substantial damage is done. Most existing network intrusion detection methods [9-10, 13-22] detect intrusions by using all the 41 attributes constructed from network traffic data. However, some of the attributes may be redundant or even noise and therefore decrease the detection system's performance, e.g., decrease the detection

accuracy while increase system's overload. Empirical evidence from the attribute selection literature also shows that redundant information as well as irrelevant attributes should be eliminated for efficient classification tasks [23]. Sung and Mukkamala [24] used ANN and SVM to find some important attributes based on the performance comparison. For example, an attribute is identified as important if the detection accuracy decreases and/or computation time increases after the attribute is deleted from the training set. In this paper, we used different criteria to select key attributes. Filter (e.g., Information Gain) and Wrapper (with Bayesian Networks and decision trees algorithms) based attribute selection methods are used to select some key subsets from the 41 attributes. The subsets of attributes are then used for fast intrusion detection. This largely simplifies the detection problem because only a smaller set of attributes is required to extract from raw network traffic and to process in detection step. The empirical results based on KDD Cup 1999 data show that only using 10 attributes, the detection accuracy almost remains the same or even becomes better compared with using all the 41 attributes with both Bayesian Networks (BN) and decision trees (C4.5) classifiers.

Distributed Denial-of-Service (DDoS) attack is one of the major threats in current computer networks. However, DDoS attacks are difficult to quickly detect due to complex attack behaviors and a very large amount of data involved during the attacks. In the real network environment, we collected various DDoS attack tools to implement attacks and collect a large set of attack data. A normal data set is also collected during normal usage in the same network. Instead of using all the 41 attributes, we made programs to extract from raw network traffic only 10 key attributes that were identified for DoS attacks based on KDD Cup 1999 data. Based on the 10 attributes we extracted, we employ BN and C4.5 to build the detection model and perform intrusion detection. Experimental results based on the real DDoS attack data show that using a small subset of attributes improves the efficiency in terms of attribute forming, models training as well as intrusion detection. The detection models are also very effective to detect DDoS attacks.

The remainder of this paper is organized as follows. Section II describes the attributes selection schemes as well as intrusion detection methods. The experiments based on KDD Cup 1999 data are given in detail in Section III. We describe the detection of DDoS attacks in real computing environments in Section IV. Concluding remarks follow in Section V.

II. ATTRIBUTES SELECTION AND INTRUSION DETECTION SCHEMES

A. Attributes selection schemes

One of the central problems in intrusion detection is identifying a representative set of attributes from which to construct a classification model. Attribute selection algorithms fall into two broad categories: the filter model or the wrapper model [23, 25]. The filter model relies on general characteristics of the training data to select some attributes without involving any learning algorithm. The wrapper model, on the other hand, requires one predetermined learning algorithm in attribute selection and uses its performance to evaluate and determine which attributes are selected. As for each new subset of features, the wrapper model needs to learn a classifier. It tends to find attributes best suited to the predetermined learning algorithm resulting in superior learning performance, but it also tends to be more computationally expensive than the filter model [23, 25]. To facilitate comparison, in this paper, we use both kinds of attribute selection scheme, namely, information gain based filter model and wrapper based model. The attribute selection schemes are shown in Fig. 1.

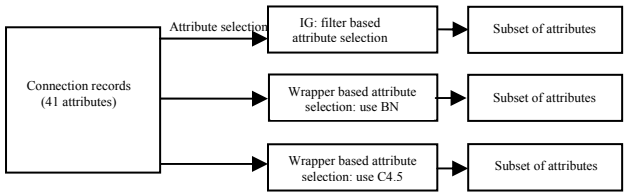

Fig.1: Attribute selcetion schemes

1) Information Gain based attribute selection

The information gain based attribute selection method is very easily accessible. The information gain of a given attribute X with respect to the class attribute Y is the reduction in uncertainty about the value of Y, after observing values of X. It is denoted as $IG(Y \mid X)$. The uncertainty about the value of Y is measured by its entropy defined as

$$H(Y) = -\sum_i P(y_i)\log_2(P(y_i)) \quad (1)$$

where $P(y_i)$ is the prior probabilities for all values of Y. The uncertainty about the value of Y after observing values of X is given by the conditional entropy of Y given X defined as

$$H(Y \mid X) = -\sum_j P(x_j)\sum_i P(y_i \mid x_j)\log_2(P(y_i \mid x_j)) \quad (2)$$

where $P(y_i \mid x_i)$ is the posterior probabilities of Y given the values of X. The information gain is thus defined as

$$IG(Y \mid X) = H(Y) - H(Y \mid X) \quad (3)$$

According to this measure, an attribute X is regarded more correlated to class Y than attribute Z, if $IG(Y \mid X) > IG(Y \mid Z)$. By calculating information gain, we can rank the correlations of each attribute to the class and select key attributes based on this ranking.

