Expert Systems with Applications 39 (2012) 13492–13500

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier

Levent Koc *, Thomas A. Mazzuchi, Shahram Sarkani

Department of Engineering Management and Systems Engineering at The George Washington University, Washington, DC, USA

ARTICLE INFO

Keywords:
Intrusion detection systems
Data mining
Multiclass classification
Hidden Naïve Bayes (HNB)
Denial-of-services (DoS)

ABSTRACT

With increasing Internet connectivity and traffic volume, recent intrusion incidents have reemphasized the importance of network intrusion detection systems for combating increasingly sophisticated network attacks. Techniques such as pattern recognition and the data mining of network events are often used by intrusion detection systems to classify the network events as either normal events or attack events. Our research study claims that the Hidden Naïve Bayes (HNB) model can be applied to intrusion detection problems that suffer from dimensionality, highly correlated features and high network data stream volumes. HNB is a data mining model that relaxes the Naïve Bayes method's conditional independence assumption. Our experimental results show that the HNB model exhibits a superior overall performance in terms of accuracy, error rate and misclassification cost compared with the traditional Naïve Bayes model, leading extended Naïve Bayes models and the Knowledge Discovery and Data Mining (KDD) Cup 1999 winner. Our model performed better than other leading state-of-the art models, such as SVM, in predictive accuracy. The results also indicate that our model significantly improves the accuracy of detecting denial-of-services (DoS) attacks.

1. Introduction

According to recent Internet security reports, the volume and sophistication of targeted network attacks has increased substantially in recent years. The increasing number of threats against and vulnerabilities of a diverse set of targets, such as military, government and commercial network systems, require increasing situational awareness and various cyber security measures (Baker, Filipiak, & Timlin, 2011; Fossi et al., 2011). Intrusion detection is a security measure that helps to identify a set of malicious actions that compromise the integrity, confidentiality, and availability of information resources (Dokas et al., 2002). Intrusion detection is a difficult problem because of the tradeoff considerations of detection accuracy, detection speed, the dynamic nature of the networks and the available processing power for processing high volumes of data from distributed networked systems (Kabiri & Ghorbani, 2005). These considerations led to intrusion detection research, which includes both misuse detection and anomaly detection. Misuse detection relies on a learning algorithm that is trained by a dataset in which each instance is labeled as either a normal event or an intrusion. Although the algorithm cannot detect novel attacks that were not included in the training set, it can be automatically retrained with the new attack instances through a new training dataset (Kumar & Spafford. 1994). Anomaly detection builds models of normal network events and detects the events that deviate from these models (Denning, 1987). This method can detect new types of attack events because it only relies on known normal events. Despite its advantages, the anomaly detection method suffers from a high rate of false alarms due to previously unobserved normal events. Hybrid models utilize both misuse detection and anomaly detection approaches to improve the prediction performance (Depren, Topallar, Anarim, & Ciliz, 2005; Zhang, Zulkernine, Haque, 2008).

Data mining explores and analyzes large datasets to discover understandable and useful patterns and models (Hand, Mannila, & Smyth, 2001). The data mining of network events is often leveraged to differentiate attack events from normal events by using various methods, such as outlier detection (Lazarevic, Ertoz, Kumar, Ozgur, & Srivastava, 2003), clustering data into categories (Frank, 1994), classifier models for predicting the categories, and association-rule-based models.

Classification is the identification of the category labels of instances that are typically described by a set of features (attributes) in a dataset. Learning classifier models learn from the given training data and infer the class labels for the instances of the new data. Scholars have applied numerous classifier models to the intrusion detection problem, including rule-based detection (Lunt, 1989), neural networks (Cannady, 1998; Lippmann & Cunningham, 2000; Zhang, 2001), fuzzy logic (Bridges & Vaughn, 2000), the hidden Markov model (Bo, Hui-Ye, & Yu-Hang, 2002), the random

* Corresponding author. Tel.: +1 5717232231.
E-mail address: lkoc@gwu.edu (L. Koc).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.07.009

forest model (Zhang, Zulkernine, Haque, 2008), data mining (Lee, Stolfo, & Mok, 1999; Wu & Yen, 2009) and Bayesian analysis (Barbara, Wu, & Jajodia, 2001).

Our intrusion detection model is a multinomial classifier that is used to classify network events as normal or attack events, such as DoS, probe, U2R, and R2L. The model is based on a new data mining method called Hidden Naïve Bayes (HNB). The HNB classifier model is applied to several datasets and shows promising results compared with the traditional Naïve Bayes and its extended methods (Jiang, Zhang, & Cai, 2009). As to our knowledge, no systematic research exists addressing the applicability and effects of using HNB based classifier in intrusion detection domain. Our experimental research study explores the traditional Naïve Bayes and leading structurally extended Naïve Bayes approaches including the new HNB approach. To our knowledge, there is no other comparative research study in intrusion detection domain looked at the Naïve Bayes and structurally extended Naïve Bayes approaches comprehensively. In our study, we augmented the Naïve Bayes and extended Naïve Bayes methods with the leading discretization and feature selection methods to increase the accuracy and decrease the resource requirements of intrusion detection problem. Based on the results of our study, HNB based classifier model stands out as simple and practical intrusion detection system with better predictive accuracy and cost.

The Naïve Bayes method, which is the simplest form of a Bayesian network, is a popular data mining method that has been applied to many domains, including intrusion detection. The method's simplicity relies on the assumption that all of the features are independent of each other. The HNB method, which relaxes this assumption, has been successfully applied to web mining (Bo, Qiurui, Zhong, & Zengmei, 2009; Xin, Rongyan, Xian, & Rongfang, 2007). Background information on the Naïve Bayes methods and its extensions are presented in the Related Work section of our paper.

Because of its good performance in earlier work in other domains, this study applies the HNB classifier model to the intrusion detection problem. In the Research Method section, we present the rationale behind our use of the HNB model and introduce our model and the conceptual framework.

We used the classic KDD Cup 1999 (KDD'99) intrusion detection dataset to test our claim that with respect to the intrusion detection problem, the HNB method outperforms the traditional Naïve Bayes method, the leading structurally extended Naïve Bayes methods and the winning KDD'99 method in terms of detection accuracy, error rate and misclassification cost. Because of the challenges associated with the dataset, we preprocessed the dataset with several discretization and feature selection methods. The KDD'99 dataset and its properties are presented in the KDD'99 dataset section.

In the Experiments and Results section, we explain and discuss our experimental setup. We also present and compare the results obtained with the HNB classifier with those obtained with the traditional Naïve Bayes classifier, the classifiers based on the structurally extended Naïve Bayes methods and the KDD'99 winner as a common benchmark. Finally, we compare our results with those of the state-of-the-art models from earlier studies. The conclusions of our research study are presented in the last section.

