Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Ensemble of Machine Learning Algorithms for Intrusion Detection

Te-Shun Chou[1], Jeffrey Fan[2], Sharon Fan[2], and Kia Makki[2]
[1]Department of Technology Systems, East Carolina University, Greenville, NC 27858, U.S.A.
[2]Department of Electrical and Computer Engineering, Florida International University, Miami, FL 33174, U.S.A.

Abstract—**Ensemble-classifier is a technique that uses a combination of multiple classifiers to reach a more precise inference result than a single classifier. In this paper, a three-layer hierarchy multi-classifier intrusion detection architecture is proposed to promote the overall detection accuracy. For making every individual classifier is independent from others, each uses a diverse soft computing technique as well as different feature subset. In addition, the performances of a variety of combination methods that fuse the outputs from classifiers are studied. In the experiments, DARPA KDD99 intrusion detection data set is chosen as the evaluation tools. The results show that our approach achieves a better performance than that of a single classifier.**

Keywords—**Intrusion detection, ensemble design, feature selection, machine learning**

I. INTRODUCTION

While designing an intrusion detection system, detection accuracy is an important consideration. The system needs to perform a proper detection task with high detection rate on malicious activities but low false alarms on normal computer usages. In the past, intrusion detection approaches based on ensemble techniques have been investigated. Ensemble is to combine the outputs of 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. However, 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 classification result. Hence, two major categories have been proposed in the ensemble classifier design. One uses different feature subset in every base classifier and the other uses different soft computing technique.

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. Example is the work of Giacinto and Roli [1]. In their research, they restricted the problem domain in the ftp service of the *DARPA KDD99* data set [2] 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 networks using all of the 30 selected features for the sake of comparison. All of the networks were three layers fully-connected multi-layer networks, which each had 5 output neurons (for normal and four attack classes), a number of input neurons that equal to 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. However they only performed their experiments on ftp service instead of all of the services *KDD99* data set provided. In the work of DeLooze [3], he created three 20×20 Self-Organizing Maps (SOM) using content, time, and connection features extracted from 41 features of *KDD99* data set. The results of individual SOMs were then combined using both majority ensemble method and belief ensemble method. Here, the difficulty is how to configure a network with proper size. The configuration plays an important role in the detection performance and the granularity of the network nodes, which training a SOM with a large amount of neurons needs long computational time and a SOM with a small volume of neurons may loss some important information.

The work of Borji [4] 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 as well as performed five-class (normal, *DoS*, *Probe*, *U2R*, and *R2L*) classification. He firstly 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. He claimed his ensemble model overall got 99.68% detection rate (*DR*) and 0.87% false positive rate (*FPR*). However, he did not mention *DR* and *FPR* in each class. Also, we argue that if his experimental result still performed so well if *KDD99* testing set was included in his experiment. The reason is that the testing set has extensive new types of attacks that are not correlated with attacks shown in the training set. Another example can be found in the work of Mukkamala et al. [5]. They also used *KDD99* training data set and performed five-class (normal, *DoS*, *Probe*, *U2R*, and *R2L*) classification. They designed two

