Network Intrusion Detection Based on Neuro-Fuzzy Classification

Adel Nadjaran Toosi
Communication and Computer Research Laboratory, Faculty of Engineering, Ferdowsi University of Mashhad
ad_na85@stu-mail.um.ac.ir

Mohsen Kahani
Computer Department, Faculty of Engineering, Ferdowsi University of Mashhad
kahani@um.ac.ir

Reza Monsefi
Computer department, Faculty of Engineering, Ferdowsi University of Mashhad
rmonsefi@um.ac.ir

***Abstract*-With rapid growth of computer networks during the past few years, network security has become a crucial issue. Among the various network security measures, intrusion detection systems (IDS) play a vital role to integrity, confidentiality and availability of resources. It seems that the presence of uncertainty and the imprecise nature of the intrusions make fuzzy systems suitable for such systems. Fuzzy systems are not normally adaptive and have not the ability to construct models solely based on the target system's sample data. One of the successful approaches which are incorporated fuzzy systems with adaptation and learning capabilities is the neural fuzzy method. The main objective of this work is to utilize ANFIS (Adaptive Neuro Fuzzy Inference System) as a classifier to detect intrusions in computer networks. This paper evaluates performance of ANFIS in the forms of binary and multi-classifier to categorize activities of a system into normal and suspicious or intrusive activities. Experiments for evaluation of the classifiers were performed with the KDD Cup 99 intrusion detection dataset. The Overall Results show that ANFIS can be effective in detecting various intrusions.**

***Keyword*-Intrusion Detection, KDD dataset, Computer network Security, ANFIS, Neuro-Fuzzy classifier.**

I. INTRODUCTION

During the past few years, the numbers of intrusions in computer networks have grown extensively, and many new hacking tools and intrusive methods have appeared. Using an intrusion detection system (IDS) is one way of dealing with suspicious activities within a network [1].

Soft computing and machine learning approaches have demonstrated their abilities in IDS, and there are continual interests in utilizing them in such systems [1] [2] [3] [4]. Fuzzy logic as a robust artificial intelligent method has been successfully employed for many IDSs [2] [3] [5] [6] [7].

Most Fuzzy systems make use of human expert knowledge to create their fuzzy rule base and hence, lack adaptation. However, elicitation of fuzzy rules from experts is usually difficult. Therefore, building fuzzy systems with learning and adaptation capabilities has recently gained much attention. Various methods have been suggested for automatic generation and adjustment of fuzzy rules without using aid of human experts; the neural fuzzy [8] [9] and genetic fuzzy [10] [11] are two most successful approaches in this regard.

From the classification point of view, the main work of building an IDS is to build a classifier which can categorize normal and intrusive event data from the original dataset. ANFIS can incorporate human expertise as well as adopt itself through repeated training. This ability among others qualifies ANFIS as a fuzzy classifier for IDS [8].

In order to promote the comparison of different works in IDS area, the Lincoln Laboratory at MIT, under the Defense Advanced Research Project Agency (DARPA) and Air Force Research Laboratory (AFRL/SNHS) sponsorship, constructed and distributed the first standard dataset for evaluation of computer network IDS [12].

The fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining collected and generated TCP dump data provided by the aforementioned DARPA in the form of train-and-test sets of features defined for the connection records (a connection is a sequence of TCP packets starting and ending at some well-defined times) which we name them as KDD cup 99 dataset [13] and will use it for our experiments.

Intrusion detection as a classifier mainly consists of two processes; training the classifier from a training dataset and using this classifier to categorize a test dataset. Hereby, we made use of neuro-fuzzy classifiers to detect intrusions in computer networks based on KDD cup 99 datasets.

The subsequent parts of this paper are organized as follows: Section 2 describes KDD Cup 99 dataset on which our experiments are conducted. Then the next section briefly outlines the basics of neuro-fuzzy concepts in general and ANFIS particularly. The proposed system and experimental results are discussed in sections 3 and 4, respectively. Finally, section 6 makes some concluding remarks and recommends areas for future research.