2. Related Work

A classifier function takes each instance of a dataset and maps it to a distinct class by prediction. In the intrusion detection case, a binary classifier assigns the network events to either a normal event class or a malicious event class, whereas a multiclass classifier further assigns the malicious event class to DoS, probe, U2R or R2L classes.

Similar to many other data mining techniques, building the optimum classifiers requires two important tasks: the selection of the input feature (attribute) from a potentially large set of possible features in a given dataset and the selection of the model (optimization) based on the selected features (Hofmann & Sick, 2003). Selecting the right features is challenging, but it must be performed to reduce the number of features for the sake of efficient processing speed and to remove the irrelevant, redundant and noisy data for the sake of predictive accuracy (Huan & Lei, 2005). A multiclass classifier G needs to map the feature space with A features into C classes on a dataset D, which consists of $\{E_1, E_2, \ldots, E_i, \ldots, E_t\}$ instances.

2.1. Bayesian network classifiers

The Bayesian network is one of the most common classifiers for statistical data mining methods. The Bayesian network is based on a directed acyclic graph, where nodes represent attributes, and arcs represent attribute dependencies. In this method, the conditional probabilities for each node, which are based on its parents' attributes, quantify the attribute dependencies. A features, which consist of attributes $\{A_1, A_2, \ldots, A_i, \ldots, A_n\}$, are represented as nodes in a Bayesian network, and $(a_1, a_2, \ldots, a_i, \ldots, a_n)$ are the attribute values of an instance E_i. The class variable C is represented as the top node in a Bayesian network, and c represents the value that C takes for instance E. The Bayesian network classifier can be defined as

$$c(E) = \arg\max_{c \in C} P(c)P(a_1, a_2, \ldots, a_n|c). \quad (1)$$

Naïve Bayes classifiers. The simplest form of a Bayesian network classifier is the Naïve Bayes (NB) classifier, in which all of the attributes are naively assumed to be independent given the class shown in Fig. 1 and defined in (3).

$$P(E|c) = P(a_1, a_2, \ldots, a_n|c) = \prod_{i=1}^{n} P(a_i|c) \quad (2)$$

Although the conditional independence assumption leads to biased posterior probabilities, a Naïve Bayes classifier is easy to construct because of the computational simplicity of reaching $P(C)$ and $P(a_i|c)$, and its accuracy performance is comparable with that of classification trees and neural networks (Langley, Iba, & Thompson, 1992).

$$c(E) = \arg\max_{c \in C} P(c) \prod_{i=1}^{n} P(a_i|C). \quad (3)$$

The Naïve Bayes classification model is one of the most popular models because of its simplicity and computation efficiency, both of which are inherited from its conditional independence assumption property, as well as its good performance on datasets for which

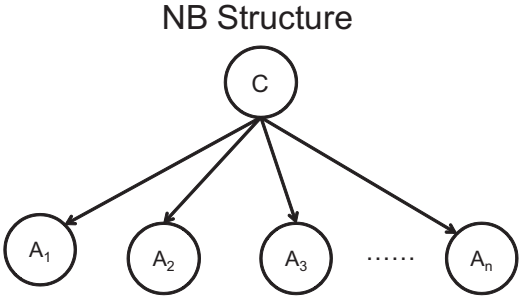

Fig. 1. Naïve Bayes structure.

this property is fairly accurate. However, the model does not perform well if this assumption property is not satisfied, as observed in datasets (Yaguang, Songnian, & Yafeng, 2011) that have complex attribute dependencies, such as the KDD'99 intrusion detection dataset. The findings also indicate that the Naïve Bayes model is not as accurate for large datasets (Kohavi, 1996).

One of the leading examples of the application of the Bayesian method in the intrusion detection domain can be found in research based on an anomaly detection system called Audit Data Analysis and Mining (ADAM). ADAM relies on pseudo-Bayes estimators to estimate the prior and posterior probabilities of new attacks and then uses these probabilities to construct a Naïve Bayes classifier for classifying normal and attack events without prior knowledge about new attacks (Barbara et al., 2001).

In the last two decades, many studies have focused on relaxing the conditional independence assumption of Naïve Bayes models. The intrusion detection model introduced in this study utilizes structure extension and feature selection approaches. The following section introduces these two approaches and briefly explains other leading approaches, such as attribute weighting, local learning and data expansion.

2.1.1. Structure extension

In this approach, the Naïve Bayes structure is extended by using directed arcs to explicitly represent attribute dependencies as a Bayesian network model. Because learning the optimal structure of the Bayesian network model is an NP-hard problem (Chickering, Heckerman, & Meek, 2004), some restrictions or heuristics must be applied to make the approach practical, such as those applied in the tree-augmented Naïve Bayes (TAN) (Friedman, Geiger, & Goldszmidt, 1997), averaged one-dependence estimators (AODE) (Webb, Boughton, & Wang, 2005), Weightily AODE (Jiang, 2006) and HNB (Jiang et al., 2009) models.

The TAN learning algorithm is an extension of Naïve Bayes in which an attribute node might have at most one additional parent node other than the class node, as shown in Fig. 2. The additional arc represents the interaction between the attribute nodes. This model is based on conditional mutual information, defined by Friedman et al. (1997) as

$$I_P(X;Y|Z) = \sum_{x,y,z} P(x,y,z)\log\frac{P(x,y|z)}{P(x|z)P(y|z)} \quad (4)$$

where x, y and z are the values of variables X, Y and Z, respectively. $I_p(A_i;A_j|C)$ is computed for each attribute pair and represents the weight of the arc's connection to attribute nodes A_i and A_j on the Bayesian network. With these values, the maximum weighted spanning tree is constructed (Jiang et al., 2009).

In the AODE learning algorithm, all of the predictions of one dependence estimator are aggregated, where each attribute has one correlated attribute (Webb et al., 2005). This algorithm utilizes the bagging method (Breiman, 1996) to reduce the variance of the predictions. The Jiang study introduced the Weightily AODE (WAODE) algorithm by assigning different weights to these one-dependence classifiers (Jiang & Zhang, 2006).

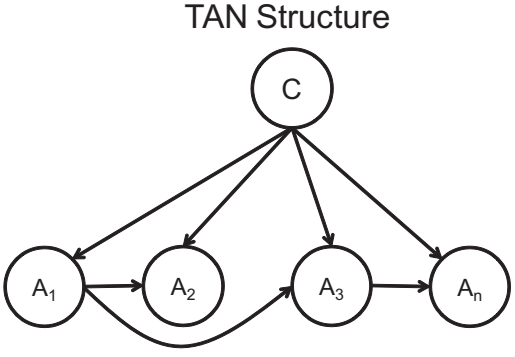

Fig. 2. TAN structure.