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

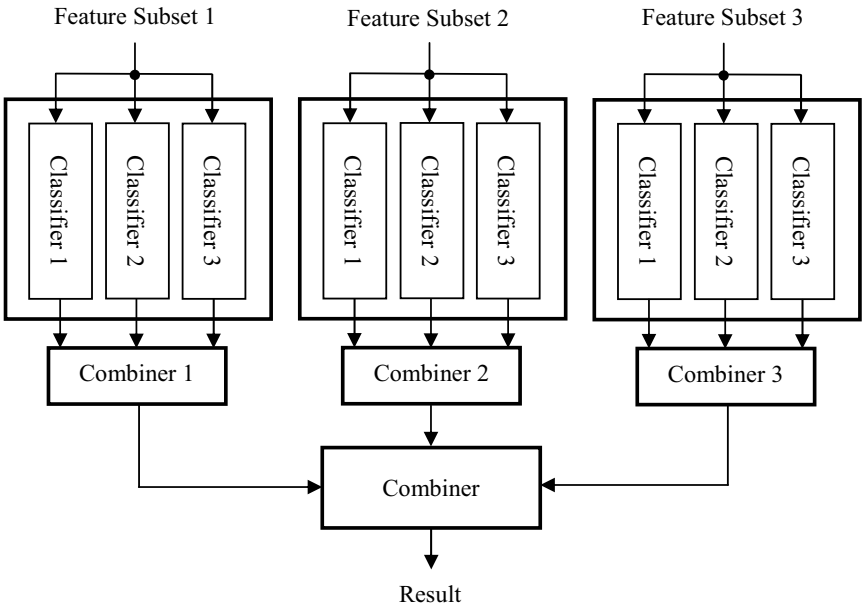

Figure 1. Topologies of Proposed Intrusion Detection System

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 then combined together. The experimental result showed that the ensemble approach produced a better result than that of each base classifier. In one of their experiments, they fused three base classifiers' outputs with 48%, 0%, and 16% accuracies together and get 56% ensemble accuracy. However, Hansen and Salamon [6] had proved that multi-classifiers will only work when it is possible to build individual classifiers which are more than 50% accurate. Furthermore, they used the same data set in both training and test procedures, which the experimental results cannot explain the detection ability of novel attacks.

For a successful ensemble intrusion detection design, whether each classifier is independent and dissimilar to others play an important role. A good design of base classifiers is their outputs should be divergent to each other as much as possible. Hence, we propose a three-layer hierarchy multi-classifier intrusion detection architecture as illustrates in Fig. 1. In the first layer, three groups are constructed and each consists of a set of three base feature selecting classifiers. In order to promote the diversity, different soft computing techniques as well as different feature spaces are applied to the base feature selecting classifiers. In the second layer, the inferences derived from three base feature selecting classifiers in the same group are integrated. Then the outputs from three groups are fused together to produce a final conclusion of the ensemble in the third layer.

This paper is organized as follows. Section 2 presents the design of proposed intrusion detection approach. Section 3 demonstrates the experimental methodology. We then describe a discussion of the experiment results. Finally, in the last section we present the conclusions and future work.

II. PROPOSED APPROACH

There is often no clear boundary between normal and abnormal of a computer user's activity. Patterns of attacks are sometimes similar to those of normal activities. Therefore in the kernel of base feature selecting classifiers we select a variety of supervised learning techniques that can provide capability of dealing with vagueness: fuzzy *k*-NN classifier, naive bayes classifier, and backpropagation neural network classifier. All of them are capable of providing a dynamic decision boundary of network traffic instead of only assigning network traffic to a member of normal category or a member of abnormal category. During the entire course of work, the intrusion detection benchmark data set *KDD99* is used for training and testing those different soft computing models. For maximizing the diversity of the ensemble, three partial feature subsets, 9 basic features (1-9), 13 content features (10-22), and 19 traffic features (23-41), of the original *KDD99* 41 features are applied to three base feature selecting classifiers.

Having finished the process of base feature selecting classifiers' derivations in each group,, all the decisions from multiple ones are combined into a fused result. Finally, the predictions of three groups are then integrated to produce an ultimate conclusion of the ensemble. In order to evaluate the result of different combination methods, we carried out four fusion techniques: the majority voting rule, the average rule, Dempster-Shafer technique, and Bayesian combination method.

A. Backpropagation Neural Network Classifier

A backpropagation neural network uses a feedforward structure to solve classification problems by its supervised

learning algorithm. It consists of a collection of processing units that are highly interconnected. The network weights are updated by using gradient-based optimization algorithm during the training period. When the network converges to the local minima of error, the output layer of the network will show the result when data is fed into the input layer.

Based on the data given for training, neural networks has the ability to learn how to process intrusion detection tasks. It acts as a computational model to process the network traffic information. By the use of training procedure, the neural network gains the knowledge to extract the normal and attack signatures from the provided data automatically. With its ability to generalize from learned data, the neural network performs generalization of attacks and fault tolerance to imprecise and uncertain information. At the end of the training procedure, the future network traffic are then identified as whether malicious attacks or normal usage behavior.