2) Wrapper based attribute selection

For wrapper based attribute selection, we use Bayesian networks and decision trees (C4.5) as classifiers that will be described in next subsections. The classifiers used for the wrapper based attribute selection, in fact, are usually employed as intrusion detection schemes in detection step because the classifiers used during attribute selection step always best suit for classification.

B. Intrusion detection schemes

1) Bayesian networks based intrusion detection

A Bayesian network is used to model a domain containing uncertainty in some manner [26-27]. It is a probabilistic graphical model that represents a set of variables and their probabilistic independencies. In intrusion detection, for example, a Bayesian network could represent the probabilistic relationships between attribute sets and types of intrusions. Given an attribute vector of an instance, the Bayesian network can also be used to compute its probabilities of the presence of various classes (normal or individual type of intrusions).

Formally, Bayesian networks are Directed Acyclic Graphs (DAG) whose nodes represent variables, and whose arcs encode conditional dependencies between the variables. Each node contains the states of the random variable that it represents and a Conditional Probability Table (CPT). The CPT of a node contains probabilities of the node being in a specific state given the states of its parents. The parent-child relationship between nodes in a Bayesian network indicates the direction of causality between the corresponding variables. That is, the variable represented by the child node is causally dependent on the ones represented by its parents. There are some efficient algorithms that can be used to perform inference and learning in Bayesian networks [26-27].

Suppose there is an arc from node A to another node B, A is called a parent of B, and thus B is a child of A. The set of parent nodes of a node X_i is denoted by $parents(X_i)$. A directed acyclic graph is a Bayesian Network relative to a set of variables if the joint distribution of the node values can be written as the product of the local distributions of each node and its parents:

$$P(X_1, \cdots, X_n) = \prod_{i=1}^{n} P(X_i \mid parents(X_i)) \qquad (4)$$

Given a training set S, learning a Bayesian network is to find a network that best matches S. The learned network represents an approximation to the probability distribution governing the training set. For classification, given a test data vector represented with attributes, we use this network to compute its probability based on which for classification.

2) Decision tree based intrusion detection

The decision tree models are found to be very useful in the domain of data mining since they obtain reasonable accuracy and they are relatively inexpensive to compute. Decision tree classifiers are based on the "divide and conquer" strategy to construct an appropriate tree from a given learning set S containing a set of labeled instances. As a well known and widely used algorithm, C4.5 algorithm developed by Quinlan [28] generates accurate decision trees that can be used for effective classification.

C4.5 builds decision trees from a set of training data also with the concept of information entropy. It uses the fact that each attribute of the data can be used to make a decision that splits the data into smaller subsets. C4.5 examines the information gain ratio (can be regarded as normalized Information Gain) that results from choosing an attribute for splitting the data. The attribute with the highest information gain ratio is the one used to make the decision. Given a learning set S and a non class attribute X, the *Gain Ratio* is defined as:

$$IGR(S \mid X) = \frac{IG(S \mid X)}{-\sum_i \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}} \qquad (5)$$

where S_i is the subset of S for which attribute X have a value and $|S|$ is the number of instances in S.

The decision trees are constructed as a set of rules during learning phase. It is then used to predict the classes of new instances based on the rules.

III. Experiments on KDD Cup 1999 data

In order to validate the proposed methods, firstly we used KDD Cup 1999 data for selection of key attributes as it is considered as a benchmark data set and has been widely used by many other methods. We then used the selected attributes for detection of DDoS attacks in real environments in our department.

