Bayesian Model Averaging of Bayesian Network Classifiers for Intrusion Detection

Liyuan Xiao, Yetian Chen, and Carl K. Chang
Department of Computer Science
Iowa State University
Ames, Iowa 50011
Email: {liyuan, yetianc, chang}@iastate.edu.

***Abstract*—Bayesian network (BN) classifiers with powerful reasoning capabilities have been increasingly utilized to detect intrusion with reasonable accuracy and efficiency. However, existing BN classifiers for intrusion detection suffer two problems. First, such BN classifiers are often trained from data using heuristic methods that usually select suboptimal models. Second, the classifiers are trained using very large datasets which may be time consuming to obtain in practice. When the size of training dataset is small, the performance of a single BN classifier is significantly reduced due to its inability to represent the whole probability distribution. To alleviate these problems, we build a Bayesian classifier by Bayesian Model Averaging(BMA) over the k-best BN classifiers, called Bayesian Network Model Averaging (BNMA) classifier. We train and evaluate BNMA classifier on the NSL-KDD dataset, which is less redundant, thus more judicial than the commonly used KDD Cup 99 dataset. We show that the BNMA classifier performs significantly better in terms of detection accuracy than the Naive Bayes classifier and the BN classifier built with heuristic method. We also show that the BNMA classifier trained using a smaller dataset outperforms two other classifiers trained using a larger dataset. This also implies that the BNMA is beneficial in accelerating the detection process due to its less dependance on the potentially prolonged process of collecting large training datasets.**

***Keywords*-Intrusion detection system, Bayesian network, Bayesian Model Averaging, detection accuracy.**

I. Introduction

The security of software applications, from web-based applications to mobile services, is always at risk because of the open society of internet. As the amount of network throughput increases and security threat intensifies, intrusion detection systems(IDS) have drawn much attention in recent years. An IDS is a mechanism used to monitor system and network situations, collect useful data such as suspicious activities and environmental context, and analyze such data to detect malicious intentions. In general, intrusion detection approaches are classified as either Signature-based Intrusion Detection (SD) or Anomaly-based Intrusion Detection (AD). SD is the process to compare signature patterns of known attacks or threats against captured events for recognizing possible intrusions. AD is the process to find deviation from a known behavior, and construct profiles representing the normal or expected behaviors derived from monitoring regular activities, network connections, hosts or users over a period of time [1].

Existing intrusion detection systems are divided into five different types by Deepa & Kavitha [2]. They are: Network-based IDS, which monitors network traffic data; Host-based IDS, which analyzes host activities like system calls, application logs and so on; Stack-based IDS, which examines the packets as they go through the TCP/IP stack; Protocol-Based IDS, which monitors the protocol in use of the computing system; and Graph-Based IDS, which concerns intrusions that involve connections between many hosts or nodes.

The challenge of intrusion detection is to build effective predictive models with low error rates by utilizing and integrating various data resources. To achieve this goal, various approaches have been proposed. These include statistic-based, pattern-based, rule-based, state-based and heuristic-based approaches. As a statistic-based approach, Bayesian network (BN) has been widely used in intrusion detection field due to its robustness in modeling the joint distribution of random variables and reasoning under uncertainty.

Amor *et al.* [3], Vijayasarathy *et al.* [4] and Altwaijry & Algarny [5] built Naive Bayesian classifier, a type of simplified BN, to identify possible intrusions. However, Naive Bayes needs a strong assumption that the feature nodes in the model are independent from each other given the root node, which is not always the case in practice.

Kruegel *et al.* [6] proposed an event classification that makes full use of BNs and allows the modeling of inter-feature-node dependencies. They showed that these extensions improve the quality of the decision process and significantly reduce false alarms. Lu *et al.* [7] gave a two-stratum BNs-based anomaly detection and decision model for IDS. Laskey *et al.* [8] created an innovative human behavior model to model user queries and detect situations and insider threats to information systems using multi-entity BNs. In [9], An *et al.* used dynamic BNs to model temporal environments and detect any privacy intrusions. In these applications, the network model structures were manually constructed from domain knowledge without utilizing the training data that better reflects the real situation. To address this problem, Wee *et al.* [10] performed model selection by learning the BN structure from data. However, they conduct the model selection using heuristic methods, which usually select suboptimal models.

