Cyber Security Threats Detection Using Ensemble Architecture

Te-Shun Chou
Department of Technology Systems, East Carolina University
Greenville, NC, U.S.A.
chout@ecu.edu

Abstract

This paper describes an ensemble design for cyber security threats detection, which fuses the results from multiple classifiers together to make a final assessment decision. For promoting both speed and accuracy in the detection performance, only some of the features in traffic data are selected for each base classifier. In the kernel of each classifier, we combine Dempster-Shafer theory with k-nearest neighbor technique to solve the uncertainty problems caused by ambiguous and limited intrusion information. In addition, we apply data mining techniques to reduce the number of false alarms. The results indicate that our ensemble approach achieves higher detection rates than that of using a full feature set of classifiers.

Keywords: *Intrusion detection, ensemble learning, feature selection, Dempster-Shafer theory*

1. Introduction

Today almost no one can exclude himself or herself from using the Internet. Like getting a cold or the flu once or twice a year, to Internet users, Internet attacks seem unavoidable. People take flu shots to prevent getting a virus and Internet users need protection to keep their network secure from attacks as well. Hence, intrusion detection systems play an important role in modern network security. To design an intrusion detection system, a variety of techniques have been proposed over the past years. They are mainly categorized into two groups: anomaly detection techniques and misuse detection techniques. Misuse detection techniques model patterns of known attacks. By simply matching signatures of traffic records with previous well defined attack patterns, activities can be declared as intrusions if any mismatch happens. The recognized attacks are detected in an efficient way with a high level of accuracy. However, it is difficult to cover all possible variations of attacks using misuse detection technique because computer attacks are usually polymorphic [1]. Computer hackers use different approaches to exploit a same vulnerability. The attack code looks different from the known signature but is functionally equivalent. It reassembles as it hits the target machine [2]. For example, the Internet worms are polymorphic and spread automatically across networks by exploiting vulnerabilities [3]. These worms are able to mutate as they spread across the network by using self-encryption mechanisms or semantics-preserving code manipulation techniques. A minor variation of an attack may not be identified during the whole detection procedure. On the other hand, Anomaly detection techniques search for intrusive activities by comparing network traffic to those established acceptable normal usage patterns learned from training data. If the pattern of observed data is different from those learned normal ones, the data is classified as an attack. It offers the ability to resist polymorphic attacks at the moment that novel attacks are introduced into a system. Because these polymorphic, novel attacks are constantly being introduced to the networks today it is necessary to be agile and prepared to stop them before they do significant damage

In this paper we propose an ensemble intrusion detection architecture that includes a set of base feature selecting classifiers and a data mining classifier. The multiple base feature selecting classifiers employ anomaly detection techniques to search for abnormal behavior simultaneously and then their decisions are fused together to reach a result. With the use of a set of divergent feature subsets in base feature selecting classifiers, we expect that the fused outcome is more accurate than those of individual ones. We also believe that the detection speed using partial feature space will be faster than that of using a full feature set. However, the anomaly detection approach always has a higher false alarm rate. Hence, misuse detection technique is employed to compensate for the disadvantage of the result inferred from the set of base feature selecting classifiers. We use a data mining technique to derive decision rule of normal activities in training set and expect the rate of false alarms can be effectively decreased.

This paper is organized as follows. Section 2 presents the related works. Section 3 describes the theoretical framework in our ensemble intrusion detection approach. We then demonstrate the experimental methodology, followed by a discussion of the experimental results. Finally, we conclude our work in the last section.

2. Related Works

From the decision-based perspective, intrusion detection in fact is a classification task, i.e., to classify network traffics into normal usage category or attack category. When using one single classifier to solve a classification problem, the classifier always uses a specific machine learning algorithm to deal with that problem from its own point of view. However there is no classifier which can completely cover all aspects of the problem. Therefore, an ensemble of multiple base classifiers catches people’s attention and becomes one of the major research topics in the field of machine learning.

An ensemble approach combines the outputs from a set of base classifiers together in a proper way when classifying input data. The fused result is expected to perform a better outcome than that of any individual base classifier within the ensemble. In the schemes of building an ensemble classifier, three distinct topologies are frequently engaged. They are: cascading, parallel, and hierarchical structures [4]. In the cascading structure, the output from the previous base classifier is fed into the next one. By cascading all the base classifiers together, the final result is obtained at the last base classifier’s output of the chain. In the parallel structure, the predictions of base classifiers are integrated to produce a fused output of the ensemble. The hierarchical structure is a combination of cascading and parallel configurations.

The types of decision generated by the individual base classifier can be classified into three major categories: abstract form, rank level, and measurement level [5]. The abstract form occurs when a classifier only outputs a solitary class label for an input pattern. The rank level occurs when a classifier ranks a list of classes in accordance with the degrees of belief on classes the input pattern belongs to. The list is always sorted in a descending way so that the first and the last components are the highest and lowest ranked output classes, respectively. Finally, the measurement level occurs when the classifier assigns a level of confidence to each class for expressing the classifier’s degree of belief for an input pattern. Among the combination methods that work with abstract form outputs, the popular methods are: behavior knowledge space method, majority voting, weighted majority voting, naive bayes method. For measurement level outputs, the combination methods are MAX, MIN, SUM, PROD, AVG, and MED methods, from which the ensemble selects the maximum, minimum, summation, product, average, or median value of the combined classifiers as its output.