A. Data sets

The original data contains traffic in a simulated military network that consists of hundreds of hosts. The data includes 7 weeks of training set and 2 weeks of test set that were not from the same probability distribution as the training set. Since the probability distribution is not the same, in our experiments, we only use the training set and sample one part of the data for training and another different part of the data for testing. The raw training set of the data contains about 4 gigabytes of compressed binary tcpdump data of network traffic and it was pre-processed into about 5 million connection records by Lee et al. [9-10] as KDD Cup 1999 data [12]. A connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows from a source IP address to a target IP address under some well defined protocol [12]. In the data set, each network connection is labeled as either normal, or as an exactly one specific kind of attack.

In KDD Cup 1999 data, each network connection is represented by 41 attributes [12]. These attributes are divided into three groups: basic attributes of individual TCP connections, traffic attributes and content attributes within a connection suggested by domain knowledge. The attributes are listed in Table I and meaning of each attribute can be found in [12]. There are 494,021 connection records in the training set in which 97,278 are normal and 396,744 are attacks. There are 22 types of attacks in total in the data set and these attacks fall into one of 4 categories: DoS: denial-of-service (e.g., teardrop); PROBE: surveillance and other probing (e.g., port scanning); R2L: unauthorized access from a remote machine (e.g., password guessing) and U2R: unauthorized access to local superuser (root) privileges by a local unprivileged user (e.g., buffer overflow attacks).

B. *Attributes selection results*

As different categories of attacks may have different key subsets of attribute, we conduct four experiments to investigate which subset of attribute is more suitable for detecting individual category of attacks. The distribution of individual attack data is not balanceable, for example, smurf attack (DoS) has 280,790 instances while spy attack (R2L) has only 2 examples. In the experiments, we then assign the training and test sets based on the distribution of the attack instances. The data used for training and testing is shown in Table II (the attack data in bold font only has few instances).

TABLE I. THE MAPS BETWEEN THE ATTRIBUTES AND THE NUMBER USED IN THE PAPER

No.	Network attributes	No.	Network attributes	No.	Network attributes	No.	Network attributes
1	duration	12	logged_in	23	count	34	dst_host_same_srv_rate
2	protocol_type	13	num_compromised	24	srv_count	35	dst_host_diff_srv_rate
3	service	14	root_shell	25	serror_rate	36	dst_host_same_src_port_rate
4	flag	15	su_attempted	26	srv_serror_rate	37	dst_host_srv_diff_host_rate
5	src_bytes	16	num_root	27	rerror_rate	38	dst_host_serror_rate
6	dst_bytes	17	num_file_creations	28	srv_rerror_rate	39	dst_host_srv_serror_rate
7	land	18	num_shells	29	same_srv_rate	40	dst_host_rerror_rate
8	wrong_fragment	19	num_access_files	30	diff_srv_rate	41	dst_host_srv_rerror_rate
9	urgent	20	num_outbound_cmds	31	srv_diff_host_rate		
10	hot	21	is_host_login	32	dst_host_count		
11	num_failed_logins	22	is_guest_login	33	dst_host_srv_count		

TABLE II. DATA USED FOR TRAINING AND TESTING IN THE EXPERIMENTS

Category	Training set (randomly selected)	Test set (randomly selected from the rest of the data)
DoS (391,458)	Normal 40,000, smurf 10,000, neptune 5000, back 1000, land 10, pod 100, teardrop 400	normal 40,000, smurf 10,000, neptune 5000, back 1203, land 11, pod 164, teardrop 579
Probe (4107)	Nomal 40,000, satan 800, portsweep 500, nmap 110, ipsweep 600	normal 40,000, satan 789, portsweep 540, nmap 121, ipsweep 647
R2L (1126)	Normal 40,000, ftp_write 4, guess_passwd 23, imap 7, multihop 3, warezclient 520, warezmaster 10, **phf 4**, **spy 2**	Normal 40,000, ftp_write 4, guess_passwd 30, imap 5, multihop 4, warezclient 500, warezmaster 10
U2R (52)	Normal 40,000, buffer_overflow 15, rootkit 4, loadmodule 4	Normal 40,000, buffer_overflow 15, rootkit 6, loadmodule 5, **perl 3**