B. *Fuzzy k-NN Classifier*

The *k*-NN classifier is simple but effective in many pattern classification applications. For an input to be classified, a number of *k* nearest training patterns are obtained based on the Euclidean distance measurement between the input and every training pattern. The input is then simply assigned to the class by majority voting, i.e., the input 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 if all selected *k* nearest training patterns are equally important without considering the differences of distances [7]. Furthermore, while processing an intrusion detection task, some of the intrusive patterns are similar to those of normal activities. The boundaries between those attacks and the normal behavior are always unclear. To eliminate this drawback, fuzzy *k*-NN classifier is proposed and fuzziness is introduced into it. It assigns multiple membership grades to classes rather than a single class by the use of the distance differences from the *k* nearest training patterns. The confidence values are in proportion to the correspondent membership grades that the input network traffic belongs to certain classes.

C. *Naive Bayes Classifier*

The naive bayes classifier is based on conditional probabilistic to perform decision of a classification problem. It uses Bayes' Theorem with independence assumptions, which assumes a set of features are conditionally independent of one another given a class. When a set of classes are observed in the training data, the naive bayes classifier then assign an observed data to one of classes with highest probability.

By applying naive bayes classifier to an intrusion detection task, a set of training network traffic data is given to find the prior probabilities for normal or a known class of attacks. As an unseen network traffic arrives, the classifier then uses Bayes Theorem to decide which class the traffic should belong to.

D. *Combination Methods*

Besides the notability of multiplicity among the base classifiers, the right choice of a combination method is also an important issue in creating a supreme performance. A variety of combination methods have been reported for combining the outputs of the base classifiers into an ensemble result. According to their characteristics, they can be classified as linear combination methods, non-linear methods, statistical-based methods, and computationally intelligent methods. Linear combination method is the simplest method to fuse base classifiers' outputs together. Summation and average are the popular ways for the combination. Non-linear method such as majority voting is used when the output of classifier is a ranked list of classes in accordance with the degrees of belief on classes the input pattern belongs to. Statistical-based methods are Dempster-Shafer techniques and Bayesian combination methods. The computationally intelligent method is based on computational intelligence techniques such as fuzzy logic, neural networks, and genetic algorithms.

For comparing the performance of different combination operations in our intrusion detection task, we carry out four fusion techniques: the majority voting rule, the average rule, Dempster-Shafer technique and Bayesian combination method to combine the outputs together. With equal posterior estimation distribution of classifiers' output, the majority voting rule assigns the input network traffic to the majority class among the outputs of classifiers. The average rule assigns the input network traffic to the maximum value of the posterior probability summation divided by the number of classifiers we implemented. As for the Dempster-Shafer and Bayesian combination methods, both assign the input network traffic to the class with highest belief value. The difference between them is that the Bayesian combination method involves the computation of the prior probability of each class but Dempster-Shafer technique does not, while it computes the probability that evidences support the attack or normal classes we consider.

III. EXPERIMENTAL METHODOLOGY

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" that not only includes a large quantity of network traffic but also collects a wide variety of attacks. It includes three independent sets: whole *KDD*, 10% *KDD*, and corrected *KDD*. The 10% *KDD* contains a total of 22 attack types, with an additional 17 types in the corrected *KDD* only. Totally 39 attack types are included and are fall into four main classes:

- *Denial of service (DoS) attacks*: Attackers disrupt a host or network service to make legitimate users can not access to a machine, e.g. ping-of-death and SYN flood;
- *Remote to Local (R2L) attacks*: Unauthorized attackers gain local access from a remote machine and then exploit the machine's vulnerabilities, e.g. guessing password;
- *User to Root (U2R) attacks*: Local users get access to local machine without authorization and then exploit the machine's vulnerabilities, e.g. various "buffer overflow" attacks; and

- *Probe*: Attackers use programs to automatically scan networks for gathering information or finding known vulnerabilities, e.g. port scanning and ping sweep.

In our experiment, 10% *KDD* and corrected *KDD* are taken as our training set and testing set, respectively. Both data sets are made up of a large volume of network traffic connections describing *TCP* connections and each includes 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.