While applying an ensemble technique to intrusion detection design, two major approaches have been used in the base classifiers. One uses different feature subsets in base classifiers and the other uses different soft computing techniques. The former technique consists of a set of base feature selecting classifiers and each uses partial feature space. By choosing dissimilar feature subsets for various base feature selecting classifiers, the diversity among these classifiers is expected to be maximized to achieve a better result. One good example of this process is the work of Giacinto and Roli [6]. In their research, they restricted the problem domain in the ftp service of the *DARPA KDD99* data set [6] and selected 30 out of the 41 available features from the data set. They built three neural networks using 4 intrinsic features, 19 traffic features, and 7 content features, respectively. Also, they built one neural network using all of the 30 selected features for the sake of comparison. All of the networks were three layers fully-connected multi-layer networks, each of which had 5 output neurons (for normal and four attack classes), a number of input neurons that equaled the number of features, and a hidden layer made up of 5 neurons for the networks using distinct features and 15 neurons for the network trained using 30 selected features. The results showed that the ensemble technique improved the overall detection performance compared with those of individual classifiers and the classifier using 30 features. In the work of DeLooze [8], Delooze created three 20×20 Self-Organizing Maps (SOM) using content, time, and connection features extracted from 41 features of *KDD99* data set. The predictions of individual SOMs were then combined using both majority ensemble method and belief ensemble method.

The work of Borji [9] is an example using different soft computing technique in every individual base classifier. He used *KDD99* training data set in both training and test procedures to perform five-class (normal, *DoS*, *Probe*, *U2R*, and *R2L*) classification. First, Borji used four base classifiers (neural networks, SVM, k-nearest neighbor (k-NN) and decision trees) to advance classification individually and then fused their inferences using three combination strategies: majority voting, average rule and belief function. Another example can be found in the work of Mukkamala et al. [10]. They also used *KDD99* training data set and performed five-class (normal, *DoS*, *Probe*, *U2R*, and *R2L*) classification. They designed two ensemble models: one consisted of three multilayer feedforward neural networks and the other was made up of neural networks, Support Vector Machine (SVM) and Multivariate Adaptive Regression Splines (MARS). By using the majority voting technique, the outcomes from individual base classifiers were combined together. Zainal et al. [11] proposed an ensemble architecture by using three soft computing algorithms, Linear Genetic Programming (LGP), Adaptive Neural Fuzzy Inference System (ANFIS) and Random Forest (RF) to classify five categories shown on *KDD99* data set. Their individual predictions were combined into a final result using voting technique.

3. Theoretical Framework

Figure 1 depicts our designed ensemble intrusion detection model. It includes a set of base feature selecting classifiers and one data mining classifier to act as anomaly detection and misuse detection, respectively. Generally speaking, the entire intrusion detection process is divided into three layers. In the first layer, we implement two ensemble classifiers and each is constructed by three individual base feature selecting classifiers. Within each base classifier, partial feature space is used to model the system behavior from a single aspect.

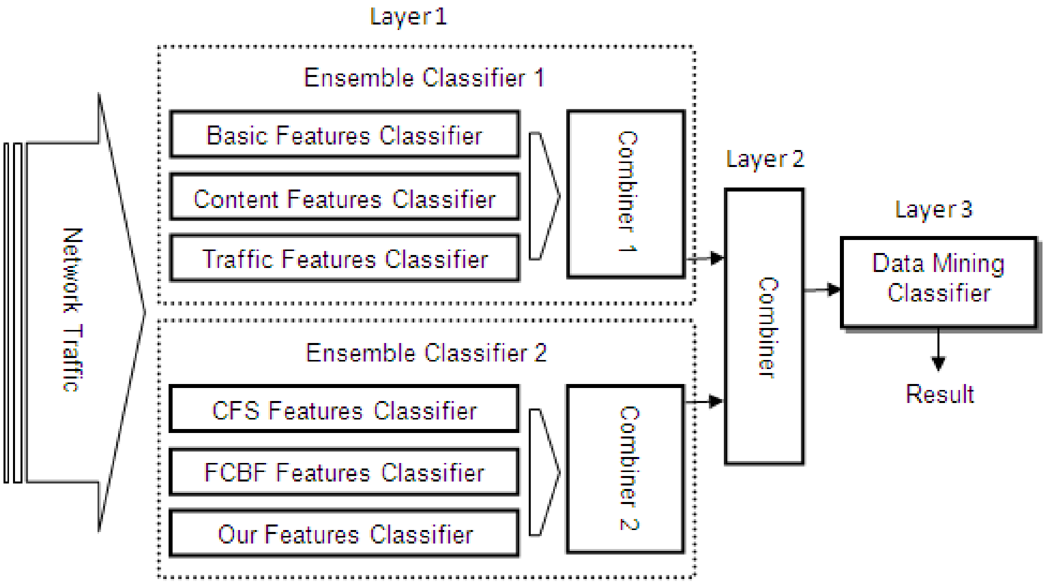


Figure 1. Proposed Intrusion Detection Model

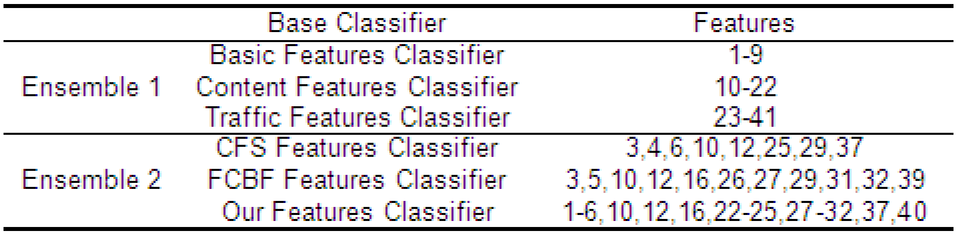
Having finished the derivations from three base classifiers, an ensemble output is generated. Then, two ensemble outputs are integrated to produce a fused output. Finally, data mining technique is applied and an ultimate result is obtained.