Further, most of the classifiers were trained and evaluated by the KDD Cup 99 dataset, which consists of about 0.5 million records [11]. A classifier trained with such a huge training dataset is usually capable of representing the probability distribution, thus achieves very good performance. However, obtaining such large-scale datasets can be challenging in practice, as it may take an unreasonably long time to collect the data resources. When the training dataset is small relative to the number of features considered, it is usually hard to select a single classifier model that properly represents the probability distribution of the model space. In such a situation, using a single model may lead to poor classification on future data.

To address the problems raised above, we built a Bayesian classifier for intrusion detection by Bayesian Model Averaging (BMA) over the k-best BN classifiers. Instead of selecting a single BN classifier, we perform model selection to find the top k BN classifiers according to a certain scoring metric. When future data points are classified, the decision is made by averaging over the prediction results of the k-best BN classifiers. The motivation of doing this is that multiple BNs are better than one BN in

representing the probability distribution of the model space, thus they offer better predictive power than one network, particularly in the domain where only small training datasets are available. To the best of our knowledge, this is the first attempt to employ BMA method in intrusion detection.

The rest of the paper is organized as follows: In Section II, we briefly introduce the concept of BN classifiers and BMA. We then introduce Bayesian Network Model Averaging classifier, which makes predictions by averaging over k-best BN classifiers. In Section III, we describe the NSL-KDD dataset from which our training and testing datasets are drawn. In Section IV, we outline the construction and evaluation of our BNMA classifier. In Section V, we illustrate the design of experiments and experimental results. In Section VI, we conclude with some discussions and ideas for future work.

II. Bayesian Networks and Bayesian Model Averaging

A. Bayesian Network Classifier

A Bayesian network(BN) G is a probabilistic graphical model that encodes a joint probability distribution over a set of variables $\mathbf{X} = \{X_1, X_2, ..., X_n\}$ based on conditional independencies [12]. It is a directed acyclic graph (DAG) where each node represents a random variable and an edge denotes a direct probabilistic dependency between the two connected nodes. For each node, there is a conditional probability distribution (CPD) containing the probabilities of the node taking different values given its parents' value. Formally, the DAG structure asserts that each node is conditionally independent of all non-descendants given its parent nodes. By these assertions, the BN compactly represents the joint probability distribution as

$$p(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} p(X_i|\mathbf{Pa}_G(X_i)), \tag{1}$$

where $\mathbf{Pa}_G(X_i)$ denotes the set of parent nodes of X_i in G, and $p(X_i|\mathbf{Pa}_G(X_i))$ specifies the conditional probability distribution (CPD) of X_i given $\mathbf{Pa}_G(X_i)$.

Figure 1 gives a simple example of BN that portrays the probabilistic relationships among binary variables *Polution* (*P*), *Smoker* (*S*), *Cancer* (*C*), *XRay* (*X*) and *Dyspnoea* (*D*). The table associated with each variable is called Conditional Probability Table, encoding the CPD of the variable given its parents. The joint probability distribution of the five variables can be written as

$$p(P, S, C, X, D) = p(P)P(S)p(C|P, S)p(X|C)p(D|C) \tag{2}$$

As the CPD can be calculated from the joint probability, a BN consisting of a class variable and feature variables is readily applicable to the classification task. Take the lung cancer network as an example, if we choose *Cancer* (*C*) as the class variable (value unobserved), we can compute the probability of $C = T$ given any observed value set (p, s, x, d) as

$$\begin{aligned} &p(C = T|p, s, x, d) \\ &= \frac{p(C = T, p, s, x, d)}{p(C = T, p, s, x, d) + p(C = F, p, s, x, d)} \end{aligned} \tag{3}$$

where $p(C = T, p, s, x, d)$ and $p(C = F, p, s, x, d)$ can be computed efficiently using Eq.(2). Similarly, we can compute $p(C = F|p, s, x, d)$. Then we decide the value of C by comparing $p(C = F|p, s, x, d)$ and $p(C = T|p, s, x, d)$. Note that this is a binary classification, easily generalized to multi-class classification by comparing the conditional probabilities of all values of the class variable.

