International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:11, 2008

Network Intrusion Detection Design Using Feature Selection of Soft Computing Paradigms

T. S. Chou, K. K. Yen, and J. Luo

***Abstract*—**The network traffic data provided for the design of intrusion detection always are large with ineffective information and enclose limited and ambiguous information about users' activities. We study the problems and propose a two phases approach in our intrusion detection design. In the first phase, we develop a correlation-based feature selection algorithm to remove the worthless information from the original high dimensional database. Next, we design an intrusion detection method to solve the problems of uncertainty caused by limited and ambiguous information. In the experiments, we choose six *UCI* databases and *DARPA KDD99* intrusion detection data set as our evaluation tools. Empirical studies indicate that our feature selection algorithm is capable of reducing the size of data set. Our intrusion detection method achieves a better performance than those of participating intrusion detectors.

***Keywords*—**Intrusion detection, feature selection, k-nearest neighbors, fuzzy clustering, Dempster-Shafer theory

I. INTRODUCTION

INTRUSION detection systems are security management systems that are used to discover inappropriate, incorrect, or anomalous activities within computers or networks. With the rapid growth of Internet, these malicious behavior are increasing at a fast pace and can easily cause millions of dollar in damage to an organization. Hence, the development of intrusion detection systems has been set with the highest priority by government, research institutes and commercial corporations.

During the past years, existing intrusion detection systems take a variety of approaches to the task of detecting intruders' activities. For developing the systems, data are collected and then provided for the use of overall design process. However, these data sources do have some problems such as problem of irrelevant and redundant features, problem of uncertainty, and problem of ambiguity. These problems not only hinder the detection speed but also decline the detection performance of intrusion detection systems. Therefore, we propose a two phases approach in our intrusion detection design in order to successfully solve these problems mentioned above. In the first phase, we develop a feature selection algorithm based on information-theoretical measures to reduce the complexity of the high dimensional network database. The algorithm uses *symmetric uncertainty* [1] to evaluate the worth of features and then eliminate both irrelevant features with poor prediction ability to the class and redundant features that are inter-correlated with one or more of the other features. After removing irrelevant and redundant features, the remaining ones contain indispensable information about the original feature space. Having reduced the complexity of the original data set, the compact data set is fed into the second phase for the task of identifying intrusions. In this phase, we propose incorporating fuzzy clustering technique [2], [3] and Dempster-Shafer theory [4], [5] into our intrusion detection design for their merits of resolving uncertainty problems caused by ambiguous and limited information. The k-nearest neighbors (k-NN) technique [6] is applied to speed up the detection process. During the entire of work, *DARPA KDD99* intrusion detection evaluation data set [7] is employed. For evaluating the performance of proposed feature selection algorithm, six *UCI* repository of machine learning databases [8], two *symmetric uncertainty* based feature selection algorithms, Correlation Based Feature Selection (CFS) [9] and Fast Correlation-Based Filter (FCBF) [10], and two machine learning algorithms, naive bayes [11] and C4.5 [12], are used. For evaluating the detection performance of proposed intrusion detection method, three k-NN based pattern classification algorithms, k-NN [6], fuzzy k-NN [13], and evidence-theoretic k-NN [14] classifiers, are chosen to compare with.

This paper is organized as follows. Section 2 describes the problems in the collected network traffic. Section 3 presents the theoretical framework in our feature selection and intrusion detection approaches. We then demonstrate the experimental methodology, followed by a discussion of the experimental results. Finally, we conclude our work and discuss the future work in the last section.

II. PROBLEM STATEMENTS

Basically, there are two approaches for intrusion detection design based on the uses of detection techniques: knowledge-based and behavior-based intrusion detection. Knowledge-based intrusion detection is also called misuse detection. In principle, it is typically realized by modeling known attack behavior with prior understanding about specific attacks and system vulnerabilities. Afterward, the intrusion detection system compares network traffic data being observed with well defined attack patterns for identifying the possible penetrations to the system. When the data is as same as one of the explicitly defined attack patterns, an alarm is raised. The

Manuscript received October 31, 2007.

T. S. Chou is with the Department of Electrical and Computer Engineering, Florida International University, Miami, FL 33174 USA, (phone: 305-227-1027; e-mail: tchou001@fiu.edu).

K. K. Yen and J. Luo are with the Department of Electrical and Computer Engineering, Florida International University, Miami, FL 33174 USA, (e-mail: {yenk,jluo001}@fiu.edu).

defined attack patterns are frequently referred to as the signatures of intrusions. The signature could be a static string or a sequence of events.

While knowledge-based intrusion detection is achieved by modeling known attack behavior, on the contrary, behavior-based intrusion detection also known as anomaly detection models normal or expected behavior of computer users. It looks for malicious activities by comparing the observed data with these acceptable behaviors. If the data diverge from the learned normal behavior, an alarm is raised. In other word, anything will be suspected as an attack if its behavior is deviated from the previously learned behaviors.

For developing intrusion detection systems, a large amount of traffic data is always necessary to be collected in advance for analysis with the use misuse detection or anomaly detection approaches. Based on the collected network audit trail, misuse detection techniques specify well defined attack signatures and anomaly detection techniques establish acceptable usage profiles to differentiate intrusions and normal activities from a future network traffic data stream. However, there are three major problems in the collected network traffic database: problem of irrelevant and redundant features, problem of uncertainty, and problem of ambiguity. The details are described as follows.

A. Problem of Irrelevant and Redundant Features

For designing an intrusion detection system, a data set is prepared for analysis. In general, the quantity of data is enormous that includes thousands of traffic records with a number of various features such as the length of the connection, the type of protocol, the network service and other information. Theoretically and ideally, the ability to discriminate attack from normal behavior should be performed better if more features are added during the analysis process. However, the answer is sometimes negative because not every feature of traffic data is relevant to the intrusion detection task. Among the large amount of features, some of the features may be irrelevant with poor prediction ability to the target patterns, and some of the features may be redundant due to they are highly inter-correlated with one of more of the other features [15]. If irrelevant and redundant features are involved in the analysis, not only the detection speed becomes slow but also the detection accuracy possibly decreases. For achieving a better overall detection performance, any irrelevant and redundant features should be discarded from the original feature space. How to select a meaningful subset of features from the network traffic data stream is therefore becomes a very important and indispensable task in the beginning of an intrusion detection process.

B. Problem of Uncertainty

Uncertainties exist in our daily life. Sometimes the uncertainty is totally random, e.g., the future state of the weather and the occurrence of failure of your home appliances. Sometimes it happens due to lack of knowledge or unpredictable factors such as the trend of stock and whether a war is going to happen. Therefore, people generally classify uncertainties into two categories, aleatory uncertainty and epistemic uncertainty, based on their fundamentally different in nature. Aleatory uncertainty is also known as variability, random uncertainty, stochastic uncertainty, objective uncertainty, and irreducible uncertainty [16], [17]. It is caused by inherent random variations associated with the physical system or the environment under consideration. Examples can be found in the outcomes while rolling a dice, the location and time of occurrence of future earthquakes, and variability in a machining operation. The random nature of aleatory uncertainty is inherent. The occurrence of an event is not predicable even a large quantity of past data is collected.

The second type of uncertainty is epistemic uncertainty. This uncertainty is also referred to as imprecision, reducible uncertainty, subjective uncertainty, parameter uncertainty, model form uncertainty, and state-of-knowledge uncertainty [16], [17]. On the contrary to aleatory uncertainty that uncertainty arises from the system itself, epistemic uncertainty is an uncertainty that is due to a lack of knowledge or information of processes of the system or the environment. Since it is not caused by the inherent random variations of the system but by the incomplete information or knowledge, the uncertainty is possible to be reduced by including new knowledge or information about the system or environmental factors. Examples of epistemic uncertainty can be seen when there are insufficient experimental data to describe physical parameters of a new material, limited understanding of a physics phenomena, and imperfect measurement of a complex physical model.