3.1 Ensemble Classifier

In an ensemble intrusion detection design, it is important to understand that individual base classifiers should be independent of each other. If the base classifiers provide similar outputs, then no significant improvement of the ensemble result can be obtained through the combination process. It is critical to notice the diversity among base classifiers in order to get effective and correct detection results. Hence, we use *ensemble feature selection* approach by choosing dissimilar feature subsets for base classifiers. The diversity among these base classifiers is expected to be maximized in order to achieve a best ensemble result.

The ensemble architecture consists of two ensemble classifiers and each includes three base feature selecting classifiers. In each base classifier, a subset of features is used to derive independent decision about input traffic data. Then all the decisions are combined into a fused decision. In order to evaluate the performance of our proposed approach, the *DARPA KDD99* benchmark data set [6] is used. It includes a large volume of traffic connection, and each includes 41 features such as protocol type, network service, and status flag. By varying the feature subsets and maximizing the disagreement among base classifiers, distinct feature representations from the original feature are selected. In ensemble classifier 1, three base classifiers use 9 basic features, 13 content features, and 19 traffic features. In ensemble classifier 2, three base classifiers use feature subsets deriving from our developed correlation-based feature selection algorithm [12] and two other algorithms CFS [13] and FCBF [14]. Table 1 illustrates the detailed feature information used in six base classifiers. Within each base classifier, a machine learning algorithm, *k*-NN belief intrusion detection algorithm, based on evidence-theoretic *k*-NN theory [15], is employed to derive predictions about input network traffic connection.

Table 1. Selected Features for Base Classifiers



	Base Classifier	Features
Ensemble 1	Basic Features Classifier	1-9
	Content Features Classifier	10-22
	Traffic Features Classifier	23-41
Ensemble 2	CFS Features Classifier	3,4,6,10,12,25,29,37
	FCBF Features Classifier	3,5,10,12,16,26,27,29,31,32,39
	Our Features Classifier	1-6,10,12,16,22-25,27-32,37,40

3.2 *k*-NN Belief Intrusion Detection Algorithm

When an input traffic connection needs to be classified, the base classifier incorporates Dempster-Shafer theory to treat the *k* nearest training connections of the input 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 connections, these evidences are then combined into beliefs with respect to each subset of the set of classes. Figure 2 depicts the general operation scheme of the proposed approach. The details are described as follows.

Let's assume the available information in a given training set is from a network with *N* 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 traffic connection as *x*, and the set of features in each connection as *F*.

$$T = \{x_1, x_2, ..., x_N\} \tag{1}$$

$$F = \{f_1, f_2, ..., f_n\} \tag{2}$$

Assume *v* be an incoming connection to be classified. In order to classify it into a 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 [16] based on the work of Arthur Dempster [17]. 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 the members in *L*, i.e., to assign *v* to a member 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

$$\sum_{A \subseteq L} m(A) = 1 \tag{3}$$

$$m(\varnothing) = 0 \tag{4}$$

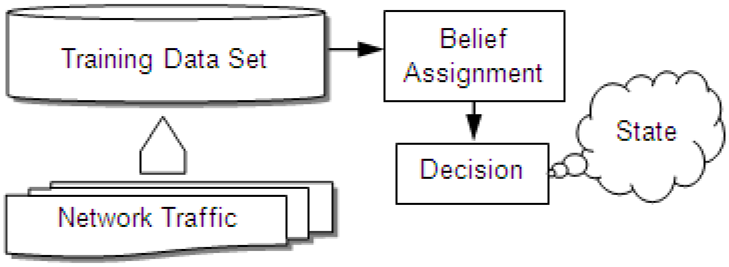


Figure 2. Belief Intrusion Detection Identification

$A \subseteq L$ 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 training set as T as pieces of evidence that alters our degrees of belief about to which class v should belong,, while classifying v into the correct class. If the distance is large between v and a traffic connection in T, it represents that v is "far" from the connection, i.e., the connection only has a little influence on v. On the other hand, we have a stronger belief that v should belong to the same class of the traffic connection if v is "close" to it, which means the distance has a smaller value. Hence, we apply the k-NN rule to find the most informative k nearest traffic connections of v. Also, we use weighted k-NN rule [18] to assign different weights to these rules in order to differentiate the degrees of importance.

$$w_i = \begin{cases} \dfrac{d(x_k,v)-d(x_i,v)}{d(x_k,v)-d(x_1,v)} & d(x_k,v) \neq d(x_1,v) \\ 1 & d(x_k,v) = d(x_1,v) \end{cases} \tag{5}$$

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

$$m(l_q) = w \cdot \alpha \tag{6}$$

where q is the class number. Up to this stage, each k nearest traffic connection creates a number of belief assignments indicating the degrees to which 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 all class labels). The reason is that only part of our beliefs are 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.