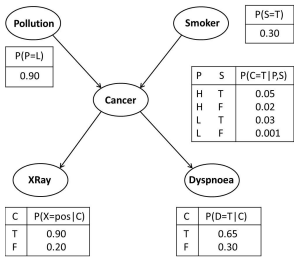

Figure 1. A simple example of Bayesian network: Lung cancer network.

The BN structure and its associated CPDs can be specified with the help of domain knowledge, e.g., the lung cancer network. However, in most cases, the network structures and CPDs are unknown due to the lack of domain knowledge. In these cases, a BN classifier can be learned from training data. The learning process contains structural learning and CPD estimation. In structural learning, a scoring metric is employed to evaluate the fitness of a structure in relation to the training data. Then, a search method is applied to find a good model [10] among possible structures. Since the number of possible structures is super-exponential with respect to the number of variables, finding the optimal structure is NP-hard [13]. Thus, some heuristic or approximate methods, such as greedy search, are used. However, the model structures selected in this way are often suboptimal. After the structure is constructed, the CPDs can be efficiently estimated using well-developed statistical methods such as Maximum Likelihood Estimation (MLE) or Bayesian Estimation [14].

B. Bayesian Model Averaging of Bayesian Network Classifiers

Model selection suffers from the lack of distinguishability of scoring metrics when the training data is sparse, i.e., the size of the dataset is small relative to the number of variables. In this case, there can be many distinct BNs fitting the training data equally well. Thus, using a single BN potentially leads to poor predictions on future data.

A promising solution to alleviate this problem is to employ BMA, which provides a principled approach to the model-uncertainty problem by integrating all possible models weighted by their respective posterior probabilities. Formally, given a training dataset D and a future data point x (a realization of the variable set $\mathbf{X}$), we compute the posterior probability of observing x as

$$p(x|D) = \sum_{G} p(x|G, D)p(G|D), \tag{4}$$

where $p(G|D)$ specifies the posterior probability of a BN G given the training data D. $p(G|D)$ can be computed from commonly used scores such as BDe score [15]. Then, $p(x|G, D)$ can be computed by Eq. (2), as the network structure G is fixed in the conditional setup.

Since computing Eq.(4) requires enumerating all possible networks, which is super-exponential with respect to the number of variables, it is not of practical use. One solution is to approximate this exhaustive enumeration by using a selected set of model structures in $\mathcal{G}$, i.e.,

$$p(x|D) \approx \frac{\sum_{G\in\mathcal{G}} p(x|G, D)p(G|D)}{\sum_{G\in\mathcal{G}} p(G|D)}. \tag{5}$$

Dash and Cooper [16] described an efficient solution to BMA for prediction over a set of BN structures. However, this approach is of limited applicability as it performs model averaging over only a restricted class of BNs consistent with a particular partial ordering and with bounded in-degree. Thus, only a small portion of probability density can be accounted for. Tian *et al.* [17] proposed to find the k-best BN structures and use them to approximately compute $p(h|D)$, i.e., the posterior probability of any hyperthesis h. They implemented this idea to address the problem of structure discovery in BNs, i.e., computing $p(f|D)$, the posterior probability of the presence of any structural feature f. (e.g., an edge, in BN structures). They showed that the approximation achieved reasonable accuracy and outperformed the classical sampling methods such as MCMC [18] for structure discovery in BNs.

In this study, we employ the k-best idea to address the problem of model averaging for prediction (classification). We select the k-best BNs $G^1, ..., G^k$, and use them to approximately compute $p(x|D)$ as shown in Eq.(6),

$$p(x|D) \approx \frac{\sum_{i=1}^{k} p(x|G^i, D)p(G^i|D)}{\sum_{i=1}^{k} p(G^i|D)}. \quad (6)$$

Once $p(x|D)$ is computed, we could build a classifier to predict the value of any class variable as shown in Section II-A. When $k = 1$, we select the best BN and use this single network to build a classifier. Thus, it is a special case of BMA.The optimal model selection is an NP-hard problem [13]. Thus, in existing applications of BN classifiers, heuristic or approximate methods are employed to find the models which are usually suboptimal. Silander *et al.* proposed a dynamic programming (DP) algorithm which is capable of finding the globally optimal BN in $O(n2^n)$ time [19]. Tian *et al.* [17] extended the DP algorithm to find the top k BNs. They demonstrated the applicability of the algorithm on networks with up to 20 variables. In this study, we employ this algorithm to select the k-best BNs. We then estimate the CPDs using Bayesian Estimation for each of the k-best networks that result in k discrete BN classifiers. Afterwards, we build our BNMA classifier by averaging the prediction results over the k-best BN classifiers.