II. KDD CUP 99 DATA SET

The KDD cup 99 dataset includes a set of 41 features derived for each connection and a label which specifies the status of connection records as either normal or specific attack type. These features had all forms of continuous, discrete, and symbolic, with significantly varying ranges and fall in four categories [13]:

- The *intrinsic* features of a connection, which includes the basic features of individual TCP connections. For example, duration of the connection, the type of the protocol (tcp, udp, etc), network service (http, telnet, etc), etc.
- The *content* feature within a connection suggested by domain knowledge is used to assess the payload of the original TCP packets, such as number of failed login attempts.
- The *same host* features examine established connections in the past two seconds that have the same destination host as the current connection, and calculate statistics related to the protocol behavior, service, etc.

This work was partially supported by the Iran Telecommunication Research Center (ITRC).

- The *similar same service* features which examine the connections in the past two seconds that have the same service as the current connection.

The data set encompasses different attack types, grouped into one of the four categories [13]:

- Probe: Host and port scans as precursors to other attacks. An attacker scans a network to gather information or find known vulnerabilities, e.g., port sweep.
- Dos (Denial of Service): Making some computing or memory resources too busy so that they deny legitimate users access to these resources, e.g., smurf.
- R2L (Root to Local): Unauthorized access from a remote machine according to exploit machine's vulnerabilities, e.g., imap.
- U2R (User to Root): Unauthorized access to local super user (root) privileges using system's susceptibility, e.g., buffer overflow.

Total number of connection records in training data set is about half a million records. This is far too large for our purposes; as such, only a subset of 10% data was employed here. The distribution of normal and attack types of connection records in this subset have been summarized in Table I.

The test data enjoys a different distribution; moreover, the test data includes additional attack types not present in the training data. This property of test makes classifying more challenging. Table II summarizes the distribution of normal and attack types of connection records in this test dataset.

TABLE I
DISTRIBUTION OF SAMPLES ON THE SUBSET OF 10% DATA OF KDD CUP 99

Class	number of Samples	Samples Percent
Normal	97277	19.69%
Probe	4107	0.83%
Dos	391458	79.24%
U2R	52	0.01%
R2L	1126	0.23%
Total	492021	100%

TABLE II
DISTRIBUTION OF SAMPLES ON THE TEST DATASET OF KDD CUP 99

Class	Number of Samples	Samples Percent
Normal	60593	19.48%
Probe	4166	1.34%
Dos	229853	73.90%
U2R	228	0.07%
R2L	16189	5.20%
Total	311021	100%

III. ADAPTIVE NEURO-FUZZY INFERENCE SYSTEMS

The past few years have witnessed a rapid growth in the number and variety of applications of fuzzy logic. Among various combinations of methodologies in soft computing, the one that has the highest visibility at this time is that of fuzzy logic and neurocomputing, leading to the so-called neuro-fuzzy systems. An effective method developed by Jang for this purpose is called ANFIS (Adaptive Neuro-Fuzzy Inference System) [8].

The rest of this section discusses the ANFIS structure as a class of adaptive network that is functionally equivalent to the Sugeno Fuzzy Inference Systems. There are some modeling situations in which one cannot just look at the data and distinguish what the Membership Functions (MFs) should look like. Rather than choosing the parameters associated with a given MF arbitrarily, these parameters could be chosen such that they tailor the MFs to the input/output data in order to account for these types of variations in the data values. This is where the so-called neuro-adaptive learning technique incorporated into ANFIS can help.

Assume a fuzzy inference system with two inputs x, y and one output z with the first order of Sugeno Fuzzy Model. Fuzzy rule set with two fuzzy if-then rules are as follows:

If x is A1 and y is B1, then f1=p1x+q1y+r1.
If x is A2 and y is B2, then f2=p2x+q2y+r2.

Figure 1 (a) illustrates the reasoning mechanism for this Sugeno Model. As it is shown in Figure 1(b), we can implement the reasoning mechanism into a feed forward neural network with supervised learning capability, which is known as ANFIS architecture.