2.1.2. Feature selection

This approach removes redundant or irrelevant features from the dataset to prevent decreases in classification accuracy and unnecessary increases in computational costs (Blum & Langley 1997). There are three major feature selection approaches: (1) embedded, (2) wrapper and (3) filter methods. Embedded methods are embedded in specific mining methods, such as random forests, in which the importance of each feature is estimated through out-of-bag data (Breiman, 2001). Wrapper methods use the feedback received from a specific classifier to evaluate the quality of the feature subset. Selective Bayesian classifiers (SBC) (Langley, 1994) and evolutional Naïve Bayes (ENB) (Jiang, Zhang, Cai, & Su, 2005) are two wrapper methods that use the classification accuracy of the Naïve Bayes method to evaluate alternative feature subsets during their search through the entire feature space. Finally, filter methods rely on the general characteristics of the training data. Filter methods are used to pre-select the feature subset independent of any classifier method. Although wrapper models tend to be more accurate than filter methods, filter methods are computationally less expensive and do not rely on the performance of a specific classifier.

2.1.2.1. Filter methods. Because this classifier-independent feature selection method only relies on the statistical characteristics of the training data, its lower computational cost makes it essential for large datasets with large feature spaces, such as the KDD'99 intrusion detection dataset. Although there are numerous filter methods, this study examines three major methods: (1) correlation-based feature selection, (2) consistency-based filter and (3) INTERACT.

The correlation-based feature selection (CFS) method (Hall, 1999) ranks and selects the feature sets with biases towards subsets containing features that are highly correlated with the class and uncorrelated with each other. The method relies on a correlation-based heuristic evaluation function to ignore the features that are irrelevant because of their low correlation with the class. Additionally, redundant features are screened out because of their high correlation with the remaining features. A feature is selected based on the extent to which it predicts classes in the areas of the instance space that are not already predicted by other features.

The consistency based filter (CONS) method (Dash & Liu, 2003; Huan & Setiono, 1997) uses an inconsistency criterion that specifies the extent to which the dimensionally reduced data can be accepted. The algorithm generates a random subset in each round. If the random subset contains fewer features than the current best subset, the inconsistency criterion of the data with the random subset of features is compared with that of the data with the current best subset. If the new subset is more consistent than the current best subset, the latter is replaced by the new set (Hall, 1999).

INTERACT (INT) is a filter algorithm (Zhao & Liu, 2007) that searches interacting features in two steps. In the first step, the features are ranked in descending order by their symmetrical uncertainty (SU) values. SU is defined as the ratio between the information gain and the entropy of two features and is used as a correlation measure to evaluate the relevance of individual features. In the second step, a backward elimination process is applied to features with low consistency contribution, which indicates the extent to which the consistency will be affected by the elimination of a feature.

2.1.3. Other approaches

Although our model does not use the following approaches, we briefly introduce these methods to be consistent and comprehensive in our efforts to address the effects of relaxing the conditional independence assumption of Naïve Bayes. Some of the methods based on these approaches are also compared (in terms of performance) with our model in this study.

In the attribute weighting approach, each attribute's weight is determined according to its contribution to the classification. The weighted Naïve Bayes (WNB) model uses this approach and is defined as

$$c(E) = arg\max_{c\in C} P(c)\prod_{i=1}^{n} P(a_i|C)^{w_i}, \tag{5}$$

where w_i is the weight of attribute A_i. In this approach, the issue of how the weights are learned is significant and is an important research area (Deng, Wang, & Wang, 2007; Hall, 1999; Jiang & Zhang, 2006; Jiang et al., 2009).

In the local learning approach, the Naïve Bayes model is built on a subset of the training dataset. Local learning utilizes the idea that the negative impact of the conditional independence assumption in a subset is smaller than that in the entire dataset. Local learning is more accurate than other methods on larger datasets and allows new models that are embedded with each other to be created. NBTree (Kohavi, 1996) and DTNB (Hall, 2008) models combine the Naïve Bayes and decision tables, whereas the locally weighted Naïve Bayes (LWNB) (Frank, Hall, & Pfahringer, 2003), lazy Bayesian rule (LBR) and selective neighborhood Naïve Bayes (SNNB) (Xie, Hsu, Zongtian, & Lee, 2002) models are leading Naïve Bayes models that are embedded with the *k*-nearest neighbors model.

The data expansion approach tackles the high variance problem in learning due to limited training data by adding more instances with the same pattern if the underlying distribution of the dataset is known. Instance-cloned Naïve Bayes (ICNB) (Jiang, Wang, Zhang, Cai, & Huang, 2008) expands the training dataset by cloning certain training instances based on similarity.

2.2. Application of structurally extended Naïve Bayes classifiers to intrusion detection problem

As shown in Table 1, some of the extended Naïve Bayes methods, including the ones that were explained in the earlier section, have already been applied in the intrusion detection domain. For example, the TAN method was applied using the KDD'99 dataset (Benferhat, Boudjelida, & Drias, 2008), and the AODE method was recently applied to the intrusion detection problem (Baig, Shaheen, & AbdelAal, 2011). In the AODE method, a feature subset is selected via an algorithm called the Group Method for Data Handling (GMDH), and a small subset of the KDD'99 dataset is used. The semi-Naïve Bayes method, a hybrid of Naïve Bayes and decision trees (DTNB), is proposed for intrusion detection problems in the Panda study (Panda & Patra, 2009).

Although all of these studies evaluated the results with different evaluation methods, the results of both studies were promising, and the aforementioned methods performed better than the traditional Naïve Bayes method. To the best of our knowledge, the recently introduced HNB method has not been applied to the intrusion detection problem.

Table 1
Research on **Naïve** Bayes and its applications to the intrusion detection problem.

Naïve Bayes and variations based on structure extension	Original model	Applied to intrusion detection problem
Naïve Bayes	Langley et al. (1992)	Barbara et al. (2001)
Tree-augmented Naïve Bayes (TAN)	Friedman et al. (1997)	Benferhat et al. (2008)
Averaged one-dependence estimators (AODE)	Webb et al. (2005)	Baig et al. (2011)
Semi-Naïve Bayesian (DTNB)	Hall and Frank (2008)	Panda and Patra (2009)
Hidden Naïve Bayes (HNB)	Jiang et al. (2009)	

2.3. Hidden Naïve Bayes classifiers

An extended version of the Naïve Bayesian classifier is the hidden Naïve Bayes (HNB) classifier, which relaxes the conditional independence assumption imposed in the Naïve Bayesian model. The HNB model relies on the creation of another layer that represents a hidden parent of each attribute; this hidden parent combines the influences from all of the other attributes (Jiang et al., 2009), as shown in Fig. 3.