III. Description of NSL-KDD Dataset

In previous IDS research, KDD Cup 99 dataset has been widely used to build and evaluate these systems [3] [10] [5] [20]. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment. However, KDD Cup 99 has two major issues that highly affect the assessment of the performance of evaluated systems [21]. The first deficiency is the huge number of redundant records in the training dataset. This deficiency will cause learning algorithms to be biased towards more frequent records. The second deficiency is that the existence of repeated records in the test set will cause the evaluation results to be biased towards favoring the methods with better detection rates on frequent records.

In our experiment, we use the NSL-KDD [21] dataset consisting of selected records of the complete KDD Cup 99 dataset with redundant and repeated records removed. As can be seen from the literature [21], the original KDD Cup 99 dataset is skewed and unproportionately distributed, training and testing directly on it can result in relatively high accuracy rate for different methods, making it difficult to effectively compare different classifiers.

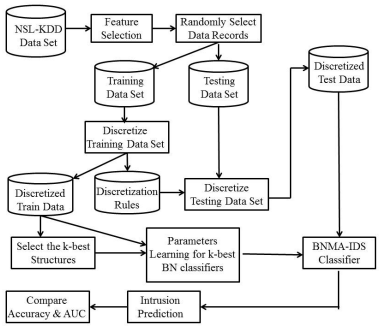

Figure 2. Flowchart illusrating the training and evaluation of BNMA classifier

Using this NSL-KDD dataset for evaluation is more objective and judicial as it does not suffer from either of the two problems mentioned above. The NSL-KDD dataset contains a training set with 125,973 records and a testing set with 22,544 records. Each of the datasets contain 41 attributes describing different features of the connection and a class label assigned to each either as attack or as normal.

IV. Construction and Evaluation of BNMA Classifier

BNMA can be used to train IDS based on intrusion dataset for detecting various types of intrusions, such as Denial of Services, User to Root attacks, Remote to Local attacks and so on. In this section, we demostrate how we construct and evaluate the BNMA classifier based on NSL-KDD dataset. Figure 2 illustrates the process from data processing to classifier training and evaluation. This BNMA classifier decides a data record to be normal or intrusive without concerning specific types of attacks. The whole process is elaborated in the following steps:

1) Select a subset out of a total 41 features from NSL-KDD dataset as the variables for classifier building.
2) Randomly sample partial datasets of varying sizes from the overall NSL-KDD training dataset as the training sets. The whole NSL-KDD testing dataset is then used as the testing set.
3) Perform data discretization on the continuous features in training and testing datasets using the information-preserving discretization method.
4) Find the k-best BN structures using the training dataset, and estimate the CPDs for each networks using Bayesian Estimation. This results in k independent BN classifiers.
5) Combine the k BN classifiers into a Bayesian classifier using BMA.
6) Apply the Bayesian classifier to the testing dataset, calculate the accuracy and Area Under ROC (AUC).
7) Conduct four groups of experiments by repeating steps 2-6 using different training sets. The results for each classifier and each configuration of different size are then averaged over those four groups of experiments.

In the upcoming subsections, we give details on the process of feature selection, data discretization, and classifier training and evaluation.

A. Feature Selection

Feature selection is an indispensable pre-processing step because extraneous features not only add burden to the computation but also confound the detection process. The NSL-KDD dataset

contains 41 features, some of which may be redundant and contribute less than the others to the detection process. Feature selection and feature deduction have been a very popular topic in intrusion detection field for identifying important input features to build computationally efficient and effective IDS. Singh & Silakari [22] proposed an ensemble approach for feature selection of the Cyber Attack dataset. Chebrolu *et al.* [23] and Olusola *et al.* [24] specifically analyzed the feature relevance on the KDD Cup 99 dataset. In [23], Markov blanket model was used to select the feature set and it was shown that a selected set of 12 features can achieve better predictive accuracy than when the whole set of 41 features is used. Our main focus in this paper is to compare the methods based on the same datasets with the same feature set, rather than to study the KDD data. And we are also particularly interested in studying their performance using a carefully selected representative subset rather than the full features. Therefore, in our experiments we used the 12 features suggested in [23]. Those features are described in the following Table I.

