American Journal of Applied Sciences 6 (11): 1948-1959, 2009
ISSN 1546-9239

From Feature Selection to Building of Bayesian Classifiers: A Network Intrusion Detection Perspective

Kok-Chin Khor, Choo-Yee Ting and Somnuk-Phon Amnuaisuk
Faculty of Information Technology,
Multimedia University, Cyberjaya, 63100, Selangor, Malaysia

Abstract: Problem statement: Implementing a single or multiple classifiers that involve a Bayesian Network (BN) is a rising research interest in network intrusion detection domain. **Approach:** However, little attention has been given to evaluate the performance of BN classifiers before they could be implemented in a real system. In this research, we proposed a novel approach to select important features by utilizing two selected feature selection algorithms utilizing filter approach. **Results:** The selected features were further validated by domain experts where extra features were added into the final proposed feature set. We then constructed three types of BN namely, Naive Bayes Classifiers (NBC), Learned BN and Expert-elicited BN by utilizing a standard network intrusion dataset. The performance of each classifier was recorded. We found that there was no difference in overall performance of the BNs and therefore, concluded that the BNs performed equivalently well in detecting network attacks. **Conclusion/Recommendations:** The results of the study indicated that the BN built using the proposed feature set has less features but the performance was comparable to BNs built using other feature sets generated by the two algorithms.

Key words: Network intrusion detection, Bayesian classifiers, feature selection

INTRODUCTION

Intrusive attempts on computer networks have become more prominent considering the increasingly important role played by Internet systems in our daily life[1]. Individuals and organizations nowadays hardly accomplish their daily tasks without relying on the conveniences provided by computer networks and Internet technologies. Therefore, intruders could have exploited the weaknesses of such technologies to take advantage of the information gained from the individuals as well as organizations.

A series of protective measures have been taken to protect Internet systems, which includes the setting up of firewall, anti-virus software, intrusion detection systems (IDSs) and implementation of a proper security policy. IDSs are one of the mentioned measures that have received extensive attention by the public to protect their Internet systems. IDSs are used to identify, classify and possibly, to respond to benign activities[2].

There are two basic types of IDSs, namely, Host-based IDS (HIDS) and Network-based IDS (NIDS). The HIDS monitors activities in a computer system without considering the activities in the computer network where the computer system is located. The NIDS, as opposed to HIDS, are not concerned with activities in individual computer systems but monitor activities in the computer network(s) where the computer systems are located. Sensors of NIDS are deployed at network entry points in order to monitor traffics that traverse the networks. Both HIDS and NIDS can be implemented as passive or inline technology. The IDSs that utilize inline technology is able to prevent damages once an intrusion is found. On the other hand, IDSs that work passively typically log intrusive activities without preventing the losses caused by intruders.

There are two basic approaches for HIDS and NIDS in detecting intrusions: (1) misuse detection and (2) anomaly detection. IDSs that employ misuse detection approach detect attacks by comparing the existing signatures against the network traffics captured by the IDSs. When a match is found, the IDSs will take action as the traffics are considered harmful to computer systems or computer networks. Actions taken by the IDSs will normally include sending alerts to network administrator and logging the intrusive events. IDSs that implement misuse detection approach are, however, incapable of detecting novel attacks. The network administrator will need to update the stored

Corresponding Author: Kok-Chin Khor, Faculty of Information Technology, Multimedia University, Cyberjaya, 63100, Selangor, Malaysia

signatures frequently to make sure that the IDSs perform well in detecting intrusions. IDSs that employ anomaly detection are capable of identifying novel attacks, that contain activities deviated from the norm. Such IDSs utilize the built profiles that are learned based on normal activities in computer networks. Nevertheless, false positive alarms are likely generated by the IDSs as activities in computer networks do not always follow the norm. For instance, a server in a computer network might receive an incredibly large number of connections from the public in a short period due to its interesting content.

In this study, we attempted a NIDS that employs Bayesian approach to detect intrusive activities in computer networks. Empirical evaluation was conducted to obtain optimal features to built different types of BNs by leveraging on a standard network intrusion detection dataset. In addition, stratified sampling of the standard dataset was performed to obtain four different sizes of datasets. Using the datasets, BNs built using the selected features were tested to investigate their performance in detecting intrusions in computer networks.

Related work: Researchers have utilized various Artificial Intelligence (AI) approaches and data mining techniques to construct a better IDS. Bayesian approach has been one of the major AI approaches utilized by the researchers in the network security domain[3-13].

A study by[3] classified intrusions using both BNs and Classification and Regression Trees (CART). The features of the intrusion data were selected based on Markov Blanket of the target variables. An ensemble classifier was constructed by combining both approaches to increase robustness, accuracy and better overall generalization.

An interesting research by[4] proposed an IDS with a cooperative agent architecture. The system allows the agents to share belief on an event occurrence and perform soft-evidence update to enable a continuous scale for intrusion detection. There are three types of agents in the proposed system: system monitoring agent, intrusion monitoring agent and registry agent. The system monitoring agent is responsible for processing log data upon request and communicates with the operating system. Such agents publish their facts and beliefs derived from observations of each other. Intrusion monitoring agent, on the other hand, performs belief update based on BNs using observed values (hard evidence) and derived values (beliefs or soft evidence) from other agents as well. Using both hard and soft findings, the system is able to identify various known attacks. In the research, each intrusion monitoring agent encapsulates an Expert-elicited BN and is responsible for monitoring a particular type of intrusion. Therefore, the modification of an intrusion pattern will not affect others.

A hybrid intelligent IDS developed by[5] incorporated BN and Self-Organizing Map (SOM). In this research, SOM theory was slightly modified for the standard network intrusion dataset, which contains labels. The experimental results showed that the performance of the hybrid intelligent IDS was better compared to the non-hybrid Bayesian learning approach.

Research for comparing performance of different classifiers were conducted as well. The research by[6] has shown that Naïve Bayes Network depicts competitive results when compared to Decision Trees, despite the fact that Naïve Bayes Network works based on the assumption that all variables involved are conditional independent from each other.