Actually, epistemic uncertainty does happen in intrusion detection tasks. From the decision-based perspective, the goal of intrusion detection is to make decisions whether future traffic data are malicious or normal. For effectively and precisely making the decisions, data are collected in advance for analysis either misuse or anomaly detection technique is used. However, the collected data always enclose uncertainty when only limited information about intrusive activities is available. In real world modern computer systems and networks, hackers constantly develop new attack codes to exploit security vulnerabilities of organizations everyday. Not only are these attacks becoming more numerous, they are also becoming more sophisticated. Accordingly, it is not realistic to cover all intrusive behavior space completely for the use of decision making in an intrusion detection system.

C. Problem of Ambiguity

The network traffic activities often contain ambiguous information about computer users' activities. The patterns generated from users' behavior always cannot be specifically defined as normality and abnormality. If the behavior is not considered anomalous, then intrusion activity may not be detected. If the behavior is considered anomalous, then system administrators may be alerted by false alarms, i.e., in cases where there is no intrusion [18]. The boundary between normal activities and abnormal ones are always unclear. In order to illustrate the type of ambiguity mentioned above, let's consider the following example of a person who tries to access

an account from a remote machine. A user attempts to retrieve forgotten passwords when he/she logins his/her own account, and this action is considered as a normal behavior. On the other hand, the action that a hacker attempts to access other people's accounts by guessing passwords is definitely an intrusive activity. Thus, ambiguity is involved during the process of classifying intrusions from normal activities. If the guessing passwords behavior of a hacker is considered as a normal activity, then the intrusion can never be detected. If the retrieving forgotten passwords behavior of a user is considered as an intrusive activity, then system administrators may fire an alarm but actually there is no intrusion happened. Hence, ambiguity is necessary to be concerned in the incomplete and imprecise available data set during the intrusion detection procedure.

III. THEORETICAL FRAMEWORK

A. Feature Selection Algorithm

The feature selection techniques are generally mainly divided into two categories, filter and wrapper, as defined in the work of John et al. [19]. Filter method operates without engaging any information of induction algorithm. By using some prior knowledge such as feature should have strong correlation with the target class or feature should be uncorrelated to each other, filter method selects the best subset of features. The most well-known filter methods are Relief [20] and Focus [21]. By employing the filter approach to intrusion detection work, Qu et al. [22] applied pairwise correlation analysis to uncover mutual information between each feature and the decision class. Irrelevant and redundant features were then removed from the *DARPA KDD99* benchmark data set. Example can also be found in the work of Kayacık et al. [23]. They performed feature relevance analysis on the *KDD99* training set. In order to get feature relevance measure for all attacks, they used information gain performing on binary classification and reported their chosen relevant features for normal connections and some of attacks.

Alternatively, wrapper method employs a predetermined induction algorithm to find a subset of features with the highest evaluation by searching through the space of feature subsets and evaluating quality of selected features. The process of feature selection acts like "wrapped around" an induction algorithm. Machine learning algorithms such as ID3 [24] and C4.5 [12] are commonly used as the induction algorithm. For increasing the detection rate and decreasing the false alarm rate in a network intrusion detection task, Stein et al. [25] used genetic algorithm to select a subset of features with C4.5 algorithm. By applying cross-validation to test the classification error rate, the fitness of individual feature was obtained and thus that feature can be decided to be added or removed from the feature subset used. The work of Mukkamala and Sung [26] is another example of using wrapper method. With the use of *KDD99* data set, they applied both Support Vector Machines (SVM) and Support Vector Decision Function Ranking Method (SVDFRM) to rank important input features for intrusion detection. They deleted one feature at a time and the remaining features were used for training and testing the classifier. They then compared the classifier's performance with that of the classifier with original feature set. Finally, the importance of the feature was ranked according to a set of rules based on the performance comparison. Based on the iterative search and evaluation procedure, the forty-one features were grouped into important features, secondary features, and unimportant features for normal, *Denial of Service* (*DoS*), *Probe*, *User to Root* (*U2R*), and *Remote to Local* (*R2L*) attacks.

Since wrapper approach includes a specific induction algorithm to optimize feature selection, it often provides a better classification accuracy result than that of filter approach. However, wrapper method is more time consuming than filter method due to it is strongly coupled with an induction algorithm with repeatedly calling the algorithm to evaluate the performance of each subset of features. It thus becomes unpractical to apply a wrapper method to select features from a large data set that contains numerous features and instances [27]. Furthermore, wrapper approach is required to re-execute its induction algorithm for selecting features from data set while the algorithm is replaced with a dissimilar one. It is less independent of any induction algorithms than filter is.

Consequently, we address aspects of feature selection based on filter method since the size of data collected from the network is always large which includes many traffic records with a number of various features. Our approach uses the concept of information theory to evaluate the worth of features and then eliminate both irrelevant and redundant features. The approach is closer to FCBF, however we treat the correlation between features in a global perspective. We measure the total amount of information enclosing in a feature as the summation of inter-correlations to all of the rest of the features, but FCBF only considers on a feature of rest ones at a time. Therefore, FCBF may be tricked in situation where the dependence between pair of features is weak but the total inter-correlated strength of one feature to the others is strong. The result is that FCBF possibly keeps a feature that its information can be found in the remaining selected subset of features. In addition, FCBF requires adjusting a threshold for its feature selection procedure, while our algorithm does not.

Based on filter method, we use information theory to evaluate the strengths of features and select a subset of features from the original ones. Figure 1 shows our proposed feature selection algorithm using information-theoretical measures. Within the algorithm, we choose *symmetric uncertainty* to find the strength of predictive from features to target classes and the strength of correlation between features themselves.

The algorithm consists of two parts to select the most informative features to target classes from the original feature space. In the first part (lines 1-5), the algorithm removes irrelevant features with poor prediction ability to target class. The second part of the algorithm (lines 6-12) eliminates redundant features that are inter-correlated with one or more of other features. Finally, the remaining selected features are

```
1   // Remove irrelevant features
2   Input original data set D that includes features X
    and target class Y
3   For each feature X_i
        Calculate mutual information SU(Y; X_i)
4   Sort SU(Y; X_i) in descending order
5   Put X_j whose SU(Y; X_i) > 0 into relevant feature
    set R_XY
6   // Remove redundant features
7   Input relevant feature set R_XY
8   For each feature X_j
        Calculate pairwise mutual information
        SU(X_j; X_k) ∀j ≠ k
9   S_XX = Σ(SU(X_j; X_k))
10  Calculate means μ_R and μ_S of R_XY and S_XX ,
    respectively. w = μ_S / μ_R
11  R = w·R_XY - S_XX
12  Select X_j whose R > 0 into final set F
```

Fig. 1 Feature selection algorithm

all significant features that contain indispensable information about the original feature set.

Given a data set with a number of input features and a target class, the algorithm first calculates the mutual information between features and class. The algorithm then ranks the features in descending order according to their degrees of association to the target class. Once the importance of the input features are ranked, those terms whose information measure are greater than zero are kept; which means those removed features are totally irrelevant to target class and the remaining ones are predictive.

In the second part, it starts with calculating the inter-correlated strengths of each pair of features. The total amount of mutual information for each feature is acquired by adding all mutual information measures together that relate to that feature. For adjusting the discriminative power of mutual information performed on feature-to-feature and feature-to-class to the same level, we introduce factor w and its value is equal to the mean of summation of feature-to-class information divided by the mean of summation of feature-to-feature information. By multiplying w to each feature-to-class measure, both feature-to-class and feature-to-feature reach to the same important rank. Finally, the differences of them are computed and we only keep those features whose values are greater than zero; which means the selected features are the most "significant features" that restrain indispensable information of the original feature space.