Table I
LIST OF SELECTED FEATURES

FEATURE NAME	DESCRIPTION	TYPE
service	network service on the destination, e.g., http, telnet, etc.	Discrete
src_bytes	number of data bytes from source to destination	Continuous
dst_bytes	number of data bytes from destination to source	Continuous
logged_in	1 if successfully logged in; 0 otherwise	Discrete
count	number of connections to the same host as the current connection in the past two seconds	Continuous
srv_count	number of connections to the same service as the current connection in the past two seconds	Continuous
serror_rate	% of connections with errors (refer to the same-host connection)	Continuous
srv_serror_rate	% of connections with errors (refer to the same-service connection)	Continuous
srv_diff_host_rate	% of connections to different hosts	Continuous
dst_host_count	sum of connections to the same destination IP address	Continuous
dst_host_srv_count	sum of connections to the same destination port number	Continuous
dst_host_diff _srv_rate	the percentage of connections to different services, among the connections aggregated in dst_host_count (32)	Continuous

B. Data Discretization

As shown in Table I, some of the selected features take continuous values. However, current implementation of BN classifiers can only handle discrete values. Thus, continuous features need to be discretized before being used to build a classifier. On the other hand, discretization can often make continuous features easier to understand and interpret, and produce faster learning models. Many learning models have been shown to perform better by discretizing continuous features [25]. Based on our knowledge, there are two types of commonly used discretization methods in many of the IDS [26] [24] [10], that is, the unsupervised discretization algorithms, e.g., equal intervals, equal frequencies, and the supervised discretization algorithms, e.g., maximum entropy discretization, χ^2 discretization, CAIM, etc. In our experiment, we adopted a discretization algorithm named CACC [27], which was a a static, global, incremental, supervised and top-down discretization algorithm. This information-theoretic algorithm extended the idea of contingency coefficient, combined with the greedy method, and was empirically shown to be promising in terms of accuracy, execution time, etc. Data discretized using such discretization scheme have much less information loss, thus better represent the distribution of original data compared to the ones discretized using other unsupervised discretization methods.

C. Classifier Training and Evaluation

One purpose of this study is to evaluate the performance of various classifiers with respect to varying sizes of the training dataset. Thus, we prepare several training datasets containing 500, 1000, 2000, 5000, 10000, 20000, 30000, 40000 records, respectively. We train and build a Bayesian classifier from each of the training sets by BMA over the k-best BN classifiers as described in Section II. For each training dataset, we build the Bayesian classifier by setting k to various values. We then evaluate the performance of the classifier with respect to these different k values. When $k = 1$, the classifier is equivalent to a single BN classifier. The larger k is, the more models are employed for model averaging, which potentially leads to better predictive power.

We evaluate all classifiers on the same testing dataset. We compute the accuracy as the percentage of correctly classified records. Note that this is a binary classification problem, i.e. an attack or normal. A record is classified as an attack if the conditional probability of being an attack given the observation of other features is greater than 0.5; it is classified as normal otherwise. In addition to accuracy, we compute AUC as the area under the Receiver Operating Characteristic (ROC) curve, which is an estimate of the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. Since AUC does not depend on the classification threshold used, it is widely recognized as a better measure than accuracy, which is based upon a single classification threshold.

For comparison, we also build Naive Bayes classifiers and BN classifiers, which are selected by using the greedy hill-climbing search method. As described before, we repeated the experiment four times on different training and testing data sets and reported the average accuracy and AUC.

V. EXPERIMENTAL RESULTS

Figure 3 compares the detection accuracy of Naive Bayes (NB), Bayes Network built using greedy search (BN-Greedy) and the BNMA ($k = 1$) by size of the training dataset. First, we observe that the accuracy is approximately a non-decreasing function of training sample size. This is understandable since a larger training sample usually produces a classifier with better predictive power. The accuracies of NB and BN-Greedy are comparable to each other, while the BNMA classifier built using the best BN ($k = 1$) is significantly better than the two classifiers. Further, the BNMA ($k = 1$) trained classifiers using a small training set (2000) even outperforms the NB and BN-Greedy classifiers trained using a very large training set (40000). The improvement is also significant when AUCs are compared (see Figure 4). This indicates that the BNMA classifier can achieve reasonably good predictive power even when a small training dataset is used.