A framework for an adaptive intrusion detection system was proposed by[7] using BN. In this research, any new network data that was considered intrusive by the system will be added to the dataset. The IDS was therefore, updated from time to time.

The technical report of[8] proposed a new model for intrusion detection that is able to classify new unlabeled data and allow for constant updating whenever new data is captured. The author exploited the possibility of developing the model using Partially Observable Markov Decision Process (POMDP).

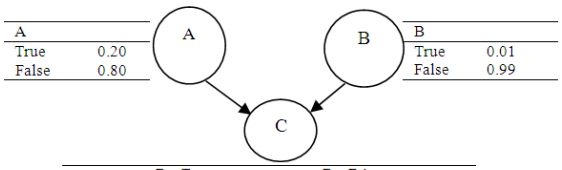
Session Anomaly Detection (SAD) was proposed by[12], which utilized Bayesian parameter estimation method to analyze web logs and detecting anomalous sessions generated by the Whisker and Nimda worms. SAD functions by developing a normal usage profile and compared it to the generated web logs against the expected frequency. The study reported that SAD performed better than SNORT, which used misuse detection technique.

A study by[13] proposed a method to effectively analyze data that were collected by the distributed IDS based on Bayesian Multiple Hypothesis Tracking (BMHT), so that the related incidents can become apparent. As discussed in[10], most of the existing research works concentrate only on a network that the IDS want to protect and therefore only the information of attack activities that occurred in the network will be gathered. To have a complete view of an intruder's action, the author suggested an approach in gathering data from more than one network via IDSs. The BMHT is used to reorganize network data so that a better view of the activities occurring in the networks can be obtained.

The above mentioned research works reported various network intrusion detection methods, which utilized a single type of BN or a BN is used together with other classifier in building a better IDS. However, these research works did not evaluate the performance of different types of BN before deciding to use either one of them. Therefore in this study, we investigated how different types of BN perform in identifying various types of attacks. Two known types of BN and a BN crafted based on the domain knowledge on attacks were built and evaluated.

Bayesian networks: BN is a prevailing method for dealing with uncertainty in real-world decision making and it has been applied to various research domains successfully. There are major advantages of using BN in various research domains. A research domain can be understood well as the BN structure provides explicit inter-relationships among the data set attributes. Besides, methods are provided for handling missing data and to prevent over-fitting of data. Data and domain knowledge can be combined because a BN model has both a causal and probabilistic semantics[14]. Human interventions are allowed to modify the BN to increase the performance of the predictive model. Furthermore, the expert-elicited network can be further enhanced using probability learning and network learning method to achieve higher accuracy of prediction. Adding decision node and utility node to the network will extend the capability of a BN for decision analysis.

BN is a Directed Acyclic Graph (DAG) and its structural representation is represented by nodes that correspond to random variables in a problem domain. Arcs in a BN represent causality or influential relationship between parent nodes and child nodes. Nodes in the BN contain states of random variables. As shown in Fig. 1, the BN is structured in such a way that only the node C has Conditional Probability Table (CPT) given its parents. Nodes A and B have only prior probability tables since they do not have any parent node. The CPT describes the strength between the parent node A and the child node C as well as the parent node B and the child node C. Assuming that all the nodes in Fig. 1 have two states, thus the CPT for node C has a $2^3 = 8$ probability value entries.



	B = True		B = False	
C	A = True	A = False	A = True	A = False
True	0.10	0.99	0.95	0.05
False	0.90	0.01	0.05	0.95

Fig. 1: A simple BN

Consider a BN with n nodes, with X representing random variables and x denotes the states of the random variables. The joint distribution is presented by $P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$, or in a more compact way, $P(x_1, x_2, ...,x_n)$. The graph specifies a factorization of the joint probability distributions based on the chain rule:

$$P(x_1, x_2, ..., x_n) = \prod_{i=1} P(x_i | x_1, ..., x_{i-1})$$

A BN can be described via qualitative and quantitative components. The qualitative component is presented by the structure while the quantitative component, through its CPT. Posterior probabilities of query variables can be calculated in light of any evidence by having both the qualitative and quantitative representation of BN. By using Bayes's rule and an inference algorithm, BN can be used to perform diagnostic, predictive and inter-causal reasoning, as well as any combination of the above[15].

There are three basic types of BN classifiers, namely, Naive Bayesian Classifier (NBC), Learned BN and Expert-elicited BN. The NBC is the simplest BN model that consumes low computational power. The NBC has child nodes where they all share the same and single parent node. The NBC assumes conditional independence for the child nodes.

There are two steps involved in building a Learned BN. Firstly, the DAG has to be induced using existing algorithms such as, PC, K2 and NPC. Secondly, the parameters as defined by the DAG have to be estimated. Parameter estimation can be conducted using algorithm such as Expectation-Maximization (EM).

Besides constructing BNs using existing machine learning algorithms, a BN can be constructed manually by eliciting knowledge of a domain expert. The construction process is a repetitive process, which involves model verification and model revision.

There are basically three categories of variables, namely, problem variables, information variables and mediating variables to be identified by domain experts in constructing a BN manually. Problem variables are related to classification, which in this study, classify intrusions in computer networks. Information variables, on the other hand, provide information relevant to classifying network intrusions. The features of the dataset we used in this study will be served as evidence for classifying intrusions. The information variables can be further divided into two sub-categories namely, background information variables and symptom information variables. Background information is the

information available before the problem exists whereas symptom variables can be viewed as consequences after the occurrence of the problem. Since the background information came before the problem, thus, background information variables will be the root of a DAG. The mediating variables serve as unobservable variables, which are used to counter the dependency of two or more information variables for solving the problem[16]. The causal relations of the variables are as shown in Fig. 2.

Attack categories in the dataset: The standard network intrusion dataset involved is commonly used in network security research for training and evaluating IDSs[17-26]. It consists of records that can further be divided into five categories, namely, normal, Denial of Service (DoS), Probing (Probe), Remote to Local (R2L) and User to Root (U2R).