In the HNB model, each attribute A_i has a hidden parent A_{hpi}, where $i = 1, 2, \ldots, n$ represents the weighted influences from all of the other attributes, as shown with the dashed circles. The joint distribution is defined as

$$P(A_1, \ldots, A_n|C) = P(C)\prod_{i=1}^{n} P(A_i|A_{hp_i}, C), \tag{6}$$

where

$$P(A_i|A_{hp_i}, C) = P(C)\sum_{j=1, j\neq 1}^{n} W_{if} * P(A_i|A_j, C). \tag{7}$$

The HNB classifier can be defined as

$$c(E) = \arg\max_{c\in C} P(c)\prod_{i=1}^{n} P(a_i|a_{hp_i}, c). \tag{8}$$

where

$$P(a_i|a_{hp_i}, c) = P(c)\sum_{j=1, j\neq 1}^{n} W_{ij} * P(a_i|a_j, c). \tag{9}$$

One approach for determining the weights W_{ij}, where $i,j = 1, 2, \ldots, n$ and i is not equal to j, uses the conditional mutual information between two attributes A_i and A_j as the weight of $P(A_i|A_j, C)$, as shown in (10) (Jiang et al., 2009).

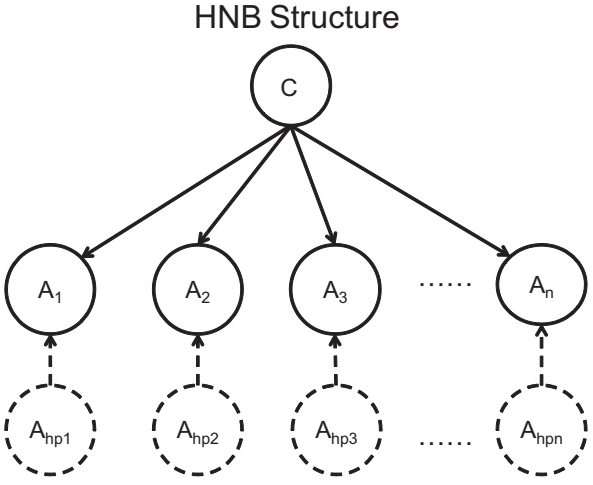

Fig. 3. HNB structure.

$$W_{ij} = \frac{I_p(A_i; A_j|C)}{\sum_{j=1, j\neq 1}^{n} I_p(A_i; A_j|C)}, \quad (10)$$

where $I_p(A_i,A_j|C)$ is the conditional mutual information defined in (11).

$$I_p(A_i; A_j|C) = \sum_{a_i, a_j, c} P(a_i, a_j, c) log \frac{P(a_i, a_j|c)}{P(a_i|c)P(a_j|c)}. \quad (11)$$

The HNB method is based on the idea of creating a hidden parent for each attribute; the influences from all of the other attributes can be easily combined through conditional mutual information by estimating the parameters from the training data. Although including the influence of complex attributes dependencies in large datasets is a promising idea, no previous studies have applied this model to the intrusion detection domain.

3. Research method

Previous scholars have applied the HNB classifier model to several datasets and have found promising results compared with the aforementioned Naïve Bayes, SBC, NBTree, TAN and AODE methods (Jiang et al., 2009). Based on the HNB classifier model's good performance from earlier results, we applied the model to the intrusion detection problem. We compared the results obtained with the HNB classifier with those obtained with the traditional Naïve Bayes classifier and other extended Naïve Bayes models, including TAN, AODE, WAODE, NBTree, DTNB and the KDD'99 winner, which were used as a common benchmark in similar studies.

Although the HNB classifier model is based on discrete features, the KDD'99 dataset mainly consists of continuous features, which need to be first converted to discrete features. In our experiments, we used two leading discretization methods: entropy minimization discretization and proportional *k*-interval discretization. We included these two methods in our study because of their good performance in the Naïve Bayes classifier method on the KDD'99 dataset (Bolon-Canedo, Sanchez-Maroo, & Alonso-Betanzos, 2009).

Finally, our research framework, as illustrated in Fig. 4, includes a feature selection model based on the three filter methods: correlation-based (CFS), consistency-based (CONS) and INTERACT feature selection methods. These approaches are leading filter-based feature selection methods that provided good results in the Naïve Bayes classifier method on the KDD'99 dataset (Bolon-Canedo et al., 2009). The Bolon-Canedo study obtained impressive Naïve Bayes classifier results by employing the combination of various discretization and filter configurations. We incorporated and used the best of these combination configurations in our study.

3.1. KDD Cup 1999 dataset (KDD-Cup., 1999)

Currently, there are only few public datasets like KDD'99 and the majority of the experiments in the intrusion detection domain performed on these datasets (Tsai, Hsu, Lin, & Lin, 2009). Since our model is based on supervised learning methods, KDD'99 is the only available dataset which provides labels for both training and test sets.

The study sample was created based on the 1998 DARPA intrusion detection evaluation offline dataset developed by the MIT Lincoln laboratory. Although there are some reported limitations (Tavallaee, Bagheri, Lu, & Ghorbani, 2009), the KDD'99 dataset has interesting properties and is believed to present a classic challenge for the intrusion detection problem. We used this dataset in our experiments because it is the most comprehensive dataset that is still widely used to compare, contrast and benchmarking the performance of intrusion detection models.

The dataset contains training data that include seven weeks of network traffic in the form of TCP dump data consisting of approximately 5 million connection records, each of which is approximately 100 bytes. The test data included two weeks of traffic, with approximately 2 million connection records. We used the labeled 10% KDD'99 dataset, which was actually used as the training dataset in the competition.

The training data contain 24 attack types, and the test data contain 38 types, all of which are mapped to four basic attack classes: probe, DoS, U2R and R2L, as shown in Table 2.

Each connection record contains 7 discrete and 34 continuous features for a total of 41 features. Each record captures various connection features, such as service type, protocol type and the number of failed login attempts. Because two of these features are constant or almost constant, they do not contribute to the classification. As shown in Table 3, the distributions of the classes are not necessarily the same in the training and test datasets. Only

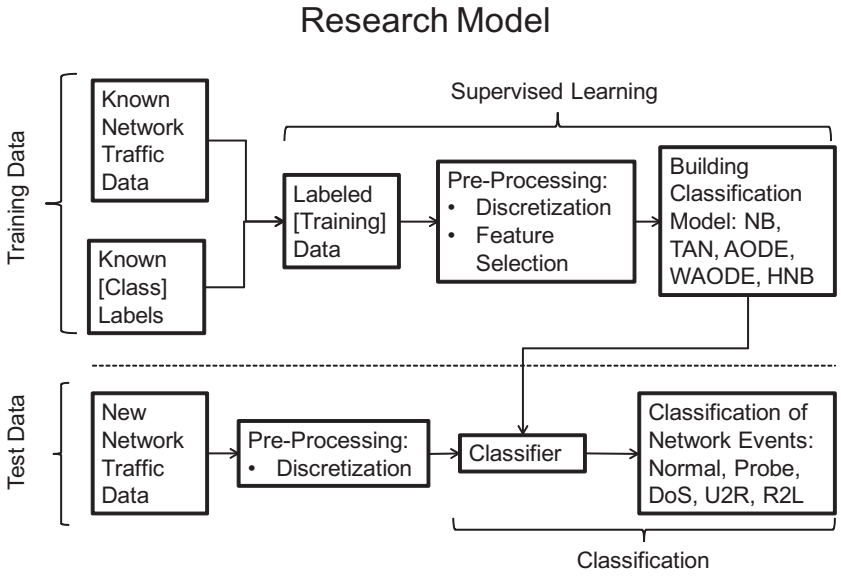

Fig. 4. Conceptual framework.