$$m(L) = 1 - \sum_{i=1}^{p} m_i(l_q) \tag{7}$$

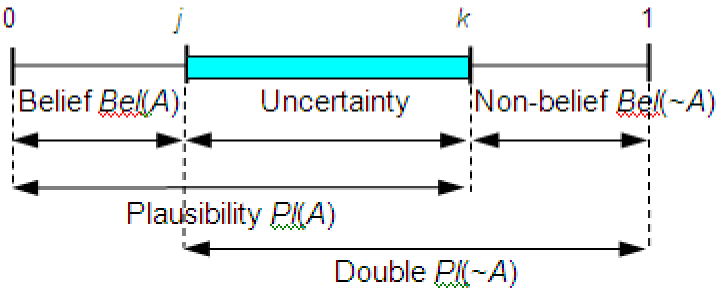


Figure 3. Functions of Belief and Plausibility

From the mass function given by Equation 6, the *belief function Bel* and *plausibility function Pl* can be derived to characterize certain hypotheses.

$$Bel(l_j) = m(l_j) \tag{8}$$

$$Pl(l_j) = 1 - Bel(\bar{l}_j) \tag{9}$$

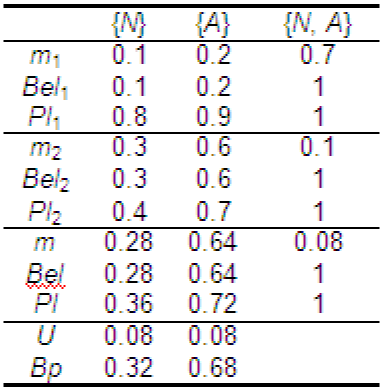
where j is class number and $\bar{l}_j$ is the hypothesis "not l_j" with value between 0 and 1. Belief function is a measure of the total amount of belief that directly supports for a given hypothesis. The greater the support assigns to a hypothesis, the higher belief that the hypothesis is true. It 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 evidence is needed as proof for one's hypotheses. Haralick and Shapiro [19] represent these various measurements over the interval unit in Figure 3.

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. The mass functions are 0.1 and 0.2 for normal class and attack class, respectively. By using Equations 8 and 9, 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. This indicates that more evidences are required to be incorporated so that we can decide whether 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 appear with the 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 with the 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 *X* and *Y*. By using *Dempster Rule of Combination*, these two independent pieces of evidence can be fused into a single belief function that expresses the support of the

Table 2. Data Fusion Result



	{N}	{A}	{N, A}
m_1	0.1	0.2	0.7
Bel_1	0.1	0.2	1
Pl_1	0.8	0.9	1
m_2	0.3	0.6	0.1
Bel_2	0.3	0.6	1
Pl_2	0.4	0.7	1
m	0.28	0.64	0.08
Bel	0.28	0.64	1
Pl	0.36	0.72	1
U	0.08	0.08	
Bp	0.32	0.68	

hypotheses in both. The combination result is called *orthogonal sum* of m_1 and m_2 and noted as $m = m_1 \oplus m_2$.

$$m(Z) = \frac{\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)}{\sum_{X \cap Y \neq \varnothing} m_1(X) \cdot m_2(Y)} = \frac{\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)}{1 - \sum_{X \cap Y = \varnothing} m_1(X) \cdot m_2(Y)} = \left(\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y) \right) \cdot k^{-1} \qquad (10)$$

where the factor k^{-1} is the *renormalization constant*. 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 the class set will be generated. For example, a total of 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 evidence 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. Table 2 shows the detailed result where normal and attack are abbreviated as *N* and *A*, respectively and uncertainty between belief and plausibility is abbreviated as *U*. We can tell that uncertainty is reduced significantly after incorporating more evidences and we have a stronger belief that the connection should be an attack.

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.

$$Bp(l_q) = m(l_q) + \frac{m(L)}{p} \qquad (11)$$

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 one. For making an optimal decision, v is assigned to a class with the highest pignistic probability.

3.3 Combination Method

Besides the notability of multiplicity among the base feature selecting classifiers, the combination method is another important issue to decide if the ensemble result is successful or not. A careful choosing of combination methods can lead to the ensemble result being one of extraordinary accuracy, while an improper selection of combination methods can lead to a poor accuracy result. Research has shown that there are many combination methods available for combining the abstract form outputs of the base classifiers into an ensemble result. However, some of them in fact are not suitable for our ensemble intrusion detection design. For example, the behavior knowledge space method requires enough representative data sets to estimate high order distribution of classifiers outputs. Otherwise overfitting is likely to occur, and the generalization error quickly increases [20]. But in our intrusion detection task, the training set only provides a very small amount of traffic data and it is insufficient to offer behavior knowledge space method a representative number of observations. While the majority voting ensemble approach is used for integration, we cannot guarantee that all the detection accuracies of our designed base classifiers would satisfy the requirement of Hansen and Salamon [21]. Accordingly, naive bayes ensemble is chosen to obtain an ensemble result. Based on the probabilistic approach, the evidences of base classifiers are computed and the most appropriate class can then be obtained. Its operation is explained as follows.

Let the possible classes of the system be $l_1, l_2, ..., l_p \in L$ and these p classes are mutually exclusive and exhaustive, i.e., the decision result of the system is definitely in only one of the classes. $D = \{d_1, d_2, ..., d_m\}$ is a set of base classifiers. It includes a number of m base classifiers and each is built from a feature subset of feature space $F = \{f_1, f_2, ..., f_n\}$. The output of an individual base classifier o is an abstract form class label. The objective is to find the probability over a class member l conditional on outcomes from m base classifiers h_1 through h_m.

$$P(h_1, \ldots, h_m \mid l_i) = \prod_{j=1}^{m} P(o_{ij} \mid l_i) \tag{12}$$

$$P(h_1, \ldots, h_m) = \sum_{i=1}^{p} \big(P(h_1, \ldots, h_m \mid l_i) P(l_i)\big) \tag{13}$$

$P(h_1, ..., h_m \mid l_i)$ is the conditional probability of $h_1, ..., h_m$ given l_i. $P(o_{ij}|l_i)$ is the conditional probability of o_{ij} given l_i. $P(l_i)$ is the prior probability of each class. $P(h_1, ..., h_m)$ is the probability of $h_1, ..., h_m$.