We performed information gain based and wrapper based attribute selection methods on each training sets. For wrapper based methods, the search method is chosen as *best first*. The selected attributes with different methods are listed in Table III. The attributes that at least two methods simultaneously select are in bold font. From table III, it is seen that some key features remains the same whatever the attack categories are. For example, attribute 5 (number of data bytes from source to destination) always ranks as very important for detection of all categories of attack. Generally, the basic attributes (attribute 1-6) are important for detection of all attack categories. For different attack categories, some key attributes are not the same. This is easily understood because different types of attack have their own patterns. For example, attribute 1 (duration: number of seconds of the connection) is an important pattern for R2L and U2R attacks while it is irrelevant for DoS and Probe attacks. Usually it requires a long time for R2L and U2R attacks to login to the system to guess the passwords or compromise some system's vulnerabilities to have the root privilege in a connection. DoS and Probe attacks, however, attack a system usually by sending a large mount of packets in a short time and thus the duration is very short. For each attack category, some attributes are always shown important whatever the attribute selection methods used. For example, attribute 23 (count: number of connections to the same host as the current connection in the past two seconds) is an important pattern for DoS attacks. This is because DoS attacks are to make a computer resource unavailable usually by flooding a network or other means with a very large mount of connections to the same host in a very short time.

For Information gain based attribute selection, the attributes selected have a ranking. For example, in table III for DoS attack detection, attribute 5 is more important than attribute 23 which is also more important than attribute 3, and so on. Based on the results in Table III, we use the most 10 common attributes that different methods simultaneously selected to form the key set of attributes shown in Table IV for detection of different categories of attacks.

TABLE III. SELECTED ATTRIBUTES FOR DETECTION OF INDIVIDUAL ATTACK CATEGORY WITH DIFFERENT METHODS

Attacks	Methods	Attributes selected
DoS	IG	**5, 23, 3**, 24, **6**, 2, 36 (ranking)
	Wrapper (BN)	4, 5, 8, 10, **13, 23**, 37
	Wrapper (C4.5)	**3, 5**, **6**, **13, 23**
Probe	IG	**5, 3**, 6, **35**, 33, 34, **4**, 27, 23 (ranking)
	Wrapper (BN)	**3, 4, 5**, **29**, 32, **35**
	Wrapper (C4.5)	**5, 29**, 30, **35**, 39, 40
R2L	IG	**5, 3, 6**, 33, 36, 10, 37, 24, **1**
	Wrapper (BN)	**1, 5, 6**, 22, 23, 32
	Wrapper (C4.5)	**1, 3, 5, 6,** 12, 31
U2R	IG	3, 33, **13, 14, 1**, 10, **5**, 17, 32, **36** (ranking)
	Wrapper (BN)	**1**, 2, **5, 14, 36**
	Wrapper (C4.5)	**1, 13, 14, 32**

TABLE IV. SELECTED ATTRIBUTES FOR INDIVIDUAL ATTACK CATEGORY

Attacks	Attributes selected
DoS	3, 4, 5, 6, 8, 10, 13, 23, 24, 37
Probe	3, 4, 5, 6, 29, 30, 32, 35, 39, 40
R2L	1, 3, 5, 6, 12, 22, 23, 31, 32, 33
U2R	1, 2, 3, 5, 10, 13, 14, 32, 33, 36

To evaluate the effectiveness of the selected attributes, we compare the detection results using the selected 10 attributes with the results using all the 41 attributes based on the same test data. In the experiments, we use Detection Rates (DR), calculated as the percentage of intrusions detected, and False Positive Rates (FPR), calculated as the percentage of normal connections falsely classified as intrusions, as criteria for evaluation. Bayesian networks and C4.5 are employed for detection based on the test sets of individual category of attacks and the contrastive results are shown in Table V.

TABLE V. RESULTS COMPARISON USING 41 ATTRIBUTES AND 10 ATTRIBUTES

Attacks	Methods	Using 41 attributes		Using 10 attributes	
		DR (%)	FPR(%)	DR (%)	FPR (%)
DoS	BN	98.73	0.08	**99.88**	**0**
	C4.5	99.96	0.15	99.87	**0.14**
Probe	BN	92.89	6.08	82.93	**3.06**
	C4.5	82.59	0.04	**82.88**	0.05
R2L	BN	92.22	0.33	89.33	**0.32**
	C4.5	80.29	0.02	**87.34**	**0.01**
U2R	BN	75.86	0.29	65.5	**0.12**
	C4.5	24.14	0	**24.14**	**0**

In general, the detection results are ideal when the detection rates are very high while the false positive rates are very low. From Table V, it is seen that the models detect high percentage of the DoS, Probe and R2L attacks but not as effective for U2R attacks. This is consistent with the results of many other papers because the behavior of U2R attacks is very similar with that of normal operations. From Table V, we can also see that the detection results only using the 10 attributes almost remain the same or even become better than those using all the 41 attributes. This shows that many of the 41 attributes are irrelevant and only a smaller set of attributes is required to extract from raw network traffic for detection of individual attacks. In next section, we thus only extract 10 attributes from network data for DDoS attack detection in actual computer networks.