DoS attacks are performed to a host by using up its resources so that it will not be able to provide network service to the legitimate users. DoS attacks are most feared as such attacks do not require intruders to access to a victim machine. Performing DoS attacks can be as simple as running a script or a tool. There are many types of DoS attacks. Smurf attack is one of its many types. By performing Smurf attack, an intruder sends large amount of spoofed Internet Control Message Protocol (ICMP) messages to broadcast addresses of a computer network. Hosts in the computer network will reply the ICMP messages and this will eventually multiply the network traffics in the computer network. A computer network can be saturated if such network traffics are huge in number.

Probing normally precedes an actual access or DoS attack. Probing can be performed by utilizing freely available tools in the Internet such as Nmap, so that vulnerabilities of a particular host or a computer network can be found. Such tool can be used to ping sweeps a computer network to generate a list of potential victim machines. Port scanning can then be performed on any of the machine in the list to find out the ports or services that are currently active.

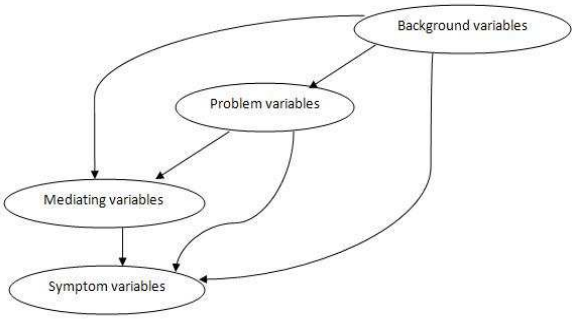


Fig. 2: The causal relations of various variables in a BN

Intruders can soon send queries to gather information such as application type, version of the application or probably operating system to figure out the possible vulnerabilities to be exploited.

R2L attacks are conducted by sending packets to a targeted machine in a computer network to gain access as if the intruders own an account in the targeted machine. R2L attacks can be performed in many forms. It takes advantage of weakly configured security features, perform buffer overflow attacks and guess or capture password of hosts in computer networks. Whereas for U2R attacks, a local user may exploit flaws in poorly designed systems so that root level privileges can be obtained[27].

MATERIALS AND METHODS

Pre-processing the dataset: The standard dataset used for network intrusion detection domain was a result of a DARPA intrusion detection evaluation program[28]. It consists of 494,021 records with 41 features and each of the records is labeled with a class Normal or any of the 22 types of attacks. One of the records was however removed due to errors.

The 22 types of attacks were later being categorized into four attack categories. The reason to categorize the attacks into four attack categories is to ease the classification tasks in the later stage as some of the attacks consist of only a few records. Nevertheless, unevenly distributed number of records could still be seen after categorization as illustrated in Fig. 3. Attack category such as U2R consists of only 52 records while DoS consists of nearly 0.4 millions of records. Consequently, classification accuracies of category such as U2R might be affected. However, better classification accuracies will be obtained in handling four attack categories rather than handling 22 types of attacks.

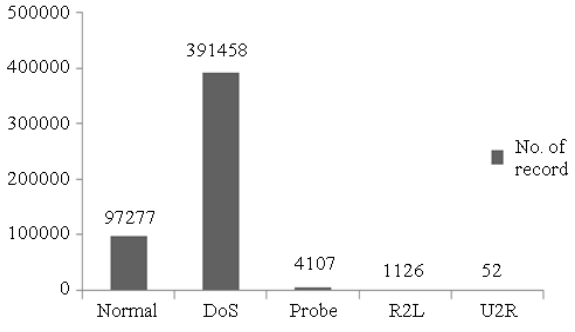


Fig. 3: The distribution of attack categories in the standard dataset

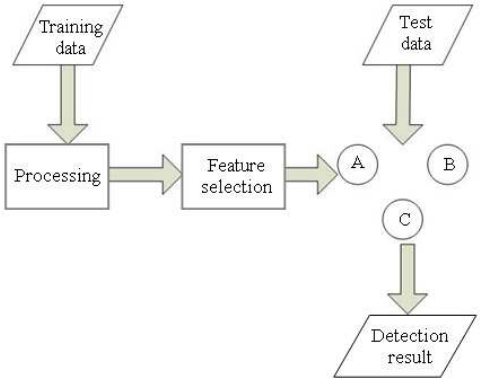


Fig. 4: The proposed IDS architecture

1. Algorithm **OptimalFeatureSet**()
2. **Input**:
3. An array D storing DARPA dataset with 494020 records.
4. An array F containing 41 features from D.
5. Feature Selection Algorithms: FSM_1 and FSM_2
6. **Output**: 5 feature sets, FS_5 is the optimal one
7. // Pre-processing phase
8. Randomize (D)
9. Discretization (F, D)
10. RemoveIrrelevantCharacters (F, D)
11. // Applying Feature Selection algorithms FSM_1 and FSM_2
12. $FS_1 \leftarrow$ Feature set obtained from selection FSM_1
13. $FS_2 \leftarrow$ Feature set obtained from selection FSM_2
14. // obtaining the shared feature set FS_3 and combined feature
15. //set FS_4
16. $FS_3 \leftarrow FS_1 \cap S_2$
17. $FS_4 \leftarrow FS_1 \cup FS_2$
18. $A_{max} \leftarrow 0$
19. $FS_{temp} \leftarrow FS_3$
20. **For each feature** F_n **in** $(FS_4 - FS_3)$ **do**
21. $FS_{temp} \leftarrow$ **addFeature** (F_n, FS_{temp}).
22. $A_{temp} \leftarrow$ **computeAccuracy** (FS_{temp}).
23. **If** $A_{temp} > A_{max}$ **then**
24. $A_{temp} \leftarrow A_{max}$
25. $FS_5 \leftarrow FS_{temp}$
26. **End If**
27. **End for**

Fig. 5: The algorithm to obtain an optimal feature set

A NIDS is proposed in the project. Preprocessing, feature selection and intrusion detection are the stages involved in constructing the NIDS. The stages are as illustrated in Fig. 4. As shown in Fig. 5, the records of the standard dataset were randomized and values of each of the features of the records were discretized at the preprocessing stage (line 8 and 9). Special characters for instance, "\" and "_" were seen after discretization. The special characters will increase the size of the dataset and consequently increase the computational cost in processing the dataset. Removal of these special characters is thus necessary (line 10).