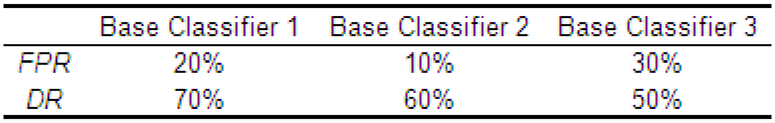
Based on the Bayes theorem, we have:

$$P(l_i \mid h_1, ..., h_m) = \frac{P(h_1, ..., h_m \mid l_i) P(l_i)}{P(h_1, ..., h_m)} \tag{14}$$

where i is the number of possible classes ranging from 1 to p. This posterior probability collects all evidences from base classifiers and integrates them together. Finally, the naive bayes combiner infers the state of system by choosing a class that achieves the highest posterior probability.

$$\theta = \arg\max_{l_i} P(l_i \mid h_1, \ldots, h_m) \tag{15}$$

Table 3. Detection Accuracies of Three Base Feature Selecting Classifiers



	Base Classifier 1	Base Classifier 2	Base Classifier 3
FPR	20%	10%	30%
DR	70%	60%	50%

To illustrate how naive bayes ensemble method works, we assume there are three base feature selecting classifiers h_1, h_2, and h_3 that are built by three different subsets of features. The intrusion detection task is a binary assignment, i.e., each base classifier assigns the network traffic data into either normal activity l_1 or attack l_2.

Suppose we have tested an amount of network traffic data and the distributions of the normal activities and attacks of the testing set are 60% and 40%, respectively. The detection rates (*DR*s) and false positive rates *(FPR*s) of three base classifiers are shown in Table 3. The goal is to classify a future network traffic data x into normal activity if $P(l_1 \mid h_1, h_2, h_3) > P(l_2 \mid h_1, h_2, h_3)$, else into attack category. Assume a traffic data passes through these three base classifiers. The first and second classifiers identify it as a normal use , however the third one recognizes it as an attack. Then,

$$P(h_1, h_2, h_3 \mid l_1) = \prod_{j=1}^{3} P(o_{1j} \mid l_1) = (1-0.2)\cdot(1-0.1)\cdot 0.3 = 0.216 \tag{16}$$

$$P(h_1, h_2, h_3 \mid l_2) = \prod_{j=1}^{3} P(o_{2j} \mid l_2) = (1-0.7)\cdot(1-0.6)\cdot 0.5 = 0.06 \tag{17}$$

$$P(h_1, h_2, h_3) = \sum_{i=1}^{2} P(h_1, h_2, h_3 \mid l_i) P(l_i) = 0.216 \cdot 0.6 + 0.06 \cdot 0.4 = 0.1536 \tag{18}$$

Based on the above calculation and the prior probabilities of normal activities and attacks, the joint posterior probabilities of $P(l_1 \mid h_1, h_2, h_3)$ and $P(l_2 \mid h_1, h_2, h_3)$ are:

$$P(l_1 \mid h_1, h_2, h_3) = \frac{P(h_1, h_2, h_3 \mid l_1) P(l_1)}{P(h_1, h_2, h_3)} = \frac{0.216 \cdot 0.6}{0.1536} = 0.84375 \tag{19}$$

$$P(l_2 \mid h_1, h_2, h_3) = \frac{P(h_1, h_2, h_3 \mid l_2) P(l_2)}{P(h_1, h_2, h_3)} = \frac{0.06 \cdot 0.4}{0.1536} = 0.15625 \tag{20}$$

We have 84.375% degree of confidence that the incoming network traffic data x belongs to class l_1 which is greater than that of class l_2. We therefore conclude x is a normal computer user activity.

3.4 Data Mining Classifier

Having finished the process of ensemble feature selecting intrusion classification, another important concern is the problem of false alarm rate. Because the entire scope of both normal and attack behavior is covered during the classification procedure, a great deal of study has shown that one of the most common problems of anomaly intrusion detection is too many false alarms might happen likely resulted from normal behavior. Hence, we suggest a model using two levels that combines parallel and serial topologies together for getting a better quality of detection. In the second level data mining classifier, we utilize data mining technique to construct a filter to eliminate false alarms.

Data mining techniques provide strategy to find useful information from a large amount of data and induce inferences from those information. Here we use C4.5 decision trees algorithm to extract patterns from training data. The goal is to find rules that represent normal behavior of network traffic stream for our intrusion detection task. In this way, we can write our decision rule as follows.

Rule: IF *conditions of features* THEN *the traffic is a normal behavior*

Within the rule, the *antecedent* part consists of a number of conditions that are satisfied by features. The *consequent* action is defined as the analyzed network traffic data as a normal behavior.

The data mining classifier compares the result derived from the second layer with decision rules, and the normal computer activities to the system can be identified if the data is matched with one of defined rules. The data mining classifier has a higher priority to determine whether a traffic data is a normal behavior or not if it has a disagreement with the result of the combiner.