Table 2
Mapping of the attack types on the KDD'99 dataset to the attack classes on the classifier.

Class	Attacks in the training data	Additional attacks in the testing data
Probe	Ipsweep, Nmap, Portsweep, Satan	Mscan, Saint
DOS	Back, Land, Neptune, Pod, Smurf, Teardrop	Apache2, Mailbomb, Processtable, Udpstorm
U2R	Buffer_overflow, Loadmodule, Perl, Rootkit	Httptunnel, Ps, Worm, Xterm
R2L	Ftp_write, Guess_passwd, Imap, Multihop, Phf, Spy, Warezclient, Warezmaster	Named, Sendmail, Snmpgetattack, Snmpguess, Sqlattack, Xlock, Xsnoop

Table 3
Characteristics of the KDD'99 dataset.

Class	10% KDD training data distributions	10% KDD test data distributions
Normal	19.69%	19.48%
Probe	0.83%	1.34%
DoS	79.24%	73.90%
U2R	0.01%	0.07%
R2L	0.23%	5.20%

approximately 20% of the records are categorized as normal connections.

In addition to the possible interdependence between some features, the high data dimensionality of the dataset due to its large feature set poses a significant challenge to any data mining model. Feature selection and dimension reduction are common data mining approaches in large datasets. The dataset's continuous features also result in difficulties for many data mining models, including HNB and other Naïve Bayes models. Discretization is commonly used to convert continuous features into their discrete counterparts. Furthermore, discretization improves the performance of classifier models on large datasets (Liu, Hussain, Tan, & Dash, 2002), including the KDD'99 dataset (Bolon-Canedo et al., 2009).

3.2. Discretization methods

Discretization is the process of converting the continuous domain of a feature into a nominal domain with a finite number of values. Front-end discretization might be necessary for some classifiers if their algorithms cannot handle continuous features by design. Additionally, earlier studies showed that discretization improves the accuracy of classifiers, including Naïve Bayes classifiers, especially in larger datasets (Liu et al., 2002). Numerous studies have examined discretization methods in the last two decades to determine how continuous values should be grouped, how cut points should be positioned on the continuous scale and how many intervals should be used to generate datasets (Dougherty, Kohavi, & Sahami, 1995; Fayyad & Irani, 1993; Liu et al., 2002; Yang, 2002). In this study, we explain and use two leading discretization methods: entropy minimization discretization and proportional *k*-interval discretization. These two methods are selected because of their performance on large datasets, particularly the KDD'99 dataset (Bolon-Canedo et al., 2009; Bolón-Canedo, Sánchez-Maroño, & Alonso-Betanzos, 2011).

3.2.1. Entropy Minimization Discretization (EMD)

Relies on the minimum entropy heuristic required to discretize continuous features. The method selects a cut point for discretization based on the class entropy of the candidate partitions; this cut point is then recursively applied to the created intervals until the stopping condition, which is based on the minimum description length (MDL) method, is reached (Fayyad & Irani, 1993).

3.2.2. Proportional k-Interval discretization (PKID)

Tunes the interval size and interval number proportional to the number of training instances to find an appropriate trade-off between the granularity of the intervals and the expected accuracy of the probability estimation. The trade-off can also be observed as a trade-off between discretization bias and variance. Equal weights are given initially to bias and variance by creating square root of n intervals with square root of n instances in each interval, where n is the number of instances for a continuous feature. As n increases, both the number and size of the intervals increase; thus, discretization can decrease both the bias and variance of the probability estimation (Yang & Webb, 2001).

3.3. Experiments and results

Using the framework given in Fig. 4, we obtained the results for the KDD'99 multiclass classification with four attack problems by using simulation experiments with configurations consisting of a combination of 2 discretization methods, 3 feature selection methods and 7 classifiers: NB, TAN, AODE, WAODE, DTNB, NBTree, and HNB. In total, 42 model combinations were tested using the Weka tool (Witten, Frank, & Hall, 2011). A two-tailed t-test with a 95% confidence level is used to compare the models in our study.

Consistent with our framework, we applied supervised Discretize method for EMD discretization and unsupervised PKI discretize method for PKI discretization to all continuous attributes of the 10% KDD'99 dataset while using the Weka Tool with the default values. The default value is used for the equal frequency in PKI discretization, where the number of bins is equal to the square root of the number of values.

As shown in Table 4, we used the combination of features and discretization based on the Bolon-Canedo study (Bolon-Canedo et al., 2009).

After the discretization and feature selection steps, we build our classifier models for NB, TAN, AODE, WAODE, DTNB, NBTree, and HNB by using the preprocessed combinations of our dataset. We applied a 10-fold cross-validation to accurately reflect the given training data used to build the classifier models. The training data

Table 4
Features selected with the given discretization algorithms.

	EMD	PKID
CFS	dst_bytes, logged_in, count, srv_diff_host_rate, dst_host_srv_diff_host_rate	service, dst_bytes, logged_in, srv_diff_host_rate, dst_host_srv_diff_host_rate
CONS	duration, service, src_bytes, count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate	duration, service, src_bytes, dst_host_count, dst_host_same_srv_rate, dst_host_srv_serror_rate
INTERACT	service, src_bytes, dst_bytes, num_access_files, count, srv_diff_host_rate, dst_host_srv_diff_host_rate	service, src_bytes, dst_bytes, logged_in, count, srv_diff_host_rate, dst_host_count

Table 5
KDD'99 contest cost matrix.

	Normal	Probe	DoS	U2R	R2L
Normal	0	1	2	2	2
Probe	1	0	2	2	2
DoS	2	1	0	2	2
U2R	3	2	2	0	2
R2L	4	2	2	2	0

Table 6
Validation results for the overall classifier performance ranked by accuracy.

Model	Accuracy	Error rate	Cost%
WAODE_EMD_CONS	0.9996	0.0004	0.0009
NBtree_EMD_CONS	0.9994	0.0006	0.0012
AODE_EMD_CONS	0.9994	0.0006	0.0013
HNB_EMD_CONS	0.9993	0.0007	0.0013
TAN_EMD_CONS	0.9993	0.0007	0.0014
NBtree_EMD_INT	0.9988	0.0012	0.0022
WAODE_EMD_INT	0.9988	0.0012	0.0022
TAN_EMD_INT	0.9988	0.0012	0.0022
HNB_EMD_INT	0.9987	0.0013	0.0023
AODE_EMD_INT	0.9986	0.0014	0.0027
WAODE_PKI_INT	0.9986	0.0014	0.0024
TAN_PKI_INT	0.9986	0.0014	0.0028
HNB_PKI_INT	0.9983	0.0017	0.0030

Table 7
Test results for the overall classifier performance ranked by accuracy.