Feature selection approach: The number of features required is another major concern in processing the dataset as well. As the number of features increase, the relationships among the features as well as the relationships between features and classes will become very complex. High computational cost will inevitably be needed in processing such complex relationships. It is thus necessary to undergo a feature selection stage to obtain an optimal feature set with less number of features but able to provide high detection accuracies.

We proposed a novel feature selection approach in which the decision of feature selection algorithms and opinion of experts were incorporated. In our approach, two filter-based feature selection methods were used to confirm important features of the dataset. Additional features which are considered important by the domain expert were added to identify network intrusions.

As shown in Fig. 5, two filter-based feature selection methods were utilized at the feature selection stage to produce two feature sets (FS_1 and FS_2) (line 12 and 13). Correlation-based Feature Selection Subset Evaluator (CFSE) and Consistency Subset Evaluator (CSE) were utilized by these two feature selection methods. CFSE uses an algorithm that works together with an evaluation formula, in which the ideas are based on test theory. Good features are then selected with an appropriate correlation measure and a heuristic search strategy. The algorithm has the advantages in identifying irrelevant, redundant and noisy features fast. Relevant features can be identified as long as their relevance does not strongly depend on other features[29]. On the other hand, inconsistency of a feature set class given different class labels is measured by CSE. The algorithm involved is monotonic and has the advantage of removing redundant or irrelevant features fast. It is also multivariate and able to handle noises in dataset[30].

Confirmation of important features was done by extracting the shared features of these two feature sets to form a shared feature set (FS_3). These two feature sets were then combined without repeating the same features to generate a combined feature set (FS_4) (line 16 and 17).

The neglected features (F_n) related to Probe, R2L and U2R attacks were selected by domain experts and added one by one into the shared feature sets to form the proposed feature set (FS_5) (line 20-27). As the numbers of records of these attacks were relatively small compared to DoS and Normal, thus classification accuracies were expected to be low. Intervention of domain expert might help in this case. Considering the characteristics of probe attacks, features such as dst_host_count and dst_host_rerror_rate needed to be added. dst_host_count was selected among the neglected features as the Probe attacks involved a large number of connections to a same destination host.

Table 1: The features of the five feature sets

Feature set	Selected features	No. of features
CFSE (FS_1)	Service, dst_bytes, logged_in, root_shell, count, srv_diff_host_rate, dst_host_count, dst_host_srv_diff_host_rate	8
CSE (FS_2)	Service, src_bytes, dst_bytes, logged_in, count, dst_host_srv_count, dst_host_diff_srv_rate, dst_host_rerror_rate	8
Combined (FS_3)	Service, dst_bytes, logged_in, root_shell, count, srv_diff_host_rate, dst_host_count, dst_host_srv_diff_host_rate, src_bytes, dst_host_srv_count, dst_host_diff_srv_rate, dst_host_rerror_rate	12
Shared (FS_4)	Service, dst_bytes, logged_in, count	4
Proposed (FS_5)	Service, dst_bytes, logged_in, count, dst_host_count*, root_shell*, dst_host_rerror_rate*	7

*: Features that were selected based on domain knowledge

Table 2: Description of the features involved

Features	Description	Value type
service[i]	Type of network service on the destination	Discrete
dst_bytes[i]	Number of data bytes from destination to source	Continuous
src_bytes[i]	Number of data bytes from source to destination	Continuous
logged_in[d]	Login successful or otherwise	Discrete
root_shell[d]	Root shell is obtained or otherwise	Discrete
count[t]	Number of connections to the same host as the current connection	Continuous
dst_host_count[c]	Number of connections to the same host as the current connection	Continuous
srv_diff_host_rate[t]	Rate of connections to different hosts	Continuous
dst_host_srv_diff_host_rate[c]	Rate of connections to different hosts	Continuous
dst_host_srv_count[c]	Number of connections to the same service as the current connection	Continuous
dst_host_diff_srv_rate[c]	Rate of connections to different services	Continuous
dst_host_rerror_rate[c]	Rate of connections that have “REJ” errors	Continuous

i: Intrinsic features; d: Features that are derived from domain knowledge; t: Features that are formed using a 2 sec time window; c: Features that are formed using a connection window that consists of 100 connections

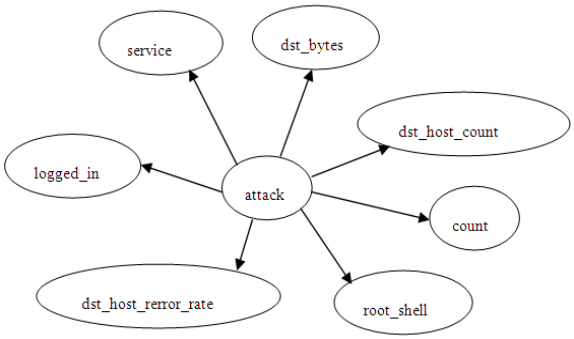


Fig. 6: The same structure of NBC was used for the four datasets

dst_host_rerror_rate was considered as well because certain probing attacks have larger time interval in scanning hosts or ports. These features are formed using a connection window that consists of 100 connections. root_shell was included to detect U2R and R2L attacks, which involve unauthorized access to a machine. The finalized features for proposed feature set and other feature sets are as shown in Table 1. Explanation of the features is given in Table 2.

Five independent datasets were formed based on the features of these five feature sets. BNs were built using K2 algorithm and 10-fold cross validation was conducted to evaluate the BNs' classification accuracies. The feature set with optimal performance will be selected for the next experiment.

Constructing BNs as classifiers to intrusion detection: In the next experiment, performances of different types of BNs were evaluated. The dataset was re-sampled to provide another three sample datasets in different sizes (75, 50 and 25 of the standard dataset). The re-sampling was done to produce sample datasets that have the same class distribution as the original dataset.