The square and circle nodes are for adaptive nodes with parameters and fixed nodes without parameters, respectively. The first layer consists of square nodes that perform fuzzification with chosen MF. The parameters in this layer are called premise parameters. In the second layer, the t-norm operation is performed to produce firing strength of each rule. The ratio of i^{th} rule firing strength to the sum of all rules' firing strength is calculated in the third layer, generating the normalized firing strengths. The fourth layer consists of square nodes that perform multiplication of normalized firing strength with the corresponding rule. The parameters in this layer are called consequent parameters. The overall output is calculated by the sum of all incoming signals in the fifth layer [8].

ANFIS provides a method for the fuzzy modeling procedure to learn information about a dataset, in order to compute the MF parameters that best allow the associated fuzzy inference system to track the given input/output data. This learning method works similarly to that of neural networks.

The parameters associated with the MFs will change through the learning process. ANFIS uses either back propagation or a combination of least square estimations and back propagation for MF parameter estimations. The readers are referred to [8] for more details on these methods.

Before we start the ANFIS training, we need to generate our Fuzzy Inference System (FIS). FIS generation can implement in grid partitioning or subtractive clustering. In grid portioning, all the possible rules are generated based on the number of MFs for each input. For example in a two dimensional input space, with three MFs in the input sets, the number of rules in grid partitioning will result in 9 rules. This partitioning strategy needs only a small number of MFs for each input and it encounters problems when we have moderately large number of inputs. So subtractive clustering has been used to determine the number of rules, the MFs and their initial points [14]. It is an extension of the Mountain Clustering Method proposed by Yager [15]. The clusters' information obtained by this method is used for determining the initial number of rules and antecedent MFs.

An important advantage of using a clustering method to find rules is that the resultant rules are more tailored to the input data than they are in a FIS generated without clustering. In this study, we use subtractive clustering to determine the number of rules and antecedent MFs. Then ANFIS is applied for further fine-tuning of the MFs.

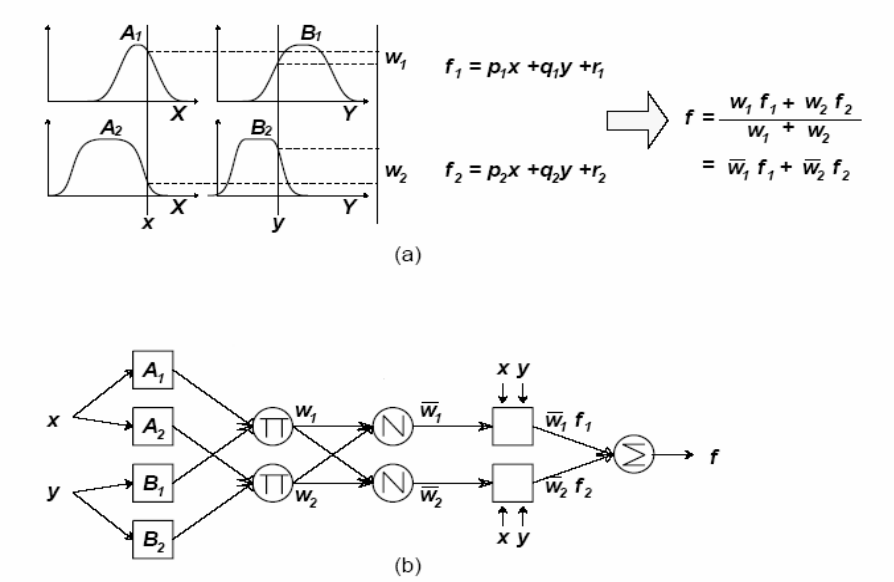

Fig. 1. (a) The Sugeno fuzzy model reasoning (b) Equivalent ANFIS structure [8]

IV. PROPOSED METHOD

This Section will elaborate methodology employed to training neuro-fuzzy classifiers. Classification basically consists of two processes, which are training the parameters of the classifier from a training dataset and using this classifier to categorize a test dataset.

A. Preprocessing

As it was mentioned earlier, the 41 features in the KDD cup 99 dataset had all forms of continuous, discrete, and symbolic, with significantly varying resolution and ranges [13]. Pattern classification methods are not able to process data in such format. Here, preprocessing involved mapping symbolic valued attributes to numeric ones. Symbolic features like *protocol_type*, *service* and *flag* were mapped to integer values ranging from 0 to N-1 where N is the number of symbols. For example *protocol_type* feature with three different symbols namely TCP, UDP, ICMP were appropriately mapped to three discrete numeric values 0, 1 and 2. All the other features were either discrete or continuous used as the original forms.