B. Fuzzy Belief k-NN Intrusion Detection Algorithm

The problem of detecting intrusions in fact can be treated as a classification task, i.e., to classify network traffic into normal usage category or attack category. In our work, the main goal is to identify attacks from the *KDD99* intrusion detection benchmark data set. For successfully achieving the goal, we divide the intrusion detection task into two phases: training phase and classification phase. In the training phase, decision rules are generated in accordance with the clustering result of provided training data. Having finished the first phase, the rules are used for classifying whether the future network traffic is a normal activity or an attack in classification phase. Figure 2 depicts the general operation scheme of the proposed approach. The details are described as follows.

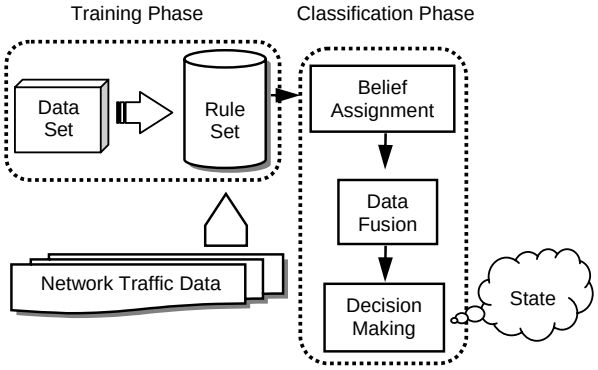

Fig. 2 Intrusion detection identification

1) The Training Phase

Let's assume the available information in the *KDD99* training set that contains N network traffic connections, and each of them is composed of n distinct features with positive numeric values. We denote the training set as T, the training connection as x, and the set of features in each connection as F.

$$T = \{x_1, x_2, ..., x_N\} \quad (1)$$

$$F = \{f_1, f_2, ..., f_n\} \quad (2)$$

As described in the previous section, a training connection sometimes could not be crisply defined as normality or abnormality, i.e., could be belonged to more than one category. Among all possible approaches in unsupervised learning techniques, clustering algorithms have been shown to be an effective way to group similar objects together from a given set of inputs. Hence, in the beginning of the intrusion detection task we apply fuzzy c-Means clustering technique to deal with the above uncertainty by assigning diverse degrees of membership to classes that a training connection may belong to. We denote the class set L and it includes a number of p possible classes.

$$L = \{l_1, l_2, ..., l_p\} \quad (3)$$

The clustering procedure is done by using iterative optimization technique to minimize objective function J.

$$J = \sum_{i=1}^{N} \sum_{j=1}^{p} u_{ij}^{\sigma} \left\| x_i - c_j \right\|^2 \quad (4)$$

where σ is a weighting exponent with a real number greater than 1, u_{ij} is the membership grade of x_i in the cluster j with a value between 0 and 1, x_i is the i^{th} connection of the training set, c_j is the center of cluster j, and $\| \ \|$ denotes norm expressing the distance between any measured data and the cluster center. The membership grades u_{ij} and cluster centers c_j are updated by the following expressions.

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^{\sigma} x_i}{\sum_{i=1}^{N} u_{ij}^{\sigma}} \quad (5)$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{L} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{\sigma-1}}} \quad (6)$$

By iteratively updating the cluster centers and the membership grades for each data record, FCM moves the cluster centers gradually to the correct values within the training set. Finally, the iteration stops when a termination criterion is met, that is $max_{ij} \left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right| < \varepsilon$ where ε is between 0 and 1 and k denotes the number of iterations.

The connection that lies "closer" to the center of a class has a higher membership grade to that class. On the contrary, the connection that lies "farther" away from the center of a class has a lower membership grade to that class. Training connections are grouped into p classes that each one has a certain membership grade to every class. The set of cluster centers C and membership partition matrix U are shown as follows.

$$C = \{c_1, c_2, ..., c_p\} \quad (7)$$

$$U = \{u_{i1}, u_{i2}, ..., u_{ip}\} \quad (8)$$

where i is the connection number of the training set. Each cluster center has a number of n values.

Within a vector (connection) of U, the p membership grades are treated intuitively to be our degrees of confidence on p classes that a connection can belong to. Consequently, we can build p decision rules from a connection and each one consists of a number of feature values F, class labels l, and confidence values α.

$$R_U = \{r_U\} \quad where \quad r_U : \langle F_i, L \rangle, \ \alpha \quad (9)$$

where i is the connection number. The confidence values are in proportion to the correspondent membership grades that connection belongs to certain classes. For a training connection, only portion of our belief is devoted to a certain class in a rule whereas the rest of beliefs are committed to other classes in other rules. The summation of the degrees of confidence on rules that generated from a training connection is equal to 1. It is not possible that the connection can belong to any other classes except these p classes.

$$\sum_{j=1}^{p} \alpha_{ij} = 1 \quad (10)$$

where j is the class number. Since the training set has N connections and each contains a number of p membership grades, totally N times p decision rules can therefore be generated. For example, a set of eight rules are generated if we have four connections and specify the number of classes to two. In the rule set, two rules are mapping to one connection and the degrees of confidence of them may be 0.65 and 0.35 individually.

In addition to the rules created from membership partition matrix U, a number of p rules are generated from the cluster centers. In each rule, the antecedent part includes n values of a cluster center and the corresponding class label. The degree of confidence is designated to 1 because we have full confidence that the cluster center should belong to that partitioned class without any doubt.

$$R_C = \{r_C\} \quad where \quad r_C : \langle c_j, l_j \rangle, \ \alpha = 1 \quad (11)$$

With equations 9 and 11, totally $(N+1) \cdot p$ rules are included in the decision rule set R. These rules will act as pieces of evidence to assign beliefs to an incoming connection in the decision making stage.

$$R = R_U \bigcup R_C \quad (12)$$

2) The Classification Phase

Assume v be an incoming connection to be classified. In order to classify it into the correct class, Dempster-Shafer theory is used to measure and combine pieces of evidence derived from the set of decision rules. The theory also known as *Evidence Theory* or *Theory of Believe Functions*, was introduced by Glenn Shafer in the late 1970s [4] based on the work of Arthur Dempster [3]. It is a mathematical theory of evidence and plausible reasoning; the aim is to allow evidence to be measured and combined by modeling someone's degrees of belief. The theory has been applied to solve pattern classification problems due to its capable of making decision based on conflict, uncertainty or ambiguous data.

Dempster-Shafer theory starts by defining a sample space named *frame of discernment* (or simply *frame*), which is a finite set of mutually exclusive and exhaustive hypotheses in a problem domain under consideration. For adapting the theory into our classification task, we identify the set of class labels L as the *frame* of the problem domain. The possible subset A of L represent hypothesis that one could present evidence. The set of all possible subsets of L, including itself and the null set $\varnothing$, is called a *power set* and designated as 2^L. To classify v means to assign it to one of members in L, i.e., deciding among a set of p classes: $v \in l_q$, $q = 1, 2, \ldots, p$.

A piece of evidence that influences our degree of belief concerning on a hypothesis can be quantified by a *mass function* which is denoted as m. It is a mapping function and defined as m: $2^L \rightarrow [0, 1]$ such that

$$\sum_{A \subseteq L} m(A) = 1 \quad (13)$$

$$m(\varnothing) = 0 \quad (14)$$

$A \subseteq L$ is called a *focal element* of m if $m(A)>0$. The quantity $m(A)$ is defined as the hypothesis A's *basic probability assignment*. It can be interpreted as the portion of total belief to hypothesis A given the available evidence. For example, if $m(A) = 0.2$, then it means that a one's belief committed to A is 20%. The left 80% beliefs are committed to other focal elements of frame L.