IV. DETECTING DDOS ATTACKS IN REAL ENVIRONMENTS BASED ON THE SELECTED 10 ATTRIBUTES

Distributed Denial-of-Service (DDoS) attack is one of the major threats in current computer networks. It is an attempt to make a computer resource unavailable to the intended users. The means to, motives for, and targets of a DoS attack may vary, but it generally consists of the concerted, malevolent efforts of a person or persons to prevent an internet site or service from functioning efficiently. DoS and DDoS attacks are difficult to detect with high accuracy due to the sheer number of ways in which they can be executed, the increasingly sophisticated attack methods, the growing range of systems targeted and the increasing data involved. As DDoS attacks are very harmful to the networks, an effective DDoS attack detection system should be capable of detecting the attacks very fast for quick reaction. Instead of using a lot of attributes that may decrease the efficiency of detection process, we only use 10 key attributes described in above Section for DDoS attack detection in real environments.

A. Data sets and intrusion detection

As important work on our DDoS attack analysis and detection project in Institute Telecom, we collected some major DDoS attack tools and experienced them in our laboratory to collect a set of DDoS attack network traffic. The attack tools are Trinoo, TFN, Stacheldraht, TFN2K, and Mstream. Using these tools, we implement DDoS attacks with ICMP flood, SYN flood, UDP flood, Steam (TCP-ACK flood) and Smurf style attacks. A large set of normal as well as DDoS attack network traffic are then collected for analysis[1].

One of the difficulties for detecting DDoS attack is to extract and select the most important attributes that represent attack behaviors to clearly distinguish them from normal activities. Since the programs for forming attributes from raw network traffic are not available, we have written the different programs (in C++) that transform tcpdump traffic into connection records with only 10 key attributes for detecting DDoS attacks as well as with all the 41 attributes for results comparison. In the experiments we used for Bayesian networks and C4.5 to build the models and detect DDoS attacks. The detection step is described in Fig. 2.

In the experiments, we randomly selected 30,000 normal connections and 36,380 DDoS attack connections to form the training set. For test set, we randomly selected 30000 different normal connections and different 33,900 DDoS attack connections. The training and test set were selected also according to the distributions of different kinds of DDoS attacks.

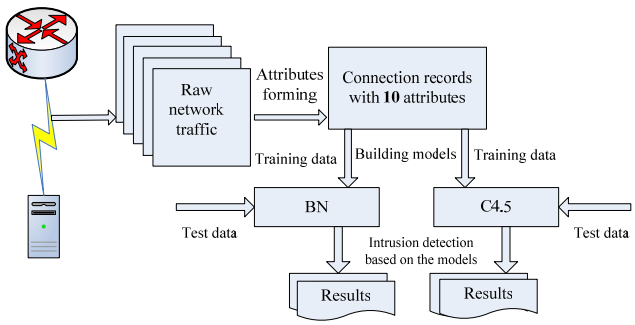

Fig. 2. Attributes selection and detection of DDoS attacks

B. Results comparison

Using all the 41 attributes and only using 10 attributes respectively, we compare the experiment results in the following three aspects based on the order of intrusion detection steps. First, we compare CPU time used for extracting attributes from raw network traffic. Second, we

[1] The data is available upon request to the third author.

compare the CPU time used for training and detection. Third, we compare the detection results. The experiments are performed in a system with 2.66 G Hz dual core CPU and 3.5G RAM memory. The CPU time used for extracting attributes is shown in Table VI and the time used for training the models and detecting intrusions is given in Table VII. Table VIII summarizes the testing results with Bayesian Networks and C4.5 based on only 10 attributes as well as on all the 41 attributes.