In another set of experiments, we evaluate the BNMA classifier with respect to various k values. Table II compares the detection accuracy by k value and training set size. Table III compares the

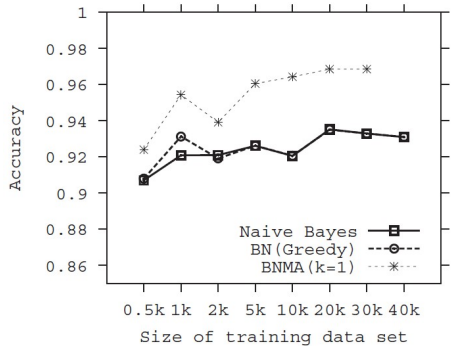

Figure 3. Comparison of detection accuracy by size of training set

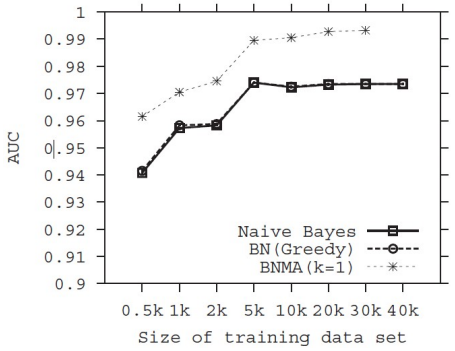

Figure 4. Comparison of AUC by size of training set

AUC by k value and training set size. It is shown that with the increase of k, both accuracy and AUC increase. AUC has a more obvious increase than accuracy. However, this improvement is not as significant as that in comparing BNMA ($k = 1$) with NB and BN-Greedy. The most significant improvement is in Table III, for sample size 500, where the AUC jumps from 0.9615 for $k = 1$ to 0.9733 for $k = 200$. With the increase of sample size, the improvement decreases. This demonstrates that the BNMA is particularly effective on small sample sizes.

Table II
ACCURACY COMPARISON BY k VALUE AND SIZE OF TRAINING SET

	k Value				
	k=1	**k=10**	**k=50**	**k=100**	**k=200**
500	92.40%	92.40%	92.40%	92.40%	92.40%
1000	95.43%	95.43%	95.43%	95.56%	95.43%
2000	93.92%	93.97%	93.97%	93.97%	93.97%
5000	96.06%	96.14%	96.11%	96.14%	96.16%
10000	96.43%	96.43%	96.44%	96.47%	96.46%
20000	96.86%	96.86%	96.87%	96.87%	96.87%
30000	96.86%	96.86%	96.85%	96.86%	96.92%

Table III
AUC COMPARISON BY k AND SIZE OF TRAINING SET

	k Value				
	k=1	**k=10**	**k=50**	**k=100**	**k=200**
500	0.9615	0.9706	0.9733	0.9733	0.9733
1000	0.9705	0.9718	0.9728	0.9728	0.9730
2000	0.9738	0.9738	0.9740	0.9744	0.9745
5000	0.9895	0.9898	0.9898	0.9900	0.9900
10000	0.9905	0.9905	0.9905	0.9908	0.9908
20000	0.9928	0.9928	0.9928	0.9928	0.9930
30000	0.9933	0.9933	0.9933	0.9933	0.9933

To investigate why the predictive power does not change very much with respect to the variation of the k value, we examine the structures of the top 10 networks produced using a training set with size 10000. The network on the right in Figure 5 illustrates the consensus structure for the 10 structures. It is surprising that all 10 structures share the same skeleton with minor differences in the direction of the edges. This indicates that the top structures represent similar distribution, and thus make similar predictions for new data points. We can speculate that the top 200 structures may have very similar structure. For comparison, we also depict the BN structure learned using greedy hill climbing search method (the left network in Figure 5). It is easily observed that this structure is significantly sparser (fewer edges) than the consensus structure. This explains why the structure selected with heuristic method is suboptimal, because it fails to identify many important dependencies among the feature nodes which can be captured by our method of using BNMA classifier. It also explains why the best BN ($k = 1$) has significantly better predictive power than the BN-Greedy classifier.