By adapting Dempster-Shafer theory, we treat the set of decision rules as pieces of evidence that alters our degrees of belief on which class v should belong to while classifying v into the correct class. If the distance is large between v and a

decision rule, it represents that *v* is "far" from the rule. It also implies that the rule only has a little influence on *v*. On the other hand, we have stronger belief that *v* should belong to the same class of the rule if *v* is "close" to it, which means the distance has a smaller value. Hence, we apply k-NN rule to find the most informative *k* nearest decision rules of *v*. Also, we use weighted k-NN rule [28] to assign different weights to these rules in order to differentiate the degrees of importance.

$$w_i = \begin{cases} \dfrac{d(x_k, v) - d(x_i, v)}{d(x_k, v) - d(x_1, v)} & d(x_k, v) \neq d(x_1, v) \\ 1 & d(x_k, v) = d(x_1, v) \end{cases} \quad (15)$$

where d is the Euclidean distance between v and a decision rule. x_i is the i^{th} nearest rule. x_k and x_1 are the farthest and nearest rule of v, respectively. The confidence value α from decision rule is added to alter the degree of our belief on v.

$$m(l_q) = w \cdot \alpha \quad (16)$$

where q is the class number. Up to this stage, each rule creates a number of belief assignment indicating the degrees that v belongs to certain classes. If the value of m is large, it means that we have a strong belief that v belongs to the class of which m indicates. Otherwise v should belong to other classes if m is small. Nevertheless, we need to notice that a belief should also be designated to the frame (with every class labels). The reason is that only part of our beliefs is committed to single classes for a given training connection, and the rest of our belief should be assigned to the whole class set. According to Dempster-Shafer theory, the summation of all mass functions inferred from one training connection is equal to 1. Thus, the belief belonged to the frame becomes one minus the summation of beliefs of all single classes.

$$m(L) = 1 - \sum_{i=1}^{p} m_i(l_q) \quad (17)$$

From the mass function given by equation 16, the *belief function Bel* and *plausibility function Pl* can be derived to characterize certain hypotheses.

$$Bel(l_j) = m(l_j) \quad (18)$$

$$Pl(l_j) = 1 - Bel(\bar{l}_j) \quad (19)$$

where j is class number and $\bar{l}_j$ is the hypothesis "not l_j" with value between 0 and 1. Belief function is a measure of the total amount of belief that directly supports for a given hypothesis. The greater the support assigns to a hypothesis, the higher belief that the hypothesis is true. It can be regarded as a lower bound that indicates the impact of evidence of the hypothesis. Plausibility quantifies the extent to which one doubts the hypothesis. It shows the belief on the given hypothesis can only up to this value, which is an upper bound on the belief. The gap between them indicates the uncertainty about the hypothesis. It is a good reference in deciding whether more evidences are needed or not.

Now let's consider an intrusion detection task and assume that the *frame* of the problem domain includes two classes: normal and attack. A network traffic connection is coming and the goal is to decide whether it is a normal activity or an attack by using belief and plausibility functions. Suppose we have two pieces of evidence regarding the connection and the mass functions are 0.1 and 0.2 for normal class and attack class, respectively. By using equations 18 and 19, the belief and plausibility that support for normal class are 0.1 and 0.8 and for attack class are 0.2 and 0.9, respectively. From the observation of the gap between belief and plausibility, it has a high degree of uncertainty. It indicates that more evidences are required to be incorporated so that we can decide the connection is a normal activity or an attack.

Generally speaking, the mass function is a piece of evidence that supports certain hypothesis concerning to the class member of a rule. When more evidences are appeared with same class label, these evidences can be integrated to generate a single belief function which represents the total support for the same class. *Dempster Rule of Combination* is applied here to combine all the beliefs induced from distinct pieces of information that with same class label together. Using this combination rule, the final belief on every subset of class set can be obtained. In our case, a number of belief functions for single classes and one belief function for the class set will be generated.

Now assume that there are two mass functions m_1 and m_2 induced by distinct items of evidence. By using *Dempster Rule of Combination*, these two independent evidences can be fused into a single belief function that expresses the support of the hypotheses in both evidences. The combination result is called *orthogonal sum* of m_1 and m_2 and noted as $m = m_1 \oplus m_2$.

$$k = \sum_{l_i \cap l_j \neq \varnothing} m_1(l_i) \cdot m_2(l_j) = 1 - \sum_{l_i \cap l_j = \varnothing} m_1(l_i) \cdot m_2(l_j) \quad (20)$$

$$m(l_i) = \left[m_1(l_i) \cdot m_2(l_i) + m_1(l_i) \cdot m_2(L) + m_1(L) \cdot m_2(l_i) \right] / k \quad (21)$$

$$m(L) = \left[m_1(L) \cdot m_2(L) \right] / k \quad (22)$$

where i and j are class number, the factor k^{-1} is called the *renormalization constant*, $m(l_i)$ are the fused mass functions of classes, and $m(L)$ is the fused mass function of the frame. Using the combination rule as described in the above equations, the final beliefs on single classes and the frame are obtained. In an intrusion detection task, a number of p belief functions for single classes and one belief function for class set will be generated. For example, totally four final belief functions are obtained if there are three classes in the frame. There are three belief functions for single classes and one belief function for the frame. They give fused allocations of belief and emphasize the agreement between multiple sources.
Let's continue on the previous example and assume that we have two more pieces of evidence regarding the same traffic connection. The mass functions of corresponding evidences are 0.3 and 0.6 for normal class and attack class, respectively. By using *Dempster Rule of Combination*, these evidences are aggregated with the previous evidences into two fused belief functions. The two fused belief functions express the total support of normal class and attack class and the results are 0.28 and 0.64, respectively.

The gap between belief and plausibility is 0.08. We can tell that uncertainty is reduced significantly after incorporating more evidences and we have stronger believe that the connection should be an attack.

TABLE I
UCI DATA SETS

Data Set	Name	Feature	Record	Class
UCI	Abalone	8	4,177	3
	Cmc	9	1,473	3
	Ionosphere	34	351	2
	Pima	8	768	2
	Wdbc	30	569	2
	Wine	13	178	3
KDD99	*Normal-DoS*	41	488,735	2
	Normal-Probe	41	101,384	2
	Normal-U2R	41	97,329	2
	Normal-R2L	41	98,403	2

At the data fusing level, each piece of evidence initializes the finite amount of belief to hypotheses of the frame. Part of the belief is allocated to the single class and part of it is allocated to the frame. To decide which class v should belong to, the *pignistic probability function* is applied to make the final decision.

$$Bp(l_q) = m(l_q) + \frac{m(L)}{p} \tag{23}$$

where q is the class number and p is the number of classes. The function quantifies our beliefs to individual classes with pignistic probability distribution. These probabilities distributed from zero to one and the summation of them equals to one. For making an optimal decision, v is assigned to a class with the highest pignistic probability.

IV. EXPERMENTIAL METHODOLOGY

For evaluating the performance of our proposed approach, we choose *DARPA KDD99* benchmark data set. In the following, we initially describe the content of the data set. We then explain the empirical settings of feature selection algorithm and fuzzy belief k-NN intrusion detection algorithm.

A. The Data Set

The data set used for the entire course of research is the *DARPA KDD99* benchmark data set, also known as "*DARPA Intrusion Detection Evaluation data set*". It includes three independent sets: whole KDD, 10% KDD, and corrected KDD. In our experiment, 10% KDD and corrected KDD are taken as our training and testing set, respectively. The training set contains a total of 22 training attack types, with an additional 17 types in the testing set only. Totally 39 attack types are included and are fall into four main classes, *Denial of Service* (*DoS*), *Probe*, *User to Root* (*U2R*), and *Remote to Local* (*R2L*).