Model	Accuracy	Error rate	Cost%
HNB_PKI_INT	0.9372	0.0628	0.2224
KDD'99 Winner	0.9271	0.0729	0.2331
AODE_EMD_INT	0.9269	0.0731	0.2336
NB_EMD_INT	0.9262	0.0738	0.2254
TAN_PKI_CONS	0.9259	0.0741	0.2393
AODE_PKI_CONS	0.9259	0.0741	0.2429
HNB_EMD_CONS	0.9254	0.0746	0.2393
WAODE_EMD_INT	0.9242	0.0758	0.2389
TAN_EMD_INT	0.9241	0.0759	0.2390
DTNB_PKI_CONS	0.9239	0.0761	0.2410
NB_PKI_INT	0.9230	0.0770	0.2443
WAODE_EMD_CONS	0.9230	0.0770	0.2454
WAODE_PKI_INT	0.9227	0.0773	0.2477

are randomly divided into 10 subsets of equal size. In each iteration, one of the subsets is used for testing, and the remaining sets are used to train the classifier. In 10 iterations, each subset is used for testing at least once. The cross-validation estimate of the accuracy is the mean of the estimates collected from each iteration. Ten-fold cross-validation is extensively used and is considered accurate in similar studies with large datasets. The validation results based solely on the training data are provided in Table 6. We then applied the classifier models built from the training data to the test data. The results based on the test data are provided in Table 7.

We used accuracy and the error rate as performance measures in our multiclass classifier study. Accuracy is the fraction of correctly classified instances, and the error rate is the fraction of misclassified instances in a dataset. These two measures effectively summarize the overall performance by considering all of the classes and generalizing the classifier performance in terms of the convergence behaviors (Japkowicz & Shah, 2011). With the interest of the space, only the results of the top-performing models in our experiments are provided on our article.

We also used cost-based evaluation, which was the primary goal and evaluation method in the KDD'99 contest (Elkan, 2000). Cost indicates the average cost of the misclassified connection and is calculated with the Eq. (12) where M_{ij} is the number of samples in class i that are misclassified as class j, C_{ij} is the corresponding cost in the cost matrix provided in Table 5 and N is the total number of instances. The cost comparisons of the evaluated models are provided in Table 7.

$$\text{Cost} = \frac{1}{N}\Sigma M_{ij} \times C_{ij} \tag{12}$$

The winner of the KDD'99 contest achieved an accuracy rate of 0.9271, an error rate of 0.0729, and an average cost of 0.2331 on the test dataset (Elkan, 2000; Sabhnani & Serpen, 2003). According to the confusion matrix of the winner entry given in the Elkan study (Elkan, 2000), the winner predicted the DoS class with an accuracy rate of 0.9712 and an error rate of 0.0288.

Our validation results indicate that the models based on entropy minimization discretization and consistency-based feature selection methods perform better, while the results of the HNB model show that HNB is one of the leading models in terms of performance, as shown in Table 6. The best validation result is obtained with the WAODE model. Because the class distributions on the KDD'99 dataset differ in the training and test data and the validation results only rely on the training data, these findings will not be sufficient for measuring the performance of the classification models.

The test results given in Table 7 provide a better indication of the class prediction performance because the test data are independent from the training data. Based on these test results, the best model is an HNB model (specifically, the HNB model with proportional k-Interval discretization and INTERACT feature selection methods in all three of the performance categories). The result for this model is also better than that for the KDD'99 winner, which is commonly accepted as a benchmark in similar studies.

These results are consistent with the earlier study on the HNB, where the HNB exhibited a remarkable performance in comparison with traditional and other extended Naïve Bayes methods using other datasets (Jiang et al., 2009).

According to the test results shown in Table 8, the model that provides a better overall performance is also better at detecting denial-of-services (DoS) attacks than the traditional and extended Naïve Bayes methods and the KDD'99 winner.

We also compared the best results obtained in our experiments with the comparable results from earlier studies on intrusion detection, as shown in Table 9. We used the results obtained using

Table 8
Test results for the denial of service (DoS) Classifier performance ranked by accuracy.

Model	Accuracy	Error rate
HNB_PKI_INT	0.9960	0.0040
TAN_PKI_CONS	0.9757	0.0243
DTNB_PKI_CONS	0.9751	0.0249
AODE_PKI_CONS	0.9750	0.0250
HNB_EMD_CONS	0.9744	0.0256
DTNB_EMD_CFS	0.9735	0.0265
TAN_EMD_INT	0.9733	0.0267
AODE_EMD_INT	0.9733	0.0267
WAODE_EMD_INT	0.9732	0.0268
HNB_EMD_CFS	0.9727	0.0273
WAODE_EMD_CFS	0.9725	0.0275
AODE_EMD_CFS	0.9725	0.0275
TAN_PKI_INT	0.9724	0.0276
NBtree_EMD_CFS	0.9724	0.0276
HNB_PKI_CFS	0.9723	0.0277
WAODE_PKI_INT	0.9722	0.0278
TAN_EMD_CFS	0.9721	0.0279
AODE_PKI_INT	0.9717	0.0283
DTNB_PKI_INT	0.9716	0.0284
NB_EMD_INT	0.9713	0.0287
KDD'99 Winner	0.9712	0.0288

Table 9
Comparison of the overall classifier performance sorted by accuracy.

Model	Accuracy	Error rate
HNB_PKI_INT	0.9372	0.0628
JRip (Nguyen & Choi, 2008)	0.9230	0.0770
NBTree (Nguyen & Choi, 2008)	0.9228	0.0772
LBk (Nguyen & Choi, 2008)	0.9222	0.0778
SVM (Ambwani 2003)	0.9218	0.0803
J48 (Nguyen & Choi, 2008)	0.9206	0.0794
MLP (Nguyen & Choi, 2008)	0.9203	0.0797
Decision Table (Nguyen & Choi, 2008)	0.9166	0.0834
SMO (Nguyen & Choi, 2008)	0.9165	0.0835
BayesNet (Nguyen & Choi, 2008)	0.9062	0.0938
OneR (Nguyen & Choi, 2008)	0.8931	0.1069
Naïve Bayes (Nguyen & Choi, 2008)	0.7832	0.2168

the state-of-the-art support vector machine (SVM) from the (Ambwani, 2003) and the results obtained using various leading methods, such as J48 (C4.5 decision tree version 8), JRip (Ripper), multilayer perceptron (MLP), sequential minimal optimization (SMO) and the lazy classifier (LBk) from the (Nguyen & Choi, 2008) study. Both of these studies used the KDD'99 dataset labeled with four attack categories. Based on the results of these two studies, our best model, the HNB model with proportional *k*-Interval discretization and INTERACT feature selection methods, has better predictive accuracy.