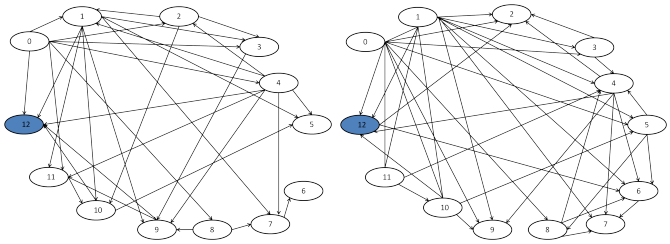

Figure 5. Comparison of BN models learned using heuristic method and BMA. Left: Bayesian network trained on sample size 10000 using greedy hill climbing search method. Right: a consensus network built from the top 10 networks trained on sample size 10000. The correspondences between the nodes and the features are : 0-service; 1-src_bytes; 2-dst_bytes; 3-logged_in; 4-count; 5-srv_count; 6-serror_rate; 7-srv_serror_rate; 8-srv_diff_host_rate; 9-dst_host_count; 10-dst_host_srv_count; 11-dst_host_diff_srv_rate; 12-class (intrusion or not intrusion). Note that the class variable is shadowed. Directed edges existing in all 10 structures are depicted as solid arrows. The set of edges that exist in all structures but with various directions are depicted as solid lines.

VI. CONCLUSION AND FUTURE WORK

We proposed a Bayesian classifier using BMA of k-best BN classifiers, called BNMA classifier, for intrusion detection. Previous IDS using BN classifier has two problems. First, the BN structure is selected using heuristic methods, which usually return suboptimal models. Second, previous classifiers are trained and evaluated using a very large training dataset, which is usually hard to collect within a short time period. We used DP algorithm to find the globally k-best structures to build a Bayesian classifier by BMA. We showed that the BNMA classifier has significantly better predictive power than Naive Bayes and BN classifier built using heuristic method. Even the classifier trained using smaller dataset outperforms the other two classifiers trained using larger dataset. We conclude that our BNMA classifier is particularly effective in detecting intrusions when only a few training records are available.

We also show that with the increase of k, i.e., more BN classifiers are used for model averaging, the better predictive power it can achieve. However, this improvement is not that significant, since the top structures actually share a very similar structure. This means the problem size (12 feature variables) is still not that large compared to the sample sizes examined. One question that users may ask is, what is the k value we should

use? The answer is the larger, the better. However, it takes more time to train and integrate over larger number of classifiers. In this study, we consider 12 feature variables and $k = 100$ is already enough. The k value that should be selected depends on the problem size, i.e., the number of feature variables used to build the model. Thus, our future work is to select a larger set of feature variables for model building. Since BN is able to inherently do feature selection through its conditional dependency assertions, using a larger set of features should not significantly impact the performance. However, with a larger set of feature variables, it may need larger k, i.e., integrating over more models to achieve reasonably good predictive power.

Another area for future work is based on the observation that intrusions happen in dynamic environments, thus they themselves could be time-series data. An *et al.* [9] proposed to use dynamic BNs to model the temporal environment. However, the problems faced by the static BN classifier persist in dynamic BNs. Thus, it is a challenge to perform model averaging in the temporal environment setup.

In summary, since it uses less data while still achieving comparable or better predictive power, our BNMA classifier can save a huge amount of time on collecting training data records so that it can catch the intrusion more promptly and more accurately to avoid loss due to intrusion.

REFERENCES

[1] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[2] A. Deepa and V. Kavitha, "A comprehensive survey on approaches to intrusion detection system," *Procedia Engineering*, vol. 38, pp. 2063–2069, 2012.

[3] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naïve bayes vs decision trees in intrusion detection systems," in *Proceedings of the 2004 ACM Symposium on Applied Computing*.

[4] R. Vijayasarathy, S. Raghavan, and B. Ravindran, "A system approach to network modeling for ddos detection using a naïve bayesian classifier," in *2011 Third International Conference on Communication Systems and Networks (COMSNETS)*, 2011, pp. 1–10.