Both training and testing sets are made up of a large number of network traffic connections and each one is represented with 41 features plus a label of either normal or a type of attack. The training set includes 494,020 connections that are distributed as 97,277 normal connections, 391,458 *DoS* attacks, 4,107 *Probe* attacks, 52 *U2R* attacks, and 1,126 *R2L* attacks. The testing set has 311,029 connections. It is made up of 60,593 normal connections, 229,853 *DoS* attacks, 4,166 *Probe* attacks, 228 *U2R* attacks, and 16,189 *R2L* attacks.

Within the four attack categories, *DoS* and *Probe* attacks continuously show up with large amounts in a short period of time when they attack systems. Generally, they have frequent sequential patterns that are different from the normal connections. Hence, they can be easily separated from normal activities. On the contrary, *U2R* and *R2L* attacks do not have any intrusion only frequent sequential patterns. They are embedded in the data portions of the packets and normally involve only a single connection. Because of this nature of *U2R* and *R2L* attacks, it becomes not easy to achieve satisfactory detection accuracies while detecting these two attacks than those of *DoS* and *Probe* attacks. In addition, the signatures in *DoS* and *Probe* attacks in the testing set provided by *KDD99* are very similar to those present in the provided training set. However, the types of *U2R* and *R2L* attacks differ significantly between the training and testing data sets. In the testing set, over 80% *U2R* attacks and 60% *R2L* attacks are new to the training set. The lack of correlation makes these two attacks harder to be identifies. Literature survey indicates that many intrusion detection systems have very low detection rates in identifying *U2R* and *R2L* attacks [29], [30]. Based on the above observations, we decide to focus on the detection of *U2R* and *R2L* attacks.

B. Empirical Setting of Feature Selection

In order to test the effectiveness of our feature selection method and compare it with other methods, we test our method in a various sizes of data sets. We apply *KDD99* data set to our feature selection algorithm to extract the most predictive features to target classes. In addition, we select six smaller data sets from *UCI* databases. Table 1 shows the data sets for evaluation. In these sets, each record is composed by a set of meaningful features. The type of features is either discrete or continuous, i.e., the former is a qualitative scale and the latter is quantitative. For qualitative scales, the values are simply labels without any order involved. They could be symbolic or numeric values which are distinct and separated. Also, it is a form of categorical data that has no "numeric" meaning. By using the features of *KDD99* data set as an example, the value of feature *protocol_type* is one of the symbolic set {icmp, tcp, udp}. The numeric value of feature *logged_in* is 1 or 0 to represent the user successfully logged in the system or not. For quantitative scales, the data are characterized by numeric values within a finite interval. The distance between any two adjacent values is not necessary the same. Examples can be found in feature *duration* where it is given by numeric values to represent the lengths of record, and the values are within an interval [0, 58329].

Since *symmetric uncertainty* is calculated for discrete features only, all the continuous features in a given data set are required to be discretized prior to the feature selection analysis. Thus, we apply discretization method to transform continuous features to discrete ones prior to the analysis. For a numeric feature, cut points effectively decompose the range of continuous values into a number of intervals. These intervals can then be treated as categorical values of a discrete feature. In our work, *equal frequency binning* technique [8] is applied to each continuous feature individually. It is an unsupervised discretization method with no class information involved. It sorts the observed values of a continuous feature and then

TABLE II
SELECTED FEATURES OF *UCI* DATA SETS

Data Set	Ours	CFS	FCBF
Abalone	3, 8	2, 3, 6, 8	8
Cmc	1, 4	2, 4	2, 4
Ionosphere	1, 5, 6, 8, 9, 16, 33, 34	1, 33	1, 33
Pima	2, 5, 6, 8	2, 5, 6	2
Wdbc	1, 3, 4, 6-8, 11, 13, 14, 21, 23, 24, 26-28	8, 21, 23, 24, 28	24
Wine	1, 7, 10-13	1, 7, 10-13	1, 2, 4, 5, 7, 10-13

TABLE III
SELECTED FEATURES OF *KDD99* DATA SETS

Data Set	Ours	CFS	FCBF
Normal-DoS	1-6, 12, 23, 24, 31, 32, 37	3, 6, 12, 37	3, 12, 31, 32
Normal-Probe	1-4, 12, 16, 25, 27-29, 30, 40	3, 4, 25, 29	3, 26, 27, 29
Normal-U2R	1-3, 10, 16	10	10, 16
Normal-R2L	1-5, 10, 22	10	5, 10, 39

divides these values into a specified number of intervals. Each of the intervals has an approximate equal number of values. With the use of discretization of features, the complexity of every continuous feature is reduced as well.

In order to evaluate the performance of our proposed feature selection algorithm on data sets, two representative feature selection algorithms, CFS and FCBF, built on the top of *symmetric uncertainty* are chosen. CFS method uses a correlation-based heuristic search algorithm to evaluate the worth of subsets of features. It considers good feature subsets contain features that are highly correlated with the class, yet uncorrelated with one another. The heuristic algorithm measures the merit of feature subsets from pairwise feature correlations and then the subset with the highest merit found during the search is reported. Rather than scoring the worth of subsets of features of CFS approach, FCBF method measures correlations between features and classes and correlations between pairs of features as well. It then selects features which are highly correlated with the class to predict but are less correlated to any feature already selected. In addition, we apply two machine learning algorithms, naive bayes and C4.5 algorithm, to evaluate the detection accuracy on selected features for each feature selection algorithm.

C. Empirical Setting of Intrusion Detection

In this stage of experiment, we reduce the sizes of the original training and testing sets by removing the duplicated connections. The reduced training set has 88,882 connections that are distributed as 87,831 normal connections, 52 *U2R* attacks, and 999 *R2L* attacks. The reduced testing set has 51,041 connections that are distributed as 47,913 normal connections, 215 *U2R* attacks, and 2,913 *R2L* attacks. Among them, features represented by symbolic values and class labels are replaced by numeric values for the use of classifiers. For example, the values of icmp, tcp, and udp of feature *protocol_type* are replaced by values 1, 2, and 3, respectively. Also, values of each feature are normalized from 0 to 1 in order to offer equal importance among features.

In order to evaluate the detection performance of our proposed intrusion detection approach, three pattern classification algorithms based on k-NN techniques are selected to compare with. One is k-NN classifier and the other two are fuzzy k-NN classifier and evidence-theoretic k-NN classifier. The k-NN classifier is simple but effective in many pattern classification applications. For an input pattern to be classified, k nearest training patterns are obtained based on the Euclidean distance measurement between the input pattern and every training pattern. The input pattern is then simply assigned to the class by majority voting, i.e., the pattern is classified to the most frequent class label among the k nearest training patterns. However, a major drawback of k-NN algorithm is that the precision of classification may decrease due to all selected k nearest training patterns are equally important without considering the differences of distances [14]. For eliminating the drawback, fuzzy k-NN classifier assigns class memberships to the input pattern rather than a single class. By using the distance differences from the k nearest training patterns, the different degrees of membership grade to classes for the input pattern are determined. As the evidence-theoretic k-NN classifier, it incorporates Dempster-Shafer theory to treat the k nearest training patterns of an input pattern as pieces of evidence to support certain hypotheses about the classes. By deriving evidences from both class labels and distances between input and k nearest training pattern pairs, these evidences are then combined into final beliefs with respect to each subset of the set of classes.

V. EXPERIMENTAL RESULTS

In the experiments, we use standard measurements such as *detection rate* (*DR*), *false positive rate (FPR*) and overall classification rates (*CR*) to evaluate the performance of intrusion detection tasks. The denotations of *True Positive* (*TP*), *True Negatives* (*TN*), *False Positive* (*FP*), and *False Negative* (*FN*) are defined as follows. Equations 24 to 26 describe *DR*, *FPR*, and *CR*, respectively.

- *True Positive* (*TP*): The number of malicious records that are correctly identified.
- *True Negatives* (*TN*): The number of legitimate records that are correctly classified.