B. Preprocessing

In the beginning of the experiment, we reduce the sizes of the original training and testing sets by removing the duplicated connections. 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 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. Values of each feature are normalized from 0 to 1 in order to offer equal importance among features. Class labels, normal, *DoS*, *Probe*, *R2L*, and *U2R*, are replaced by 1, 2, 3, 4, and 5, respectively. In addition, a class label with values 1 and 2 is added to indicate normal traffic and attacks (*DoS*, *Probe*, *R2L*, and *U2R*), respectively.

C. Data Selection

Although the *KDD99* data set includes 39 different types of attacks, the problem of uncertainty exists caused by limited information of network traffic data. In real world modern computer systems and networks, hackers constantly develop new attack codes to exploit security vulnerabilities of organizations everyday. It is impossible to cover all intrusive behavior space in the collected data set. Accordingly, in order to simulate the problem of uncertainty, only a small amount of normal and attack connections are randomly selected from training and testing sets in each experiment. In the training set, all the 52 *U2R* attacks and 999 *R2L* attacks are included. For balancing the distribution of normal traffic and each attack group, 999 connections are randomly selected for normal class and each attack group (*DoS*, *Probe*, and *U2R*). In the testing set, all the 215 *U2R* attacks are included. Also, 215 connections are randomly selected for normal class and each attack group (*DoS*, *Probe*, and *R2L*).

IV. EXPERIMENTAL RESULTS

For detecting the attacks, training and testing are performed in each trial. In the training phase, three classifiers, fuzzy *k*-NN classifier, backpropagation neural network classifier, and naive bayes classifier, are constructed using the training data. The testing data are then fed into each trained classifier to identify normal behavior and intrusions in the testing phase. For fuzzy *k*-NN classifier, three nearest neighbors are selected for each testing connection. Within each neural network, the number of hidden neurons is decided by the number of input features, which is equal to the square root of number of input features multiply by two. We evaluate the performances of intrusion detection tasks by using standard measurements such as detection rate (*DR*), false positive rate (*FPR*), and classification rate (*CR*). To minimize the inaccuracy and variation factor of experiment results, 10 trials are performed in every detection task and then the average of those trials is recorded.

Table 1 shows the averaged *DR* and *FPR* performances of three classifiers in each group of the first layer, which classifiers 1, 2, and 3 represent fuzzy *k*-NN classifier, backpropagation neural network classifier, and naive bayes classifier, respectively. For groups 1, 2, and 3, the 9 basic features, the 13 content features, and the 19 traffic features are used, respectively. The results indicate that the fuzzy *k*-NN classifier using content feature set has the worst performance compared with those of other classifiers using partial feature set. It has both very low *FPR* (0.33%) and *DR* (14.55%), which implies either normal connections or malicious attacks are classified into normal behavior. On the contrary, by using the same content feature set, naive bayes classifier has the best overall performance, which its *CR* reaches 88.11%. For the backpropagation neural network classifier using basic feature set, it has both high *FPR* (93.21%) and *DR* (94.16%), which

*T*ABLE I. THE PERFORMANCES OF THREE FEATURE SELECTING CLASSIFIERS IN THREE GROUPS

		Group 1			Group 2			Group 3		
		DR	*FPR*	*CR*	*DR*	*FPR*	*CR*	*DR*	*FPR*	*CR*
Layer 1	Classifier 1	86.59	16.23	86.03	14.55	0.33	31.57	76.21	6.33	79.70
	Classifier 2	94.16	93.21	76.69	85.98	13.72	86.04	83.49	10.28	84.73
	Classifier 3	63.53	3.35	70.16	88.50	13.44	88.11	65.47	1.07	72.16