Consistent with the findings of the earlier Bolon-Canedo study (Bolon-Canedo et al., 2009), our results based on the experiments using all of the instances of the 10% KDD'99 dataset and the feature set consists of as few as 7 out of 41 features, as shown in Table 4. Our model provides a competitive advantage in that it provides better predictive performance while significantly reducing the feature set used in the intrusion detection dataset.

4. Conclusion

In this paper, we explained the need to apply data mining methods to network events to classify network attack events. We summarized the results of earlier studies and explored the earlier models on the performance improvement of the Naïve Bayes model in data mining and introduced the HNB model as a solution to the intrusion detection problem. We augmented the Naïve Bayes and structurally extended Naïve Bayes methods with the leading discretization and feature selection methods to increase the accuracy and decrease the resource requirements of intrusion detection problem. We compared the performance of the Naïve Bayes and leading extended Naïve Bayes approaches with the new HNB approach as an intrusion detection system. The results of our experimental study, which uses the KDD'99 dataset, show that the Hidden Naïve Bayes multiclass classification model augmented with various discretization and feature selection methods exhibits better overall results in terms of detection accuracy, error rate and misclassification cost than the traditional Naïve Bayes model, the leading extended Naïve Bayes models and the KDD'99 winner. The results also indicate that our model significantly improves the detection of denial-of-service attacks compared with the other models.

Considering its simplicity and its advantage over the Naïve Bayes model's conditional independence assumption, hidden Naïve Bayes is a promising model for datasets with dependent attributes, such as the KDD'99 intrusion detection dataset.

References

Ambwani, T. (2003). Multi class support vector machine implementation to intrusion detection. In *Paper presented at the international joint conference on neural networks*, 20–24 July.

Baig, Z. A., Shaheen, A. S., & AbdelAal, R. (2011). An AODE-based intrusion detection system for computer networks. In *Paper presented at the world congress on internet security (WorldCIS)* (Feb., pp. 21–23).

Baker, S., Filipiak, N., & Timlin, K. (2011). In the dark: crucial industries confront cyberattacks McAfee annual critical infrastructure protection report (2nd ed.). Santa Clara, CA: The Center for Strategic and International Studies (CSIS).

Barbara, D., Wu, N., & Jajodia, S. (2001). Detecting novel network intrusions using Bayes estimators. In *Paper presented at the first SIAM conference on data mining*, Chicago, IL.

Benferhat, S., Boudjelida, A., & Drias, H. (2008). On the use of TAN in intrusion detection systems. In *Proceedings of the second international conference on information processing* (pp. 1–13).

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence, 97*(1–2), 245–271. http://dx.doi.org/10.1016/s0004-3702(97)00063-5.

Bo, Hui-Ye, & Yu-Hang. (2002). HMMs (Hidden Markov Models) based on anomaly intrusion detection method. In *Paper presented at the international conference on machine learning and cybernetics*.

Bo, Qiurui, Zhong, & Zengmei. (2009). A study on automatic web pages categorization. In *Paper presented at the IEEE international advance computing conference IACC 2009*.

Bolon-Canedo, V., Sanchez-Maroo, N., & Alonso-Betanzos, A. (2009). A combination of discretization and filter methods for improving classification performance in KDD Cup 99 dataset. In *Paper presented at the international joint conference on neural networks, IJCNN 2009* (pp. 14–19).

Bolón-Canedo, V., Sánchez-Maroño, N., & Alonso-Betanzos, A. (2011). Feature selection and classification in multiple class datasets: An application to KDD Cup 99 dataset. *Expert Systems with Applications, 38*(5), 5947–5957. http://dx.doi.org/10.1016/j.eswa.2010.11.028.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140. http://dx.doi.org/10.1007/bf00058655.

Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5–32. http://dx.doi.org/10.1023/a:1010933404324.

Bridges, S. M., & Vaughn, R. B. (2000). Fuzzy data mining and genetic algorithms applied to intrusion detection. In *Proceedings of the 23rd national information systems security conference (NISSC)*.

Cannady, J. (1998). The application of artificial neural networks to misuse detection: Initial results. In *Proceedings of the recent advances in intrusion detection '98 conference* (pp. 31–47).

Chickering, D. M., Heckerman, D., & Meek, C. (2004). Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research, 5*, 1287–1330.

Dash, M., & Liu, H. (2003). Consistency-based search in feature selection. *Artificial Intelligence, 151*(1–2), 155–176. http://dx.doi.org/10.1016/s0004-3702(03)00079-1.

Deng, W., Wang, G., & Wang, Y. (2007). Weighted Naïve Bayes classification algorithm based on rough set. *Computer Science, 34*(2), 204–206.

Denning, D. (1987). An intrusion-detection model. *IEEE Transactions on Software Engineering, 13*(2), 222–232.

Depren, O., Topallar, M., Anarim, E., & Ciliz, M. K. (2005). An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. *Expert Systems with Applications, 29*(4), 713–722. http://dx.doi.org/10.1016/j.eswa.2005.05.002.

Dokas, P., Ertoz, L., Kumar, V., Lazarevic, A., Srivastava, J., & Tan, P. (2002). Data mining for network intrusion detection. In *Paper presented at the proceedings of the nsf workshop on next generation data mining*, Baltimore.

Dougherty, J., Kohavi, R., & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. In *Proceedings of 12th international conference machine learning* (pp. 194–202).

Elkan, C. (2000). Results of the KDD'99 classifier learning. *ACM SIGKDD Explorations Newsletter, 1*(2), 63–64. http://dx.doi.org/10.1145/846183.846199.

Fayyad, U., & Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th international joint conference on artificial intelligence* (pp. 1022–1029).

Fossi, M., Egan, G., Haley, K., Johnson, E., Mack, T., Adams, T., et al. (2011). *Symantec internet security threat report: Trends for 2010* (Vol. 16). Mountain View, CA.

Frank, J. (1994). Artificial intelligence and intrusion detection: Current and future directions. In *Proceedings of the 17th national computer security conference*.

Frank, E., Hall, B., & Pfahringer, B. (2003). Locally weighted Naïve Bayes. In *Proceedings of conference on uncertainty in artificial intelligence* (pp. 249–256.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning, 29*(2), 131–163. http://dx.doi.org/10.1023/a:1007465528199.

Hall. (1999). Correlation-based feature selection for machine learning. (Ph.D), The University of Waikato, Hamilton, New Zealand.

Hall, E., & Frank, M. (2008). Combining Naïve Bayes and decision tables. In *Paper presented at the proceedings of 21st Florida artificial intelligence research society conference*, Miami, Florida.

Hand, D. J., Mannila, H., & Smyth, P. (2001). *Principles of data mining*. Cambridge, Mass.: MIT Press.