TABLE IV
CR OF *UCI* DATA SETS USING FULL AND SELECTED FEATURE SETS

Data Set	C4.5				Naive Bayes			
	Full Set	Ours	CFS	FCBF	Full Set	Ours	CFS	FCBF
Abalone	51.90	56.00	51.90	51.90	63.23	53.60	51.90	51.90
Cmc	63.68	54.65	52.89	52.89	53.36	52.61	52.27	52.27
Ionosphere	74.93	74.93	74.93	74.93	99.15	97.72	94.02	94.02
Pima	65.10	65.10	65.10	65.10	89.97	87.50	85.03	77.34
Wdbc	62.74	62.74	62.74	62.74	99.30	99.30	99.65	94.02
Wine	94.94	94.94	94.94	94.94	98.88	97.75	97.75	98.88
Average	68.88	68.06	67.08	67.08	83.98	81.41	80.10	78.07

TABLE V
DR AND FPR OF *KDD99* DATA SETS PERFORMED ON C4.5 USING FULL AND SELECTED FEATURE SETS

Data Set	DR				FPR			
	Full Set	Ours	CFS	FCBF	Full Set	Ours	CFS	FCBF
Normal-DoS	99.97	99.97	99.86	99.31	0.04	0.03	2.19	7.58
Normal-Probe	98.51	97.78	95.52	94.91	0.02	0.38	0.36	0.36
Normal-U2R	48.08	48.08	0	7.69	0	0	0	0
Normal-R2L	93.52	97.69	0	27.44	0.01	0.01	0	0.02
Average	85.02	85.88	48.85	57.34	0.02	0.11	0.64	1.99

TABLE VI
DR AND FPR OF *KDD99* DATA SETS PERFORMED ON NAIVE BAYES USING FULL AND SELECTED FEATURE SETS

Data Set	DR				FPR			
	Full Set	Ours	CFS	FCBF	Full Set	Ours	CFS	FCBF
Normal-DoS	99.12	99.16	99.37	99.19	0.01	0.01	2.76	7.77
Normal-Probe	98.27	96.54	62.53	45.31	1.29	0.87	0.15	0.10
Normal-U2R	82.69	69.23	0	7.69	0.63	0.50	0	0
Normal-R2L	99.11	93.25	0	33.84	1.31	0.49	0	0.08
Average	94.80	89.55	40.48	46.51	0.81	0.47	0.73	1.99

- *False Positive* (*FP*): The number of records that were incorrectly identified as attacks however in fact they are legitimate activities.
- *False Negative* (*FN*): The number of records that were incorrectly classified as legitimate activities however in fact they are malicious.

$$DR = \frac{TP}{TP + FN} \tag{24}$$

$$FPR = \frac{FP}{TN + FP} \tag{25}$$

$$CR = \frac{TP + TN}{TP + TN + FP + FN} \tag{26}$$

A. Feature Selection

We perform the feature selection experiments on the six *UCI* data sets and binary classification (normal/attack) of *KDD99* data set. Four new sets of data are generated according to the normal class and four categories of attack (*DoS*, *Probe*, *U2R*, and *R2L*). In each data set, records with the same attack category and all the normal records are included. For each data set, we run our proposed approach and the other two feature selection algorithms CFS and FCBF, and record these selected features from each algorithm. Throughout the entire experiments, the threshold of FCBF is set to 0. Having finished the feature selection procedures, we then apply C4.5 and naive bayes machine learning algorithms on each original full data set as well as each newly obtained data set that includes only those selected features from feature selection algorithms. By applying 10-fold cross-validation evaluation on each data set, we get the classification accuracies of these experimental data sets.

Tables 2 and 3 show the results of feature selection of *UCI* and *KDD99* data sets, respectively. Table 4 summarizes the classification accuracies of six *UCI* data sets. Tables 5 and 6 summarize the percentages of DRs and FPRs performed on four *KDD99* data set using C4.5 and naive bayes algorithms, respectively. For an intrusion detection task, abnormal activities are expected to be correctly identified and normal activities are anticipated not to be misclassified. Therefore, a higher DR and a lower FPR are desired. For each data set, the highest DR and the lowest FPR are highlighted

From the results shown in Table 4, we observe that our approach achieve higher averaged classification accuracies in comparison with the outcomes of CFS and FCBF feature selection algorithms while small data sets are applied. Especially in the experiment of the abalone data, we get the highest classification accuracy by using 2 out of 8 features performed on C4.5 learning algorithm, which is better than that of using full feature set. In the experiment in cmc data set, 2 out of 9 features are selected from all of three algorithms, but we achieve the highest CR. The averaged accuracies of Tables 5 and 6 also show that our approach outperforms over both CFS and FCBF feature selection algorithms while using large data sets. Among the averaged DRs shown in Table 5, we reach the highest accuracy. Our approach also has the best performance of averaged FPR shown in Table 6.

TABLE VII
AVERAGED RATES OF FOUR CLASSIFIERS PERFORMED ON *NORMAL-U2R* DATA SET WITH *K* RANGING FROM 1 TO 10

Classifier	Full Set		Ours		CFS		FCBF	
	FPR	DR	FPR	DR	FPR	DR	FPR	DR
k-NN	0.18	13.51	2.57	18.84	0.19	14.56	0.19	14.80
Fuzzy k-NN	0.29	15.59	2.53	18.75	0.19	14.44	0.20	15.14
Evidence-Theoretic k-NN	0.31	17.11	2.64	19.67	0.23	17.14	0.25	18.59
Fuzzy Belief k-NN	14.75	95.67	9.59	83.21	0.25	12.20	0.13	6.87

TABLE VIII
AVERAGED RATES OF FOUR CLASSIFIERS PERFORMED ON *NORMAL-R2L* DATA SET WITH *K* RANGING FROM 1 TO 10

Classifier	Full Set		Ours		CFS		FCBF	
	FPR	DR	FPR	DR	FPR	DR	FPR	DR
k-NN	0.35	17.41	3.80	18.68	0.28	13.90	2.75	17.77
Fuzzy k-NN	0.36	18.75	18.50	21.67	0.28	14.47	3.11	20.55
Evidence-Theoretic k-NN	0.41	19.57	5.64	23.71	0.30	16.54	5.04	26.16
Fuzzy Belief k-NN	11.38	66.81	9.73	69.02	0.21	5.98	0.21	6.40

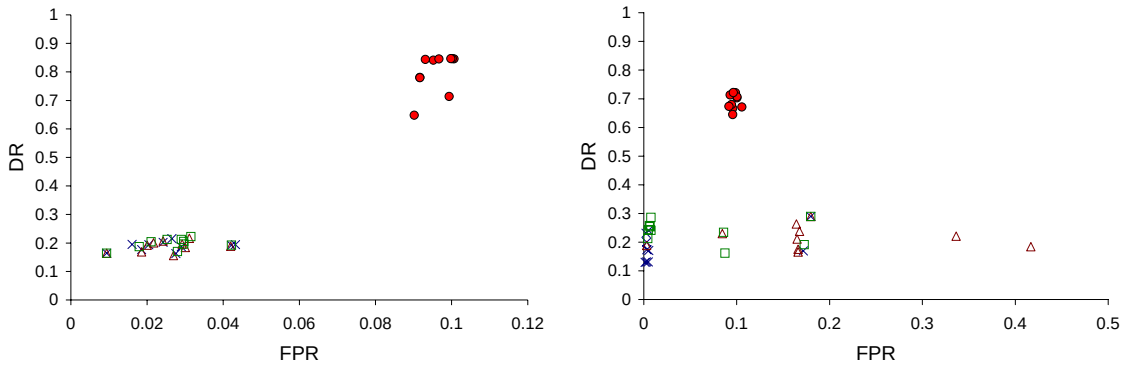

Fig. 3 Results of four classifiers performed on *Normal-U2R* (left) and *Normal-R2L* (right) data sets using our selected features with *k* ranging from 1 to 10
x: k-NN, Δ: fuzzy k-NN, □: evidence-theoretic k-NN, •: fuzzy belief k-NN