TABLE II. THE PERFORMANCES OF COMBINERS OF LAYERS 2 AND 3 USING DIFFERENT COMBINATION METHODS

		Majority Voting			Average Rule			Dempster-Shafer			Bayesian		
		DR	*FPR*	*CR*	*DR*	*FPR*	*CR*	*DR*	*FPR*	*CR*	*DR*	*FPR*	*CR*
Layer 2	Combiner 1	85.70	16.56	85.25	89.02	16.74	87.87	88.60	16.74	87.53	90.55	17.95	88.85
	Combiner 2	88.74	13.58	88.28	52.79	7.16	60.80	15.14	0.37	32.04	90.12	13.81	89.33
	Combiner 3	80.35	5.44	83.19	80.00	4.74	83.05	77.17	4.93	80.75	86.74	11.53	87.09
Layer 3	Final Result	87.21	5.26	88.72	85.03	2.19	87.59	83.49	1.91	86.41	93.35	9.63	92.75

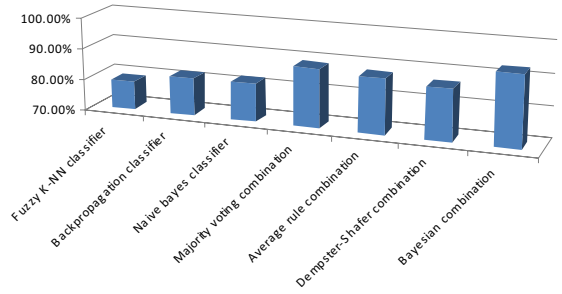

Figure 2. *CR*s of Three Classifiers Using Full Feature Set and Approaches Using Four Different Combination Methods

TABLE III. THE PERFORMANCES OF THREE CLASSIFIERS USING FULL FEATURE SET

	DR	*FPR*	*CR*
Classifier 1	74.59	3.26	79.02
Classifier 2	79.16	6.88	81.95
Classifier 3	78.22	1.81	82.21

TABLE IV. THE DETECTION RATES ON FOUR ATTACK GROUPS OF FINAL RESULT USING BAYESIAN COMBINATION METHOD

Attack	*DR*
DoS	98.51
Probe	99.91
U2R	98.84
R2L	76.14

represents most of the connections are classified into attack group. In general, the performances of naïve bayes classifier using three diverse partial feature sets are equally well compared with those of the other two classifiers.

Table 2 shows the performances of combiners on layers 2 and 3 using different combination methods. The results show that all of the four fusion techniques improve the overall performances in *FPR*s, *DR*s, and *CR*s compared with those of individual classifiers using partial feature sets shown in Table 1. For evaluating the performance of the proposed ensemble model, the experiments of three classifiers using the entire 41 features are also done and the results are demonstrated in Table III. Fig. 2 shows a comparison of three classifiers using full feature set and approaches using four different combination methods. The result indicates that all of the three classifiers using full feature set have equivalent *CR*s. All of their *FPR*s are below 7% and all the *DR*s do not reach to 80%. It also shows all of the four combination methods outperform the three classifiers using full feature set. Especially, the Bayesian combination method achieves the best outcome, which *FPR*, *DR*, and *CR* are 9.63%, 93.35% and 92.75%, respectively. Consequently, we further analyze its detection accuracies of four attack groups and Table 4 shows the result. From the values we observe, the ensemble approach using the Bayesian combination method performs well in detecting *DoS*, *Probe*, and *U2R* attacks that each one has over 98.5% *DR* and a relative low 76.14% *DR* in *R2L* attacks.

V. CONCLUSIONS AND FUTURE WORK

In this paper, the ensemble-classifier technique is applied to the intrusion detection task. We develop a three-layer hierarchy structure that includes three groups of classifiers and each consists of three base feature selecting classifiers. In each base feature selecting classifier, we apply different machine learning algorithm and feature subset to solve uncertainty problem and maximize the diversity. During the experiments, we only include a very small amount of network traffic to simulate uncertainty caused by limited information. Also, we compare the performances of different combination methods in fusing the outputs derived from the first and second layers of proposed model. The experimental results demonstrate that this hierarchy structure obtain a better detection performance than that of a single classifier using either partial feature subset or full feature set. The result also shows that the Bayesian combination method achieves the best detection accuracy among those four diverse combination techniques. In the future, we will continue the research of further improving detection performance of both normal and malicious activities, especially in promoting the detection accuracy in *R2L* attacks.

REFERENCES

[1] 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.

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

[3] 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.

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

[5] 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.

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

[7] 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.