The intrusion detection stage involved three BN classifiers, namely, NBC, Learned BN and Expert-elicited BN. The optimal feature set decided in the previous experiment was used to construct the BNs. The NBC is made simplified by assuming the variables are conditional independence of each other (Fig. 6). The Learned BN can be constructed using a few existing search algorithms. Experiment was conducted based on the datasets in order to choose an algorithm that has the optimal performance.

On the other hand, the Expert-elicited BN allowed researchers to incorporate expert views into it. To construct an Expert-elicited BN, various types of variables need to be identified. Intrinsic features of the resulted dataset such as service, dst_bytes and service existed in raw dataset. Thus, they would be treated as the background variables in constructing the BN. Classes of various types of intrusions (DoS, Probe, R2L and U2R) will be represented using a problem variable.

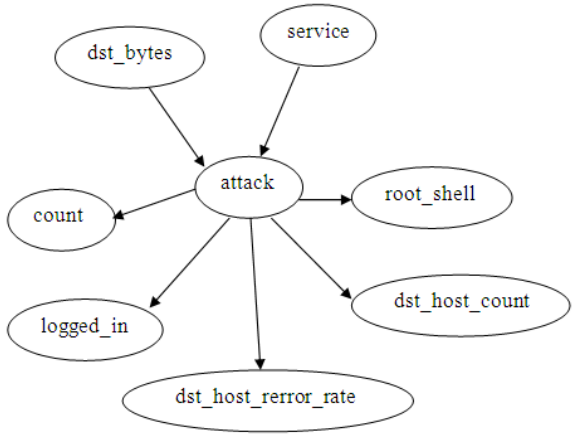


Fig. 7: The Expert-elicited BN, which was used for the four datasets

There are features in the dataset, which are derived from raw dataset. Features such as logged_in, count, dst_host_count, dst_host_rerror_rate and root_shell were formed based on knowledge of the domain and they were treated as symptom variables. The symptoms variables served as evidences for classifying the intrusions. Mediating variables were not in our consideration in constructing the BN as the variables in this study are observable. The Expert-elicited BN is as shown in Fig. 7. The root of the BN was the background variables as they have direct influence on the problem variables. The domain experts incorporated their views regarding the attacks by refining and verifying the parameters of the nodes of the Expert-elicited BN.

RESULTS

The first experiment was conducted to select an optimal feature set while the second experiment involved a selection of search algorithm for building Learned BN. Various types of BNs were then constructed based on the optimal feature set and the performances of these BNs were compared in the last experiment.

The results of the first experiment showed that the performances of the feature sets were comparable to each other except the shared feature set (Table 3). The performance of the BN built using the shared feature set was poor especially in Probe, R2L and U2R attacks (63.3, 33.8 and 23.1%).

To compare the performances of the BNs built using the proposed feature set and other feature sets, an independent-samples t-test was conducted. The result in Table 4 shows that there was no significant difference between the proposed feature set and other feature sets except shared feature set. The poor performance of the BN built in Probe, R2L and U2R was the cause of the significant difference between the proposed feature set and shared feature set.

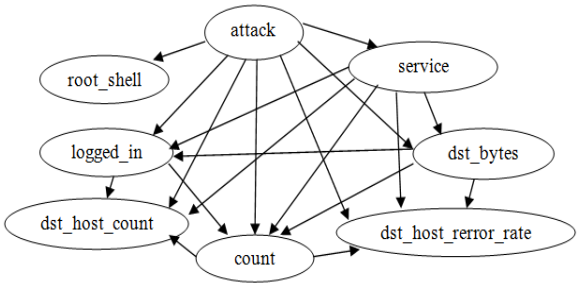


Fig. 8: One of the BNs built using K2 algorithm

Table 3: The classification accuracy (%) of BNs built based on five different feature sets

	Feature sets				
Category	CFSE	CSE	Combined	Shared	Proposed
Normal	99.9	99.9	99.9	99.6	99.8
DoS	100.0	100.0	100.0	99.9	99.9
Probe	66.8	98.3	98.1	63.3	89.4
R2L	91.0	96.4	97.3	33.8	91.5
U2R	65.4	34.6	55.7	23.1	69.2

Table 4: Significance test of classification accuracy between BNs built using the proposed and other feature sets

	CFSE	CSE	Combined	Shared
Proposed	0.29	0.63	0.95	0.09

*: $p<0.1$

A selection was then conducted to select a search algorithm with optimal performance for building Learned BN. Despite many more search algorithms, the selected search algorithms K2, Hill-climber and Tabu[31-33] were identified because they are computationally feasible in processing huge dataset compared to others.

As shown in Table 5, the Tabu search algorithm performed relatively weak in classifying R2L and U2R attacks compared to others irrespective to the sizes of dataset. We thus compared only the performance of K2 and Hill-climber search algorithms. An independent-samples t-test was conducted as well. The results show that there was no significant difference between the BNs built using K2 and hill-climber search algorithms (Table 6). K2 was selected to build Learned BN and later compared with the performances of other types of BNs (Fig. 8).

Table 7 shows the classification accuracies of different BNs based on various sizes of dataset. We conducted an independent-samples t-test and no difference in accuracy between the BNs regardless of the sizes of the dataset (Table 8).