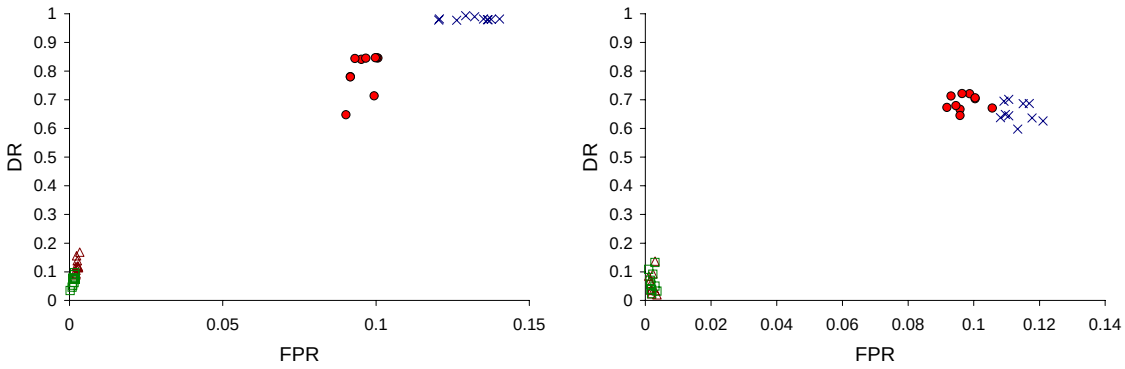

Fig. 4 Results of fuzzy belief k-NN classifier performed on *Normal-U2R* (left) and *Normal-R2L* (right) data sets using full feature set and three feature subsets with *k* ranging from 1 to 10
x: full feature set, Δ: CFS, □: FCBF, •: ours

In the *Normal-DoS* data set, the difference in DRs is very slight for all of the feature selection algorithms. By using our approach performed on C4.5 learning algorithm, we get the highest DR 99.97% and the lowest FPR 0.03%. In the *Normal-Probe* data set, both CFS and FCBF approaches fail to achieve an acceptable presentation on DRs while using naive bayes algorithm, whereas our approach gains the best detection performances performed on both C4.5 and naive bayes algorithms. In the *Normal-U2R* and *Normal-R2L* data sets, we have satisfactory performances, especially we get the highest detection accuracies using C4.5 learning algorithm. Though CFS and FCBF approaches achieve low FPRs, they have very poor detection operations.

Generally, FPRs are low in the result of any one of feature selection algorithm because sufficient normal records present in all of four data sets. For the number of misclassification attack connections, our approach provides acceptable DRs in *Normal-DoS*, *Normal-Probe*, and *Normal-R2L* data sets. It is

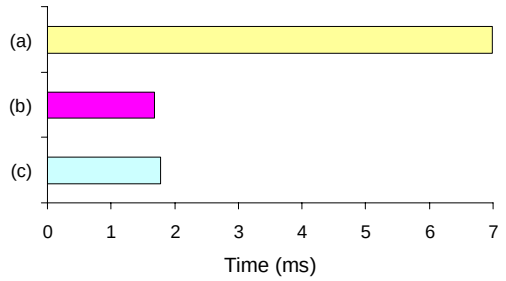

Fig. 5 Detection processing time of one connection using fuzzy belief k-NN classifier
(a) With all 41 features
(b) With 5 selected features in *Normal-U2R* set
(c) With 7 selected features in *Normal-R2L* set

not only because each of the above data set supplies sufficient attack records but also most of attacks in the data set have same attack signatures. For instance in the *Normal-DoS* data set, the *DoS* attack includes near 400,000 data records and distributes in 10 different attacks. Most of them are *netpune* and *smurf* attacks that account for around 99%. In the *Probe* attack category, 95% of attacks are *ipsweep*, *portsweep*, and *satan* that are distributed in 4107 attacks. As for *R2L* attack class, more than 90% of attacks are *warezclient* attacks while 8 different kinds of attacks present in the *Normal-R2L* data set. In contrast, the classification presented on *Normal-U2R* data set is satisfactory neither on full feature set approach nor on any one of feature selection algorithms. The reason is because the *Normal-U2R* data set only includes 52 attack records which are insufficient for learning on any classification algorithm either using full feature set or subset of features.

B. Intrusion Detection

Having reduced the complexity of the preliminary large feature space, we then incorporate the feature selection results into four k-NN based classifiers to classify the traffic data into normality or abnormality. To minimize the inaccuracy and variation factor of experiment results, 10 trials are performed in every *Normal-U2R* and *Normal-R2L* detection task. In order to simulate the uncertainties caused by limited and ambiguous information of network traffic data, a very small amount of normal and attack connections are randomly selected from reduced training and testing sets in each trial. In *Normal-U2R* data sets, the training and testing sets comprise 930 (878 normal and 52 *U2R*) and 694 (479 normal and 215 *U2R*) connections, respectively. In *Normal-R2L* data sets, the training and testing sets include 977 (878 normal and 99 *R2L*) and 770 (479 normal and 291 *R2L*) connections, respectively.

For detecting the attacks, training and testing are performed in each trial. In the training phase, the four classifiers, k-NN, fuzzy k-NN, evidence-theoretic k-NN, and fuzzy belief k-NN, are constructed using the training data. The testing data are then fed into each trained classifier to identify intrusions in the testing phase. We evaluate the performances of four classifiers using distinct numbers of k nearest neighbors. Tables 7 and 8 summarize the averaged rates of *Normal-U2R* and *Normal-R2L* data sets with k ranging from 1 to 10, respectively.

TABLE IX
AN EXAMPLE OF C4.5 DECISION RULE

IF *wrong_fragment* < 3 AND
num_compromised < 1 AND
srv_serror_rate < 0.06 AND
rerror_rate < 0.06 AND
flag = SF AND *hot* < 1 AND
protocol_type = tcp AND
service = http
THEN normal connection

TABLE X
CLASSIFICATION PERFORMANCES
BEFORE AND AFTER ADDING C4.5 DECISION RULES

	Normal-U2R		*Normal-R2L*	
	FPR	DR	FPR	DR
Before	9.59	83.21	9.73	69.02
After	3.11	83.21	3.10	68.77

In the comparison of four classifiers performed in different feature sets, k-NN, fuzzy k-NN and evidence-theoretic k-NN classifiers have similar detection performances using either full feature set or one of selected feature subsets, which all the three k-NN based classifiers have poor detection performances. The maximum DRs in rows 1 to 3 of Tables 7 and 8 are 19.67% and 26.16% for *Normal-U2R* and *Normal-R2L* data sets, respectively. With our proposed fuzzy belief k-NN classifier, the results of using three feature selection algorithms are differ a lot, which our selected features provide much accurate DRs than those from CFS and FCBF. In the last row, the DRs of our approach reach 83.21% and 69.02% for *Normal-U2R* and *Normal-R2L* data sets, respectively. Although both CFS and FCBF achieve low FPRs in the data sets, it is because they treat most of the network traffic data as normal usages no matter the traffic are normal or malicious activities. For a better demonstration of our proposed classifier outperforms than the other three k-NN based classifiers, Figure 3 shows the Receiver Operating Characteristics (ROC) graphs of four classifiers performed on *Normal-U2R* and *Normal-R2L* data sets using our selected feature subset with k ranging from 1 to 10. It shows the points of k-NN, fuzzy k-NN, and evidence-theoretic k-NN classifiers are all gathering near point (0, 0), which indicates that none of them can correctly identify attacks. However, all the points of fuzzy believe k-NN classifier are much closer to point (0, 1), which have higher DRs and have lower FPRs as well.