4. Experimental Methodology

4.1. The Data Set

The *DARPA KDD99* benchmark data set is chosen for analyzing the performance of our proposed approach. 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 data set and testing data set, respectively. The original training and testing data sets have 494,020 and 311,029 connections, respectively. Each connection is composed of 41 features plus a label of either normalor a type of attack. The training set contains a total of 22 attack types, with an additional 17 types in the testing set only. There are a total of 39 attack types included and they fall into four main classes: Denial of Service (*DoS*), *Probe*, User to Root (*U2R*), and Remote to Local (*R2L*).

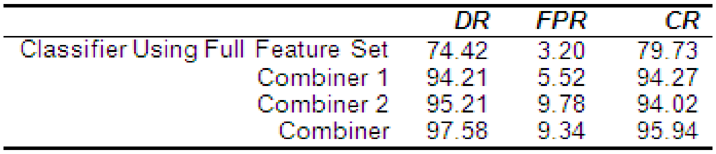
- *Denial of Service* (*DoS*) attacks: Attackers disrupt a host or network service in order to make legitimate users unable to have access to a machine;
- *Probe* attacks: Attackers use programs to automatically scan networks for gathering information or finding known vulnerabilities;
- *User to Root* (*U2R*) attacks: Local users get access to root access of a system without authorization and then exploit the machine’s vulnerabilities; and
- *Remote to Local* (*R2L*) attacks: Unauthorized attackers gain local access from a remote machine and then exploit the machine’s vulnerabilities.

4.2. Preprocessing

In order to reduce the sizes of the training and testing sets, the duplicated connections are removed from the original data sets. The new training set has 145,585 connections that are distributed as 87,831 normal connections, 54,572 *DoS* attacks, 2,131 *Probe* attacks, 52 *U2R* attacks, and 999 *R2L* attacks. The new testing set has 51,041 connections that are distributed as 47,913 normal connections, 23,568 *DoS* attacks, 2,682 *Probe* attacks, 215 *U2R* attacks, and 2,913 *R2L* attacks.

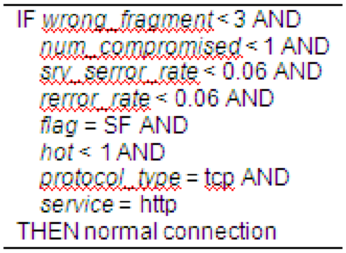
For each connection, features represented by symbolic values and class labels are replaced by numeric values. For example, the values of *icmp*, *tcp*, and *udp* of feature *protocol_type* are replaced by values 1, 2, and 3, respectively. Also, values of each feature are normalized from 0 to 1 in order to offer equal importance among features. Class labels,

Table 4. Ensemble Results vs. Classifier Using Full Feature Set



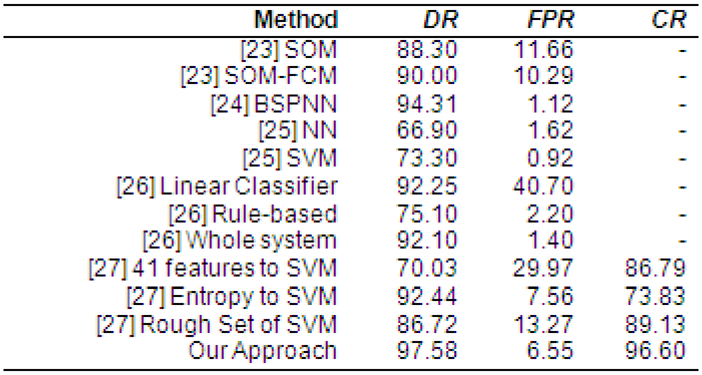
	DR	*FPR*	*CR*
Classifier Using Full Feature Set	74.42	3.20	79.73
Combiner 1	94.21	5.52	94.27
Combiner 2	95.21	9.78	94.02
Combiner	97.58	9.34	95.94

Table 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 6. Comparison results



Method	*DR*	*FPR*	*CR*
[23] SOM	88.30	11.66	-
[23] SOM-FCM	90.00	10.29	-
[24] BSPNN	94.31	1.12	-
[25] NN	66.90	1.62	-
[25] SVM	73.30	0.92	-
[26] Linear Classifier	92.25	40.70	-
[26] Rule-based	75.10	2.20	-
[26] Whole system	92.10	1.40	-
[27] 41 features to SVM	70.03	29.97	86.79
[27] Entropy to SVM	92.44	7.56	73.83
[27] Rough Set of SVM	86.72	13.27	89.13
Our Approach	97.58	6.55	96.60

normal, *DoS*, *Probe*, *R2L*, and *U2R*, are replaced by 1, 2, 3, 4, and 5, respectively. A class label with values 1 and 2 is added to indicate normal traffic and attacks (*DoS*, *Probe*, *R2L*, and *U2R*), respectively. In addition, equal frequency binning technique [22] is applied to transform continuous features to discrete ones.