B. Classifiers Structure

We used all the above features as the inputs of our neuro-fuzzy classifiers. This paper utilizes ANFIS in two form of binary and multi-classifier to classify activities of a system into normal and suspicious or intrusive activity.

In binary classification, the classifier model has been trained with training dataset were labeled to one of the two classes, 0 for normal and 1 for attack. Unlike the binary classifier, attacks in training dataset for multi-classifier were mapped to one the five classes, 0 for normal, 1 for Probe, 2 for Dos, 3 for U2R, and 4 for R2L.

48840 randomly selected points form the subset of 10% of data is used as training dataset and 4884 records of data which does not overlap with training set dataset, selected as the checking data (used for validating the model) for both the binary and multi-classifiers.

The basic idea behind using a checking dataset for model validation is that after a certain point in training, the model begins overfitting the training dataset. If overfitting does occur, we cannot expect the classifier to respond well to other independent datasets. If checking data is used for ANFIS training, the final FIS that is associated with the minimum checking error will be chosen.

Table III summarizes the distribution of the samples in the Training and Checking dataset for the binary and the multi-classifier.

Afterward, subtractive clustering method with neighborhood radius equal to 0.5 has been chosen to partition the training data to generate an FIS structure. All the input MFs are the Gaussian function which are specified by four parameters. Then for further fine-tuning and adaptation of the MFs, training dataset was used for training ANFIS, while the checking dataset was used for validating the model identified. The ANFIS architecture used is equivalent to Sugeno Fuzzy Model and is the same structure as the one that has been explained in section 3.

The ANFIS used for the binary classification contains 380 nodes and a total number of 496 fitting parameters, of which 328 are premise parameters and 168 are consequent parameters. The average RMSE (Root Mean Squared Error) for the training and checking dataset of binary classifier after 50 epochs of learning is 2349.74 and 0.207448, respectively. ANFIS used for multi classification hold 328 premise parameters and 168 consequent parameters and the RMSE for the training and checking dataset after 50 epochs of learning is 10164.7 and 0.264419, respectively, too. It is unusual to observe that RMSE for the training data is larger than the checking data during the training process, as is the case here. It seems that it is the result of many differences between the number of training and checking samples and also the loss of training epochs. As it has been previously mentioned in section 3, ANFIS structure has one output. In this paper, the ANFIS output specifies the class number of the 41 input featured vector. The output of each ANFIS is not necessarily an integer as the class number. For this reason, we need to gain an approximate class number by rounding off the given number. Γ is the parameter for rounding off, which gives us the integer value. Output will be rounded based on Γ, which gives us an integer value. If it is equivalent to one of the attack's class numbers, then the current connection record is

classified as an attack, otherwise the related connection will be assumed as normal record. Below, we will investigate the effect of Γ on the performance evaluation parameters.

TABLE III
DISTRIBUTION OF SAMPLES ON THE TRAINING AND CHECKING DATA RANDOMLY SELECTED OF 10% DATA OF KDD CUP 99 DATASET FOR THE CLASSIFIERS

Class	Binary Classifier		Multi-Classifier	
	Training	Checking	Training	Checking
Normal	25000	2500	25000	2500
Dos	20000	2000	20000	2000
U2R	40	4	40	4
R2L	800	80	800	80
PROBE	3000	300	3000	300
	48840	4884	48840	4884

C. Performance Comparison Measures

Standard metrics that were developed for evaluating network intrusion detections are detection rate and false alarm rate. Detection rate is computed as the ratio of the number of correctly detected attacks to the total number of attacks, while false alarm (false positive) rate is computed as the ratio of the number of normal connections (that is incorrectly misclassified as attacks) to the total number of normal connections.

Table IV shows detection rate and false alarm rate based on binary and multi-classifier for training and checking data after 50 epoch of training with Γ = 0.5.