[5] H. Altwaijry and S. Algarny, "Bayesian based intrusion detection system," *Journal of King Saud University - Computer and Information Sciences*, vol. 24, no. 1, pp. 1–6, 2012.

[6] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, "Bayesian event classification for intrusion detection," in *Proceedings of the 19th Computer Security Applications Conference*, 2003, pp. 14–23.

[7] L. Huijuan, C. Jianguo, and W. Wei, "Two stratum bayesian network based anomaly detection model for intrusion detection system," in *Proceedings of the 2008 International Symposium on Electronic Commerce and Security*, ser. ISECS '08, 2008, pp. 482–487.

[8] K. Laskey, G. Alghamdi, X. Wang, D. Barbará, T. Shackelford, E. Wright, and J. Fitzgerald, "Detecting threatening behavior using bayesian networks," in *Conference on Behavioral Representation in Modeling and Simulation – BRIMS*, Arlington, VA, USA, 2004.

[9] X. An, D. N. Jutla, and N. Cercone, "Privacy intrusion detection using dynamic Bayesian networks," in *International Conference on Electronic Commerce*, 2006, pp. 208–215.

[10] Y. Y. Wee, W. P. Cheah, S. C. Tan, and K. Wee, "Causal discovery and reasoning for intrusion detection using bayesian network," *International Journal of Machine Learning and Computing*, vol. 1, no. 2, 2011.

[11] "KDDCUP 1999 data," http://kdd.ics.uci.edu/databases/kddcup99.

[12] D. Heckerman, "A tutorial on learning with bayesian networks," Microsoft Research Advanced Technology Division, Tech. Rep., 1996.

[13] D. M. Chickering, D. Geiger, and D. Heckerman, "Learning Bayesian networks: Search methods and experimental results," in *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*. Ft. Lauderdale, FL: Society for Artificial Intelligence and Statistics, Jan. 1995, pp. 112–128.

[14] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[15] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *Machine Learning*, vol. 20, no. 3, pp. 197–243, Sep. 1995.

[16] D. Dash and G. F. Cooper, "Model averaging for prediction with discrete bayesian networks," *The Journal of Machine Learning Research*, vol. 5, pp. 1177–1203, 2004.

[17] J. Tian, R. He, and L. Ram, "Bayesian model averaging using the k-best bayesian network structures," in *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*. Corvallis, Oregon: AUAI Press, 2010, pp. 589–597.

[18] D. Eaton and K. Murphy, "Bayesian structure learning using dynamic programming and mcmc," in *Proceedings of Conference on Uncertainty in Artificial Intelligence*, 2007.

[19] T. Silander and P. Myllymäki, "A simple approach for finding the globally optimal bayesian network structure," in *Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2006, pp. 445–452.

[20] N. Devarakonda, S. Pamidi, V. V. Kumari, and A. Govardhan, "Intrusion detection system using bayesian network and hidden markov model," *Procedia Technology*, vol. 4, no. 0, pp. 506–514, 2012.

[21] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *IEEE Symposium on Computational Intelligence for Security and Defense Applications,(CISDA).*, 2009, pp. 1–6.

[22] S. Singh and S. Silakari, "An ensemble approach for feature selection of cyber attack dataset," *International Journal of Computer Science and Information Security*, vol. 6, no. 2, pp. 297–302, 2009.

[23] S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," *Computers & Security*, vol. 24, no. 4, pp. 295–307, 2005.

[24] A. A. Olusola, A. S. Oladele, and D. O. Abosede, "Analysis of kdd 99 intrusion detection dataset for selection of relevance features," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1. San Francisco, USA: WCECS, 2010.

[25] S. Kotsiantis and D. Kanellopoulos, "Discretization techniques: A recent survey," *GESTS International Transactions on Computer Science and Engineering*, vol. 32, no. 1, pp. 47–58, 2006.

[26] G. Kou, Y. Peng, Z. Chen, and Y. Shi, "Multiple criteria mathematical programming for multi-class classification and application in network intrusion detection," *Information Sciences*, vol. 179, no. 4, pp. 371–381, 2009.

[27] C.-J. Tsai, C.-I. Lee, and W.-P. Yang, "A discretization algorithm based on class-attribute contingency coefficient," *Information Sciences*, vol. 178, no. 3, pp. 714–731, 2008.