In Figure 4, we show the result of fuzzy belief k-NN classifier using full feature set and three feature subsets selected by our approach, CFS, and FCBF. For both data sets using features from CFS and FCBF, the diagrams show that all of the points are in the vicinity of (0, 0), which represents all the traffic are classified as normal activities and only a very few amount of attacks are correctly detected. In the left diagram, the DR with full feature set is higher than that of using our feature subset, however our selected features provide a better FPR result than that of using full feature set. In the right diagram, we could notice that the points with our selected features are closer to point (0, 1) than those of using full feature set, which show that our selected features achieve

better detection outcomes in both DR and FPR than those of using full feature set. In addition to the consideration of detection performance, we furthermore consider the detection processing time because an intrusion detection system has to perform its analysis as quick as possible before the attacks make any damage to the protected system. Consequently, we compare the computation time of fuzzy belief k-NN classifier using full feature set and our selected feature subset. Figure 5 illustrates the detection time on each testing connection of both data sets. The results show that we successfully reduce the computation time if our selected feature subset is used, which our approach only needs 0.24 and 0.25 of the time of using full feature set in *Normal-U2R* and *Normal-R2L* data sets, respectively.

With our proposed fuzzy belief k-NN classifier, the averaged FPRs using our selected feature subset are 9.59% and 9.73% for *Normal-U2R* and *Normal-R2L* data sets, respectively. For increasing the correct identification number of normal connections, we create ten C4.5 decision rules using the training set for these two data sets. Table 9 shows an example rule. Table 10 summarizes the averaged FPRs and averaged DRs of fuzzy belief k-NN classifier with and without adding C4.5 decision rules. It is obviously that both percentages of DRs remain in the same level, but FPRs are significantly decreased from around 10% to 3%, i.e., the number of misclassified normal connections is decreased.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we present an information-theoretical feature selection algorithm on both low and high dimensional feature spaces. Correlation analysis is employed to measure the strengths of feature-feature and feature-class. In order to evaluate the feasibility of our selected feature subset, we compare our result with outcomes from CFS and FCBF feature selection algorithms in C4.5 and naive bayes learning algorithms. The experimental results demonstrate that our feature selection algorithm has a superior performance. We then verify the performance of our proposed fuzzy belief K-NN classifier which is based on the combination of k-nearest neighbors, fuzzy clustering technique, and Dempster-Shafer theory. In this stage, we compare the performance of our approach with those of three k-NN based classifiers, k-NN, fuzzy k-NN, and evidence-theoretic k-NN classifiers. During the experiments, we only include a very small amount of network traffic data to simulate uncertainties caused by limited and ambiguous information. The results show that our approach has a superior performance to the other three classifiers. Also, the detection processing time is significantly reduced if our selected feature subset is employed. In the future, we will continue on our research of improving detection performance of both normal and malicious activities.

REFERENCES

[1] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, 1988.

[2] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

[3] J. C. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters," *Journal of Cybernetics*, vol. 3, pp. 32-57, 1973.

[4] A. P. Dempster, "A Generalization of Bayesian Inference," *Journal of the Royal Statistical Society*, Series B, vol. 30, pp. 205-247, 1968.

[5] G. Shafer, *A Mathematical Theory of Evidence, Princeton*, University Press, Princeton, NJ, 1976.

[6] E. Fix and J. L. Hodges, "Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties," Report Number 4, Project Number 21-49-004, *USAF School of Aviation Medicine*, Randolph Field, Texas, 1951.

[7] KDD99 archive: The Fifth International Conference on Knowledge Discovery and Data Mining. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[8] C. L. Blake and C. J. Merz, UCI Repository of Machine Learning Databases, 1998.

[9] M. Hall, *Correlation Based Feature Selection for Machine Learning*, Doctoral Dissertation, The University of Waikato, Department of Computer Science, 1999.

[10] L. Yu and H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," in *Proceedings of The Twentieth International Conference on Machine Leaning*, pp. 856-863, Washington, D.C., August, 2003.

[11] T. M. Mitchell, *Machine Learning*, McGraw-Hill, New York, 1997.

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

[13] M. Keller, M. R. Gray, and J. A. Givens Jr., "A Fuzzy k-Nearest Neighbor Algorithms," *Transactions on Systems, Man and Cybernetics*, vol. SMC-15(4), pp. 580-585, 1985.

[14] T. Denoeux, "A k-Nearest Neighbor Classification Rule Based on Dempster-Shafer Theory," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 5, pp. 804-813, May 1995.

[15] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157-1182, 2003.

[16] J. M. Booker, M. C. Anderson, M. A. Meyer, "The Role of Expert Knowledge in Uncertainty Quantification (Are We Adding More Uncertainty or More Understanding?)," in *Seventh Army Conference on Applied Statistics*, pp. 155-161, 2001.

[17] W. L Oberkampf, J. C. Helton, C. A. Jos lyn, S. F. Wojtkiewicz, and S. Ferson, "Challenge Problems: Uncertainty in System Response Given Uncertain Parameters," *Reliability Engineering and System Safety*, vol. 85 pp. 11-19, 2004.

[18] K. Jones and R. S. Sielken, *Computer System Intrusion Detection: A Survey*, Technical Report, Computer University of Virginia, 2000.

[19] G. John, R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem," in *Proceedings ML-94*, pp. 121-129, Morgan Kaufmann, 1994.

[20] K. Kira and L. A. Rendell, "The Feature Selection Problem: Traditional Methods and a New Algorithm," in *Proceedings AAAI-92*, pp. 129-134, MIT Press, 1992.

[21] H. Almuallim and T. G. Dietterich, "Learning with Many Irrelevant Features," in *Proceedings AAAI-91*, pp. 547-551, MIT Press, 1991.

[22] G. Qu, S. Hariri, and M. Yousif, "A New Dependency and Correlation Analysis for Features," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 9, pp. 1199-1207, September 2005.

[23] H. G. Kayacık, A. N. Zincir-Heywood, and M. I. Heywood, "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets," in *Third Annual Conference on Privacy, Security and Trust*, St. Andrews, New Brunswick, Canada, October 2005.

[24] J. R Quinlan, "Induction of Decision Trees," *Machine Learning*, vol. 1, pp. 81-106, 1986.

[25] G. Stein, B. Chen, A. S. Wu, and K. A. Hua, "Decision Tree Classifier For Network Intrusion Detection With GA-based Feature Selection," in *Proceedings of the 43rd ACM Southeast Conference*, Kennesaw, GA, March 2005.

[26] S. Mukkamala and A. H. Sung, "Feature Selection for Intrusion Detection Using Neural Networks and Support Vector Machines", *Journal of the Transportation Research Board of the National Academics*, Transportation Research Record No 1822, pp. 33-39, 2003.

[27] J. Biesiada and W. Duch, "Feature Selection for High-Dimensional Data: A Kolmogorov-Smirnov Correlation-Based Filter Solution," in

Proceedings of the 4th International Conference on Computer Recognition Systems, 2005.

[28] S. A. Dudani, "The Distance-Weighted k-NN Rule," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 6, no. 4, pp. 325-327, 1976.

[29] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman, "Evaluating Intrusion Detection Systems: the 1998 DARPA Off-Line Intrusion Detection Evaluation," in *Proceedings of the 2000 DARPA Information Survivability Conference and Exposition*, vol. 2, IEEE Press, January 2000.

[30] M. Sabhnani and G. Serpen, "Why Machine Learning Algorithms Fail in Misuse Detection on KDD Intrusion Detection Data Set," *Intelligent Data Analysis*, vol. 8, no. 4, pp. 403-415, 2004.