Table 5: The classification accuracies (%) of BNs built based on three different sizes of dataset using three different search algorithms

	Quarter			Semi			3-Quarter			Full		
Category	K2	Hc	Tb	K2	Hc	Tb	K2	Hc	Tb	K2	Hc	Tb
Normal	99.7	99.7	99.4	99.8	99.7	99.5	99.8	99.7	99.0	99.8	99.7	99.1
DoS	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.7	99.9	99.9	99.7
Probe	87.1	88.5	78.5	88.0	90.2	78.5	89.1	91.8	73.0	89.4	92.2	74.8
R2L	84.3	84.7	33.5	89.0	87.5	32.2	89.3	90.7	15.7	91.5	91.9	18.8
U2R	47.1	64.7	11.8	66.7	66.7	37.0	71.8	74.4	12.8	69.2	67.3	9.6

Note: Tabu search algorithm performed poorly in classifying R2L and U2R attacks. *Hc: Hill-climber search algorithm; Tb: Tabu search algorithm

Table 6: Significance test of classification accuracy between BNs built using algorithms K2 and hill-climber

	Hill-climber			
	Quarter	Semi	3-Quater	Full
K2	0.32	0.85	0.19	0.77

*: $p<0.1$

Table 7: Comparison in terms of classification accuracies of the three BNs built based on four different sizes of dataset

	Quarter			Semi			3-Quarter			Full		
Category	NBC	BN	EE	NBC	BN	EE	NBC	BN	EE	NBC	BN	EE
Normal	96.9	99.7	97.3	96.8	99.8	97.4	96.7	99.8	97.3	96.7	99.8	97.5
DoS	99.3	99.9	99.3	99.3	99.9	99.2	99.2	99.9	99.2	99.3	99.9	99.2
Probe	93.0	87.1	93.2	92.6	88.0	92.8	93.8	89.1	93.0	93.5	89.4	93.0
R2L	89.9	84.3	83.5	91.8	89.0	85.4	92.5	89.3	87.0	92.8	91.5	86.8
U2R	52.9	47.1	76.5	59.3	66.7	77.8	61.5	71.8	76.9	55.8	69.2	69.2

*NBC: Naïve Bayes Classifier; BN: Learned BN; EE: Expert Elicited BN

Table 8: Significance test of classification accuracies for different types of BNs

	Quarter		Semi		3-Quarter		Full	
	NBC	BN	NBC	BN	NBC	BN	NBC	BN
EE	0.53	0.35	0.57	0.54	0.61	0.68	0.66	0.58

*: $p<0.1$

Table 9: Comparison of classification results of various BNs on Full dataset and Wenke Lee

Category	NBC	BN	EE	Wenke lee
Normal	96.7	99.8	97.5	N/A
DoS	99.3	99.9	99.2	79.9
Probe	93.5	89.4	93.0	97
R2L	92.8	91.5	86.8	75
U2R	55.8	69.2	69.2	60

Number in boldface indicates the highest classification accuracy for an attack category

The results of three BNs were used to compare with the results of the researchers that prepared the dataset[34]. The BNs performed better in DoS, R2L and U2R categories and give comparable results in Probe category (Table 9).

As shown in Table 10, NBC was built and tested with the least requirement of time in almost all the datasets regardless of its size. However with full dataset, it required a slightly more time in network building compared to Expert-elicited BN.

Table 10: Time required for building and testing the BNs

	Build (s)			Test (s)		
Size of dataset	NBC	BN	EE	NBC	BN	EE
Quarter	0.20	7.31	0.34	1.95	3.00	2.74
Semi	0.36	20.84	0.50	4.01	5.69	5.34
3-Quarter	0.53	21.83	0.70	5.80	8.58	7.92
Full	0.69	23.27	0.66	7.89	11.56	10.87

The BNs are able to perform well in identifying DoS and Normal attacks (Fig. 9a and b). The BNs gave good performance as well in probe and R2L attacks although the number of records involved was medium as compared to DoS and normal.

Although the differences in terms of classification accuracy were not significant, classification accuracies of Learned BN for Probe category, especially involving small datasets, were slightly lower as compared to other BNs (Fig. 9c). The classification accuracies of NBC for R2L were slightly higher in all sizes of datasets as compared to others (Fig. 9d). The Expert-elicited BN was able to give a better performance as compared to others in this attack category. A difference of 29.4% in classification accuracy was observed in quarter dataset (Fig. 9e).