TABLE VI. TIME USED DURING ATTRIBUTES FORMING

No. of connections	Time used (s)	
	41 attributes	10 attributes
30,000	2043	**289**

TABLE VII. TIME USED DURING TRAINING AND DETECTION

Methods	41 attributes		10 attributes	
	Training(s)	Detection(s)	Training(s)	Detection (s)
BN	4.42	0.9	**0.7**	**0.2**
C4.5	15.27	0.9	**1.03**	**0.2**

TABLE VIII. DETECTION RESULTS USING 41 ATTRIBUTES AND 10 ATTRIBUTES

Methods	41 attributes		10 attributes	
	DR (%)	FPR (%)	DR (%)	FPR (%)
BN	99.03	1.53	**99.49**	1.92
C4.5	99.80	0.26	**99.90**	0.34

Attribute extraction is the first step for intrusion detection. From Table VI we find that using a smaller set of attributes saves a lot of computation in the attribute forming step. Only using the 10 attributes required less time for both training and detection based on the results shown in Table VII. From Table VIII, it is seen that the detection results based on the real networks are consistent with those based on the KDD Cup 1999 data for DDoS detection. The detection results using only 10 attributes remain the same or even become better than the results using all the 41 attributes. All these are towards fast network intrusion detection for quick response against various DDoS attacks.

V. CONCLUDING REMARKS

Data for intrusion detection has become increasingly larger in both number of instances and number of attributes. Selection of a key subset of attributes is thus very essential in reducing dimensionality, removing irrelevant data, reducing low use of resources and increasing detection accuracy for quick and effective response against attacks. In this paper, instead of using all the 41 attributes that most related papers used before, we select a key subset of attributes for intrusion detection based on several attribute selection schemes, namely, information gain and wrapper with Bayesian networks and with decision trees (C4.5) methods. Based on KDD Cup 1999 data, we selected subsets of key attributes for each attack category, DoS, Probe, R2L and U2R. The detection accuracy remains the same or even has some improvement based on only 10 attributes comparing to using all the 41 attributes. Using fewer attributes, therefore, cannot only enhance detection accuracy, but also improve detection efficiency as smaller data is required to process.

Distributed Denial of Service (DDoS) attacks are a major threat to the stability of the Internet. However, DDoS attacks are difficult to quickly detect as the data involved is usually very large and the attack behaviors frequently vary. In order to analyze and detect DDoS attacks in real computing environments, we collected a large data set of network traffic by implementing various DDoS attacks. We made some programs to extract only 10 attributes from the raw network traffic for representing each network connection. Bayesian networks and decision trees are then used for detecting intrusions. Experimental results show that with only 10 attributes, the detection rates almost remain the same or are even better than using all the 41 attributes. The detection rates achieve as high as 99.9% with false positive rate as 0.34%. The experimental results also show that using a smaller set of attributes largely reduce the time required for attribute forming, models training and intrusion detection. The small set of attributes thus can be regarded as a useful reference for further analysis and detection of DDoS attacks.

For our future work, we are developing an online self-adaptive intrusion detection model that includes two modules. The first module is to extract several key attributes from raw network traffic for fast attack detection. We plan to use some attributes that can be formed before a connection is finished for real-time intrusion detection. The other module is to upgrade the detection model dynamically and automatically for addressing the concept drift problem.

ACKNOWLEDGMENTS

For IRISA/INRIA part, the research in this paper was supported by SéSur (Sécurité et Surveillance dans les Flots de Données) project. For TELECOM Bretagne, the research in this paper was supported by French Ministry of Research (CNRS ACI-SI), Dependable Anomaly Detection with Diagnosis (DADDi) project and by Institute TELECOM, DDOS Detection and passivation project. The authors thank Dr. René Quiniou and Prof. Marie-Odile Cordier for the valuable discussions and comments.

REFERENCES

[1] W. Lee and D. Xiang, "Information-theoretic measures for anomaly detection". Proceedings of 2001 IEEE Symposium on Security and Privacy, pp. 130-143, 2001.

[2] J.Beale, Caswell (Editor), “Snort 2.1 Intrusion Detection (Second Edition)”. Syngress, 2004.

[3] S.Kumar, E.H.Spafford, “A Software architecture to support misuse intrusion detection”, Proceedings of the 18th National Information Security Conference, pp.194-204, 1995.

[4] D. E. Denning, “An intrusion-detection model”. IEEE Transactions on Software Engineering, vol. 13, no.2, pp. 222-232, 1987.