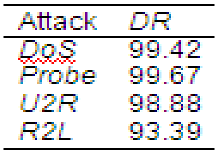
4.3. Data Selection

Generally, a set of network traffic is necessary to be collected in advance for designing intrusion detection systems. Based on the collected data set, misuse detection specifies well defined attack signatures and anomaly detection constructs acceptable user behavior. However, it is difficult to collect all attack information because in real world hackers constantly develop new attack codes to exploit security vulnerabilities of organizations. The collected data always encloses uncertainty when only limited information about intrusive activities is available. Accordingly, in order to simulate the problem of uncertainty existing in the *KDD99* data set, only a small number of normal connections and attacks are randomly selected from training and testing sets for each experiment. In the training set, all 52 *U2R* attacks and 999 *R2L* attacks are included. For balancing the distribution of normal traffic and each attack group, 1000 connections are randomly selected from normal class and each remaining attack group (*DoS*, *Probe*, and *U2R*). In the testing set, all 215 *U2R* attacks are included. Also, 1000 connections are randomly selected from normal class and each remaining attack group (*DoS*, *Probe*, and *R2L*).

5. Experimental Results

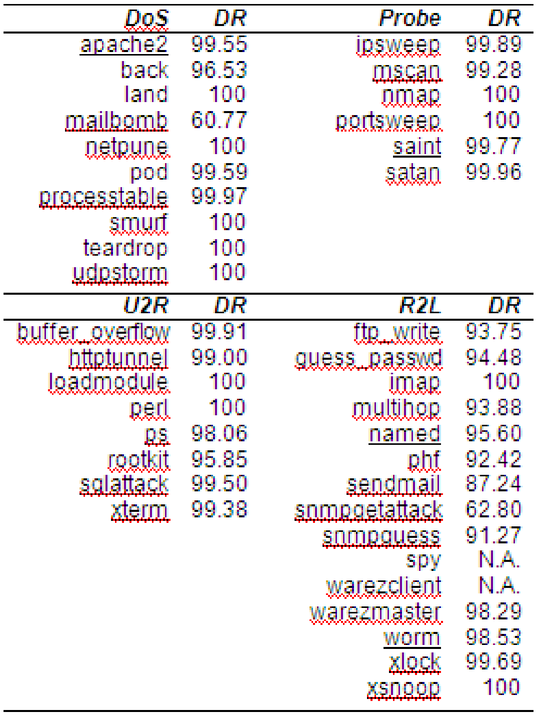
The experiments are performed on the binary (normal/attack) detection and evaluated using standard measurements detection rate (*DR*), false positive rate *(FPR*) and overall classification rate (*CR*). To minimize the inaccuracy and variation factor of experiment results, 10 trials are performed in every intrusion detection task. In each trial, we evaluate the performances of proposed approach using distinct values of *k* nearest neighbors that ranges from 1 to 10. Table 4 summarizes the averaged accuracies of ensemble outcomes in the first and second layers and *k*-NN belief intrusion detection classifier using 41 full feature set. The experimental results show that the classification accuracy in the second layer is improved after fusing two outputs derived from the first layer. Although the single classifier using full feature set has a better *FPR* compared with those from ensemble approaches, it has the worst performance of *DR*, which implies either normal connections or malicious attacks are classified into normal behavior.

Table 7. Detection Rates on Four Attack Groups of Final Result



Attack	*DR*
DoS	99.42
Probe	99.67
U2R	98.88
R2L	93.39

Table 8. Detection Rates of 39 Attacks



DoS	***DR***	***Probe***	***DR***
apache2	99.55	ipsweep	99.89
back	96.53	mscan	99.28
land	100	nmap	100
mailbomb	60.77	portsweep	100
netpune	100	saint	99.77
pod	99.59	satan	99.96
processtable	99.97		
smurf	100		
teardrop	100		
udpstorm	100		
U2R	***DR***	***R2L***	***DR***
buffer_overflow	99.91	ftp_write	93.75
httptunnel	99.00	guess_passwd	94.48
loadmodule	100	imap	100
perl	100	multihop	93.88
ps	98.06	named	95.60
rootkit	95.85	phf	92.42
sqlattack	99.50	sendmail	87.24
xterm	99.38	snmpgetattack	62.80
		snmpguess	91.27
		spy	N.A.
		warezclient	N.A.
		warezmaster	98.29
		worm	98.53
		xlock	99.69
		xsnoop	100

Having finished the combination process in layer 2, the objective is to further reduce the number of legitimate connections that are incorrectly identified as attacks. Hence in the third layer, we utilize one decision rule originated from data mining technique to correct those misclassified normal activities. Table 5 shows the decision rule that covers 90% of normal behavior in the training set. With the combination of both anomaly and misuse techniques, the decisions of network traffic data are finally obtained in layer 3. Table 6 shows the final performance of our model and those of [23]-[27] with binary detection approach. With our designed ensemble intrusion detection model, we achieve higher accuracies in comparison with the outcomes of other methods.

We then further analyze the detection accuracies of four attack groups and Table 7 shows the result. From the values we observe, our model performs well in detecting *DoS*, *Probe*, and *U2R* attacks with *DR*s around 99% but a relative low 93.39% *DR* in *R2L* attacks. To further investigate 39 individual attacks showing in those four attack groups, Table 8 describes the detailed detection rates of all attacks. The attacks shown in testing set but not in the training set are marked by underlining. The result shows that our model is capable of detecting most attacks, especially eleven attacks (*land*, *netpune*, *smurf*, *teardrop*, *udpstorm*, *nmap*, *portsweep*, *loadmodule*, *perl*, *imap*, and *xsnoop*) which are detected perfectly. However, the model achieves low *DR* in detecting *mailbomb* and *snmpgetattack* attacks.