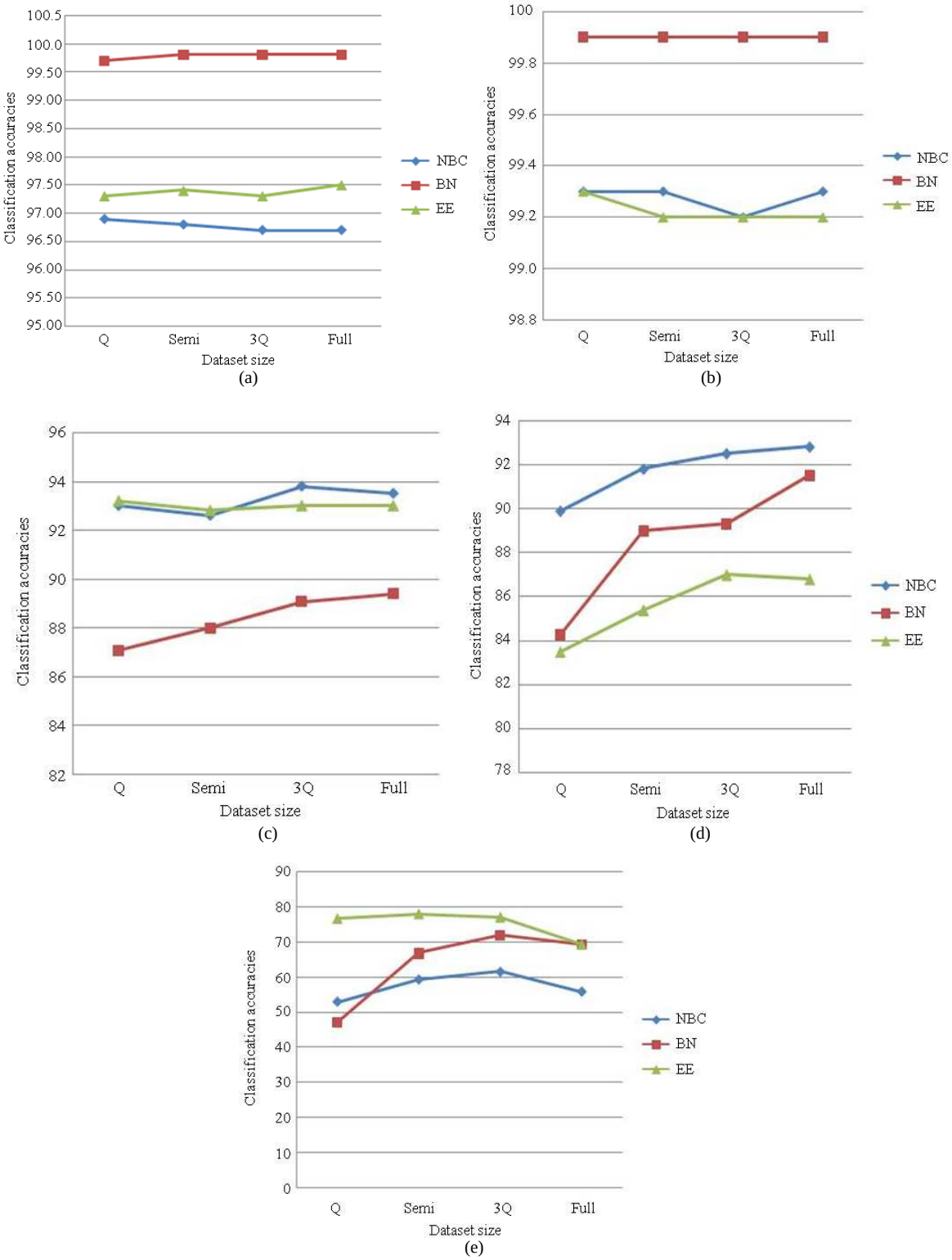


Fig. 9: Comparison of BNs in terms of classification accuracies in different attack categories. (a) Normal category; (b) DoS category; (c) probe category; (d) R2L category; (e) U2R category

DISCUSSION

The proposed feature set generated in the first experiment has fewer features compared to other feature sets. This has suggested that the required computational cost in processing network data can be reduced while preserving the classification accuracy.

In the second experiment, K2 was selected as the search algorithm to build Learned BN considering that it exhibited no difference in terms of performance as compared to Hill-climber search algorithm.

The subsequence experiment also indicated that any type of BN is suitable for identifying attacks utilizing the standard dataset regardless of its sizes. However, NBC can be utilized in identifying attacks if time for building network and testing is a factor.

The huge number of DoS and Normal records in the dataset was the reason for high and consistent classification accuracies of the BNs. The number of records for U2R attacks was very small in the dataset. Classification accuracies were affected and therefore the performances of BNs in the attack category were not that promising. Nevertheless, the results also indicated NBC and Expert-elicited BN demonstrated advantages in identifying R2L and U2R attacks, respectively.

CONCLUSION

In this study, we propose a novel approach for selecting features and comparing the performance of various BN classifiers utilizing the standard network intrusion dataset. The feature selection approach required two feature selection algorithms to confirm the important features of the dataset and later intervention of domain experts to form a proposed feature set. The interventions of domain experts in feature selection has proven to be useful as less features were generated compared to other feature sets produced by the two feature selection algorithms. Moreover, the performance of the BN built by the proposed feature set was comparable to others. Empirical experiment conducted in this research also indicated that any of these three BNs can be used in identifying attacks utilizing the standard network intrusion dataset. The performances of the BNs are comparable to each other. However, NBC should be considered if the dataset involved is huge in size and time for building network and testing is the main factor to consider.

We are considering to implementing multiple-BN classifiers in the future by looking at the advantage of the BNs in identifying certain type of attacks. It was observed that even though the number of records is low for an attack category, the classification accuracies can be maintained or improved with the intervention of domain experts. As such, the Expert-elicited BN should be improved so that it will work better if multiple classifiers or a single classifier is implemented in NIDS.

REFERENCES

1. MyCert, 2007. Network Security Incidents in Malaysia. http://www.mycert.org.my
2. Conklin, W.A., G.B. White, C. Cothren, D. Williams and R.L. Davis, 2005. Principles of Computer Security: Security + and Beyond, 1st Edn., McGraw-Hill, Singapore, ISBN: 007-124500-6, pp: 309-332.
3. Chebrolu, S., A. Abraham and J.P. Thomas, 2005. Feature deduction and ensemble design of intrusion detection system. Comput. Secur., 24: 295-307. DOI: 10.1016/j.cose.2004.09.008.
4. Gowadia, V., C. Farkas and M. Valtorta, 2005. PAID: A probabilistic agent-based intrusion detection system. Comput. Secur., 24: 529-545. DOI: 10.1016/j.cose.2005.06.008
5. Thames, J.L., R. Able and A. Saad, 2006. Hybrid intelligent systems for network security. Proceedings of the 44th Annual ACM Southeast Regional Conference, Mar. 10-12 ACM, pp: 286-289. DOI: 10.1145/1185448.1185513
6. Amor, N.B., S. Benferhat and Z. Elouedi, 2004. Naïve bayes Vs decision trees in intrusion detection systems. Proceedings of the ACM Symposium on Applied Computing, Mar. 14-17, ACM, Nicosia, Cyprus, pp: 420-424. DOI: 10.1145/967900.96798
7. Jemili, F., M. Zaghdoud and A.M. Ben, 2007. A framework for an adaptive intrusion detection system using Bayesian network. IEEE International Conference on Intelligence and Security Informatics, May 23-24, IEEE Xplore Press, NJ., pp: 66-70. DOI: 10.1109/ISI.2007.379535
8. Lane, T., 2004. A decision-theoretic, semi-supervised model for intrusion detection. Technical Report TR-CS-2004-16, University of New Mexico. http://www.cs.unm.edu/~treport/tr/04-07/intrusion.pdf
9. Cha, B.R. and D.S. Lee, 2007. Network-based anomaly intrusion detection improvement by Bayesian network and indirect relation. Lecture Notes Comput. Sci., 4693: 141-148. DOI: 10.1007/978-3-540-74827-4_18
10. Bulatovic, D. and D. Velasevic, 1999. A distributed intrusion detection system based on Bayesian alarm networks. Lecture Notes Comput. Sci., 1740: 219-228. DOI: 10.1007/3-540-46701-7_19