[5] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, “A sense of self for Unix processes”. Proceedings of the 1996 IEEE Symposium on Security and Privacy, pp. 120-128, 1996.

[6] C. Warrender, S. Forrest and B. Pearlmutter, “Detecting intrusions using system calls: alternative data models,” Proceedings of 1999 IEEE Symposium on Security and Privacy, pp.133-145, 1999.

[7] W. Wang, S. Gombault, “Distance measures for anomaly intrusion detection”. Proceedings of 2007 internatinal conference on security and management, SAM'07, pp. 15-21, Las Vegas, NV, June 2007.

[8] W. Wang, X. Guan, X. Zhang, and L. Yang, “Profiling program behavior for anomaly intrusion detection based on the transition and frequency property of computer audit data”. Computers & Security, Elsevier, vol. 25, no 7, pp. 539-550, 2006.

[9] W. Lee, S. Stolfo, K. Mok, “A Data Mining Framework for Building Intrusion Detection Models”, Proceedings of the 1999 IEEE Symposium on Security and Privacy, pp. 120-132, 1999.

[10] W. Lee, S. Stolfo, “A framework for constructing features and models for intrusion detection systems”, ACM Transactions on Information and System Security, vol. 3, no. 4, pp.227-261, 2000.

[11] MIT Lincoln Laboratory-DARPA Intrusion Detection Evaluation, http://www.ll.mit.edu/IST/ideval/docs/docs_index.html, 1999.

[12] KDD Cup 1999 data (Computer network intrusion detection): http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[13] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, S. Stolfo, “A Geometric framework for unsupervised anomaly detection”, Applications of Data Mining in Computer Security. Kluwer Academics, 2002.

[14] S. Jin, D. Yeung, X. Wang, “Network intrusion detection in covariance feature space”, Pattern Recognition, vol. 40, no. 8, pp. 2185-2197, 2007.

[15] V. Katos, “Network intrusion detection: Evaluating cluster, discriminant, and logit analysis”, Information Sciences, vol. 177, no. 15, pp. 3060-3073, 2007.

[16] M. Shyu, S. Chen, K. Sarinnapakorn, L. Chang, "Principal Component-based Anomaly Detection Scheme", Foundations and Novel Approaches in Data Mining, pp. 311-329, Springer-Verlag, Vol. 9, 2006.

[17] S. Mukkamala, A. H. Sunga, A. Abrahamb, “Intrusion detection using an ensemble of intelligent paradigms”, Journal of Network and Computer Applications, vol. 28, no. 2, pp. 167-182, 2005.

[18] W. Wang, X. Guan and X. Zhang, "Processing of Massive Audit Data Streams for Real-Time Anomaly Intrusion Detection". Computer Communications, vol. 31, no. 1, pp. 58-72, 2008.

[19] W. Wang, S. Gombault, A. Bsila, “Building multiple behavioral models for network intrusion identification”, 2nd IEEE Workshop on Monitoring, Attack Detection and Mitigation, November 5-6, Toulouse, France, pp. 31-36, 2007.

[20] W. Wang and R. Battiti, “Identifying Intrusions in Computer Networks with Principal Component Analysis”. Proceedings of the First International Conference on Availability, Reliability and Security (ARES 2006), IEEE press society, pp. 270-277, Vienna, Austria, 2006.

[21] Y. Bouzida and F. Cuppens, “Neural networks vs. decision trees for intrusion detection”. First IEEE workshop on Monitoring, Attack Detection and Mitigation, Tuebingen, Germany, 2006.

[22] Y. Bouzida, F. Cuppens, N. Cuppens-Boulahia, S. Gombault, “Efficient intrusion detection using principal component analysis”. 3ème Conférence sur la Sécurité et Architectures Réseaux , France, 2004.

[23] L. Yu, H. Liu, “Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution”, pp. 856-863, ICML 2003.

[24] A. H. Sung, S. Mukkamala, “Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks”, Symposium on Applications and the Internet, 2003.

[25] Das, S. (2001). Filters, wrappers and a boosting-based hybrid for feature selection. pp. 74-81, ICML 2001.

[26] D. Heckerman, A Tutorial on Learning With Bayesian Networks, Microsoft Research, Technical Report MSRTR-95-06, March 1995.

[27] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern Classification. China Machine Press, Beijing, 2nd edition edition, 2004.

[28] J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.