6. Conclusions

In this paper, we apply the ensemble technique to our intrusion detection task and develop an ensemble feature selection approach, which includes six base classifiers that are created by diverse subsets of *KDD99* data set. In each base feature selecting classifier, we apply Dempster-Shafer theory to solve uncertainties caused by limited information. Also, we use data mining technique to extract decision rules for normal behavior in order to construct a filter to reduce the rate of false alarms. Finally, we have an integrated testing of our proposed model. The experimental results demonstrate that this three-layer hierarchy structure improves detection performance.

References

[1] S. Zanero and S. M. Savaresi, “Unsupervised Learning Techniques for an Intrusion Detection System,” Proceedings of the 14th ACM Symposium on Applied Computing, 2004.

[2] Z. Li, A. Das, and J. Zhou, “Model Generalization and its Implications for Intrusion Detection,” Proceedings of Applied Cryptography and Network Security, Lecture Notes in Computer Science, June 2005.

[3] C. Kruegel, E. Kirda, D. Mutz, W. Robertson, and G. Vigna, “Polymorphic Worm Detection Using Structural Information of Executables,” 8th Symposium on Recent Advances in Intrusion Detection, Lecture Notes in Computer Science, Springer Verlag, USA, September 2005.

[4] Y. Lu, “Knowledge Integration in a Multiple Classifier System,” Application Intelligence, 6(2), pp. 75–86, 1996.

[5] L. Xu, A. Krzyzak and C.Y. Suen, “Several Methods for Combining Multiple Classifiers and Their Applications in Handwritten Character Recognition,” IEEE Transactions on System, Man and Cybernetics, SMC-22(3), pp. 418-435, 1992.

[6] G. Giacinto and F. Roli, “Intrusion Detection in Computer Networks by Multiple Classifier Systems,” 16th International Conference on Pattern Recognition, Volume 2, pp. 390-393, 2002.

[7] KDD’99 archive: The Fifth International Conference on Knowledge Discovery and Data Mining.
URL: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[8] L. L. DeLooze, “Attack Characterization and Intrusion Detection using an Ensemble of Self-Organizing Maps,” 2006 International Joint Conference on Neural Networks, pp. 2121-2128, Vancouver, BC, Canada, July, 2006.

[9] A. Borji, “Combining Heterogeneous Classifiers for Network Intrusion Detection,” Lecture Notes in Computer Science, Springer, Volume 4846, pp. 254-260, 2008.

[10] S. Mukkamala, A. H. Sung, and A. Abraham, “Intrusion Detection Using an Ensemble of Intelligent Paradigms,” Journal of Network and Computer Applications, Volume 28, Issue 2, pp. 167-182, 2005.

[11] A. Zainal, M. A. Maarof, and S. M. Shamsuddin, "Ensemble of classifiers for detecting network intrusion," International Conference on Advances in Computing, Communication and Control archive, pp. 510-515, 2009.

[12] T. S. Chou, K. K. Yen, J. Luo, N. Pissinou, and K. Makki, "Correlation-Based Feature Selection for Intrusion Detection Design," IEEE Military Communications Conference, pp. 1-7, Orlando, FL, October 2007.

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

[14] 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.

[15] T. Denoeux, "A k-Nearest Neighbor Classification Rule Based on Dempster-Shafer Theory," IEEE Transactions on Systems, Man and Cybernetics, Volume 25, Number. 5, pp. 804-813, May 1995.

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

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

[18] S. A. Dudani, "The Distance-Weighted k-NN Rule," IEEE Transactions on Systems, Man and Cybernetics, Volume 6, Number 4, pp. 325-327, 1976.

[19] R. Haralick and L. Shapiro, *Computer and Robot Vision*, Volume 2, Addison-Wesley, 1993.

[20] S. Raudys and F. Roli, "The Behavior Knowledge Space Fusion Method: Analysis of Generalization Error and Strategies for Performance Improvement," Proceedings of International Workshop on Multiple Classifier Systems, pp. 55–64, Guildford, Surrey, June 2003.

[21] L. K. Hansen and P. Salamon, "Neural Network Ensembles," IEEE Transactions on Pattern Analysis Machine Intelligence, 12(10), pp. 993-1001, 1990.

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

[23] M. Jazzar and A. Jantan, "A Novel Soft Computing Inference Engine Model for Intrusion Detection," IJCSNS International Journal of Computer Science and Network Security, Volume 8 Number 4, pp. 1-9, April 2008.

[24] T. P. Tran, L. Cao, D. Tran, and C. D. Nguyen "Novel Intrusion Detection using Probabilistic Neural Network and Adaptive Boosting," International Journal of Computer Science and Information Security, Volume 6, Number 1, pp. 83-91, 2009.

[25] A. Osareh and B. Shadgar, "Intrusion Detection in Computer Networks based on Machine Learning Algorithms," International Journal of Computer Science and Information Security, Volume 8, Number 11 pp. 15-23, 2008.

[26] Z. Bankovic, J. M. Moya, Á. Araujo, S. Bojanic, and O. Nieto-Taladriz, "A Genetic Algorithm-based Solution for Intrusion Detection," Journal of Information Assurance and Security 4, 192-199, 2009.

[27] R. C. Chen, K. F. Cheng, and C. F. Hsieh, "Using Rough Set and Support Vector Machine for Network Intrusion Detection," International Journal of Network Security & Its Applications, Volume 1, Number 1, pp.1-13, April 2009.

Authors

Dr. Te-Shun Chou is an Assistant Professor in the Department of Technology Systems at East Carolina University. He received his Bachelor degree in Electronics Engineering and both Master's degree and Doctoral degree in Electrical and Computer Engineering at Florida International University. His primary research interests are network security, especially in intrusion detection.