11. Kruegel, C., D. Mutz, W. Robertson and F. Valeur, 2003. Bayesian event classification for intrusion detection. Proceeding of 19th Annual Computer Security Applications Conference, Dec. 8-12, IEEE Computer Society, Washington DC., pp: 14-23. DOI: 10.1109/CSAC.2003.1254306
12. Cho, S. and S. Cha, 2004. SAD: Web session anomaly detection based on parameter estimation. Comput. Secur., 23: 312-319. DOI: 10.1016/j.cose.2004.01.006
13. Burroughs, D.J., L.F. Wilson and G.V. Cybenko, 2002. Analysis of distributed intrusion detection systems using Bayesian methods. Proceeding of the 21st IEEE International Performance, Computing and Communication Conference, Apr. 3-5, IEEE Computer Society, Washington DC., pp: 329-334. DOI: 10.1109/IPCCC.2002.995166
14. Heckerman, D., D. Geiger and D.M. Chickering, 1995. Learning Bayesian networks: The combination of knowledge and statistical data. Machi. Learn., 20: 197-243. DOI: 10.1007/BF00994016
15. Korb, K.B. and A.E. Nicholson, 2004. Bayesian Artificial Intelligence. 1st Edn., Chapman and Hall/CRC Press, Boca Raton, ISBN: 1584883871, pp: 29-43.
16. Kjaerulff, U.B. and A.L. Madsen, 2008. Bayesian Networks and Influence Diagram: A Guide to Construction and Analysis. 1st Edn,. Springer, New York, ISBN: 978-0-387-74100-0, pp: 140-172.
17. Sharma, A., A.K. Pujari and K.K. Paliwal, 2007. Intrusion detection using text processing techniques with a kernel based similarity measure. Comput. Secur., 26: 488-495. DOI: 10.1016/j.cose.2007.10.003
18. Abadeh, M.S., J. Habibi and C. Lucas, 2007. Intrusion detection using a fuzzy genetics-based learning algorithm. J. Network Comput. Appli., 30: 414-428. DOI: 10.1016/j.jnca.2005.05.002
19. Hansen, J.V., P.B. Lowry, R.D. Meservy and D.M. McDonald, 2007. Genetic programming for prevention of cyberterrorism through dynamic and evolving intrusion detection. Decis. Support Syst., 43: 1362-1374. DOI: 10.1016/j.dss.2006.04.004
20. Liu, G., Z. Yi and S. Yang, 2007. A Hierarchical intrusion detection model based on the PCA neural networks. NeuroComputing, 70: 1561-1568. DOI: 10.1016/j.neucom.2006.10.146
21. Abrahm, A., R. Jain, J. Thomas and S.Y. Han, 2007. D-SCIDS: Distributed soft computing intrusion detection system. J. Network Comput. Appli., 30: 81-98. DOI: 10.1016/j.jnca.2005.06.001
22. Chen, W.H., S.H. Hsu and H.P. Shen, 2005. Application of SVM and ANN for intrusion detection. Comput. Operat. Res., 32: 2617-2634. DOI: 10.1016/j.cor.2004.03.019
23. Mukkamala, S., A.H. Sung and A. Abraham, 2005. Intrusion detection using an ensemble of intelligent paradigms. J. Network Comput. Appli., 28: 167-182. DOI: 10.1016/j.jnca.2004.01.003
24. Giacinto, G., F. Roli and L. Didaci, 2003. Fusion of multiple classifiers for intrusion detection in computer networks. Patt. Recog. Lett., 24: 1795-1803. DOI: 10.1016/S0167-8655(03)00004-7
25. Haines, J.W., L.M. Rossey, R.P. Lippmann and R.K. Cunningham, 2001. Extending the DARPA off-line intrusion detection evaluations. Proceedings of DARPA Information Survivability Conference and Exposition II, June 12-14, IEEE Xplore Press, Anaheim, CA., USA., pp: 35-45. DOI: 10.1109/DISCEX.2001.932190
26. Cabrera, J.B.D., B. Ravichandran and R.K. Mehra, 2000. Statistical traffic modeling for network intrusion detection. Proceedings of 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Aug. 29Sept. 1, IEEE Computer Society, Washington DC., pp: 466-473. DOI: 10.1109/MASCOT.2000.876573
27. Lippmann, R.P., J.W. Haines, D.J. Fried, J. Korba and K. Das, 2002. The 1999 DARPA off-line intrusion detection evaluation. Proceedings of DARPA Information Survivability Conference and Exposition, pp: 12-26. DOI: 10.1016/S1389-1286(00)00139-0
28. ACM KDDCUP, 2009. Computer network intrusion detection. http://www.sigkdd.org/kddcup/index.php
29. Hall, M.A., 1999. Correlation-based Feature Subset Selection for Machine Learning. Ph.D. Thesis, Department of Computer Science, Waikato University. http://www.cs.waikato.ac.nz/~mhall/thesis.pdf
30. Dash, M. and H. Liu, 2003. Consistency-based search in feature selection. Artif. Intel., 151: 155-176. DOI: doi:10.1016/S0004-3702(03)00079-1
31. Cooper, G.F. and E. Herskovits, 1992. A Bayesian method for the induction of probabilistic networks from data. Mach. Learn., 9: 309-347. DOI: 10.1007/BF00994110
32. Buntine, W.L., 1996. A guide to the literature on learning probabilistic networks from data. IEEE Trans. Knowl. Data Eng., 8: 195-210. DOI: 10.1109/69.494161

33. Bouckaert, R.R., 1995. Bayesian belief networks: From construction to inference. Ph.D. Thesis, University of Utrecht. http://igitur-archive.library.uu.nl/dissertations/01856336/inhoud.htm

34. Lee, W. and S.J. Stolfo, 2000. A framework for constructing features and models for intrusion detection systems. ACM Trans. Inform. Syst. Secur., 3: 227-261. DOI: 10.1145/382912.382914