Hofmann, A., & Sick, B. (2003). Evolutionary optimization of radial basis function networks for intrusion detection. In *Paper presented at the international joint conference on neural networks*.

Huan, L., & Lei, Y. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering, 17*(4), 491–502.

Huan, L., & Setiono, R. (1997). Feature selection via discretization. *IEEE Transactions on Knowledge and Data Engineering, 9*(4), 642–645.

Japkowicz, N., & Shah, M. (2011). *Evaluating learning algorithms: A classification perspective*. Cambridge; New York: Cambridge University Press.

Jiang, L., & Zhang, H. (2006). Weightily averaged one-dependence estimators. In *Paper presented at the proceedings of the 9th Pacific Rim international conference on artificial intelligence*, Guilin, China.

Jiang, Zhang, Cai, & Su. (2005). Evolutional Naïve Bayes. In *Paper presented at the proceedings first international symposium intelligent computation and its applications (ISICA '05)*.

Jiang, L., Wang, D., Zhang, H., Cai, Z., & Huang, B. (2008). Using instance cloning to improve Naïve Bayes for ranking. *International Journal of Pattern Recognition and Artificial Intelligence, 22*(6), 1121–1140.

Jiang, L., Zhang, H., & Cai, Z. (2009). A novel Bayes model: Hidden Naïve Bayes. *IEEE Transactions on Knowledge and Data Engineering, 21*(10), 1361–1371.

Kabiri, P., & Ghorbani, A. A. (2005). Research on intrusion detection and response: A survey. *International Journal of Network Security, 1*(2), 84–102.

KDD-Cup. (1999). KDD Cup 1999 Data Retrieved July 29, 2011, from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Kohavi, R. (1996). Scaling up the accuracy of Naïve –Bayes classifiers: A decision-tree hybrid. In *Proceedings of second international conference knowledge discovery and data mining (KDD'96)* (pp. 202–207).

Kumar, S., Spafford, E.H. (1994). A pattern matching model for misuse intrusion detection. In *Proceedings of the 17th national computer security conference* (pp. 11–21).

Langley, P., & Sage, S. (1994). Induction of selective Bayesian classifiers. In *Paper presented at the proceedings tenth conference on uncertainty in artificial intelligence*.

Langley, P., Iba, W., & Thompson, K. (1992). An analysis of Bayesian classifiers. In *Proceedings of the tenth national conference on artificial intelligence* (pp. 223–228).

Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., & Srivastava, J. (2003). A comparative study of anomaly detection schemes in network intrusion detection. In *Paper presented at the proceedings of the third SIAM international conference on data mining*, San Francisco, CA.

Lee, W., Stolfo, S.J., & Mok, K.W. (1999). A data mining framework for building intrusion detection models. In *Proceedings of the 1999 IEEE symposium on security and privacy* (pp. 120–132).

Lippmann, R. P., & Cunningham, R. K. (2000). Improving intrusion detection performance using keyword selection and neural networks. *Computer Networks, 34*(4), 597–603. http://dx.doi.org/10.1016/s1389-1286(00)00140-7.

Liu, H., Hussain, F., Tan, C. L., & Dash, M. (2002). Discretization: An enabling technique. *Data Mining and Knowledge Discovery, 6*(4), 393–423. http://dx.doi.org/10.1023/a:1016304305535.

Lunt, T. F. (1989). Real-time intrusion detection. In *Paper presented at the COMPCON Spring '89.Tthirty-fourth IEEE computer society international conference*. Intellectual leverage, digest of papers.

Nguyen, H. A., & Choi, D. (2008). Application of data mining to network intrusion detection: classifier selection model. In *Paper presented at the proceedings of the 11th Asia-Pacific symposium on network operations and management: challenges for next generation network operations and service management*, Beijing, China.

Panda, M., & Patra, M. (2009). Semi-Naïve Bayesian method for network intrusion detection system. In C. Leung, M. Lee, & J. Chan (Eds.). *Neural information processing* (Vol. 5863, pp. 614–621). Berlin, Heidelberg: Springer.

Sabhnani, M., & Serpen, G. (2003). Application of machine learning algorithms to kdd intrusion detection dataset within misuse detection context. In *Paper presented at the proceedings of the international conference on machine learning, models, technologies and applications (MLMTA 2003)*, Las Vegas.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD cup 99 data set. In *Paper presented at the IEEE symposium on computational intelligence in security and defense applications (CISDA2009)*.

Tsai, C., Hsu, Y., Lin, C., & Lin, W. (2009). Intrusion detection by machine learning: A review. *Expert Systems with Applications, 36*(10), 11994–12000. http://dx.doi.org/10.1016/j.eswa.2009.05.029.

Webb, G. I., Boughton, J. R., & Wang, Z. (2005). Not so Naïve Bayes: Aggregating one-dependence estimators. *Machine Learning, 58*(1), 5–24. http://dx.doi.org/10.1007/s10994-005-4258-6.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data mining: Practical machine learning tools and techniques* (3rd ed.). Burlington, MA: Morgan Kaufmann.

Wu, S., & Yen, E. (2009). Data mining-based intrusion detectors. *Expert Systems with Applications, 36*(3, Part 1), 5605–5612. http://dx.doi.org/10.1016/j.eswa.2008.06.138.

Xie, Z., Hsu, W., Zongtian, L., & Lee, M. (2002). SNNB: A selective neighborhood based Naïve Bayes for lazy learning. In *Paper presented at the proceedings of the 6th Pacific-Asia conference on advances in knowledge discovery and data mining*.

Xin Jin, Rongyan Li, Xian Shen, & Rongfang. Bie. (2007). Automatic web pages categorization with Relief and Hidden Naïve Bayes. In *Paper presented at the proceedings of the 2007 ACM symposium on applied computing*, Seoul, Korea.

Yaguang, J., Songnian, Y., & Yafeng, Z. (2011). A novel Naïve Bayes model: Packaged Hidden Naïve Bayes. In *Paper presented at the information technology and artificial intelligence conference (ITAIC), 6th IEEE joint international* 20–22 Aug

Yang, Y., & Webb, G.I. (2002). A comparative study of discretization methods for Naïve –Bayes classifiers. In *Proceedings of PKAW 2002: The 2002 Pacific Rim knowledge acquisition workshop* (pp. 159–173).

Yang, Ying, & Webb, Geoffrey I. (2001). Proportional *k*-Interval discretization for Naïve –Bayes classifiers. *Machine Learning: ECML, 2001*(2167), 564–575.

Zhang, Z. (2001). HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification. In *Proceedings of the 2001 IEEE workshop on information assurance and security* (pp. 85–90).

Zhang, J., Zulkernine, M., & Haque, A. (2008). Random-forests-based network intrusion detection systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 38*(5), 649–659.

Zhao, Z., & Liu, H. (2007). Searching for interacting features. In *Paper presented at the proceedings of the 20th international joint conference on artifical intelligence*, Hyderabad, India.