TABLE IV
FALSE ALARM RATE AND DETECTION RATE S FOR THE TRAINING AND CHECKING DATA

Classifier	Data	False Alarm Rate%	Detection Rate%
Binary Classifier	Training	0.17	96.50
	Checking	0.08	96.64
Multi Classifier	Training	5.63	98.43
	Checking	3.64	99.41

V. RESULTS

Two different classifiers have been used in this work. TABLE V shows the notation used here after.

TABLE V
ABBREVIATIONS USED FOR OUR APPROACHES

Abbreviation	Approach
B-NFC	Binary Neuro-Fuzzy Classifier
M-NFC	Multi Class Neuro-Fuzzy Classifier

Two different types of experiments have been performed here; in the first experiment all the records of labeled test dataset (known as corrected label dataset) were used as the testing data to evaluate our classifiers. The results are shown in Table VI. As it can be seen, our classifiers has a good performance at intrusion detections with approximately low false alarm rate, although only we have only used about fifty thousand records of 10% of the dataset. It is important to mention that unlike the methods of comparison in the quoted references, here, our test data in the current experiment contains new attacks which were not present in the training dataset. In the second experiment, we randomly selected 40000 sampled connections from the source of the training dataset. To reduce the effects due to random sampling, five trails, that does not overlap neither with training set nor each others, have been carried out and average of the resulting value over 5 trials have been computed.

TABLE VI
FALSE ALARM RATE AND DETECTION RATE S FOR THE TEST DATA OF THE FIRST EXPRIMENT

Classifier	False Alarm Rate (%)	Detection Rate (%)
B-NFC	0.3	89.43
M-NFC	3.4	91.14

We compare our classifiers with two different fuzzy algorithms' performances proposed in [5] and [2]. Also different algorithms' performance exploit at the above papers has been referenced here again [16] [17].

Our classifier demonstrates better performance in reducing false alarm rate and increasing detection rate. Based on the results shown in the TABLE VII, it can be easily seen that our approach has an overall better performance than the other methods.

In the rest of this section, we have applied the Receiver Operating Characteristic (ROC) analysis to evaluate the performance of our classifiers with respect to parameter Γ. To generate the ROC curve, we changed Γ to a value between 0 and 0.5 and plotted coordinate point $(FA, DR)_{\Gamma}$, where FA is the false alarm rate and DR is the detection rate [2]. Figure 2 displays ROC curve for the classifiers which has been used here with respect to Γ.

The ROC curve can be used to determine when a classifier has good performance. For each curve, the point at the upper left corner represents the optimal detection with high detection rate and low false alarm rate [4]. If the ROC curve of a classifier 'A' dominates the classifier 'B' then classifier 'A' is better than classifier 'B'. The ROC curve for the proposed neuro-fuzzy classifiers shows how the parameter Γ affects the false alarm rate and detection rate.

VI. CONCLUSION

In this paper, we have applied ANFIS as a neuro-fuzzy classifier to the intrusion detection methods. Subtractive clustering determines the number of rules and MFs with their initial locations. The method used here is capable of producing fuzzy rules without the aid of human experts. Results of experiments show that these fuzzy rules are effective for detecting intrusion in a computer network. Also results illustrate the suggested that ANFIS is more appropriate as a binary classifier rather than a multi-classifier one.

Our future work will focus on reducing fuzzy input variables by methods of feature selection.

VII. ACKNOWLEDGMENT

The first author would like to thank FUM Communication and Computer Research Laboratory for their in-kind support and encouragement during this research. He also wishes to express his appreciations to the helpful suggestions and comments of his colleagues E. Bagheri, M. Amini.

TABLE VII
FALSE ALARM RATE, DETECTION RATE AND COMPLEXITY OF DIFFRENET APPROACHES

Classifier	False Alarm Rate (%)	Detection Rate (%)	Complexity
B-NFC	0.13	99.60	O(n)
M-NFC	4.61	99.90	O(n)
SRPP [5]	3.58	99.08	O(n)
EFRID [2]	7	98.96	O(n)
RIPPER [16]	2.02	94.26	$O(n \times \log^2 n)$
SMARTSIFTER[17]	-	82	$O(n^2)$

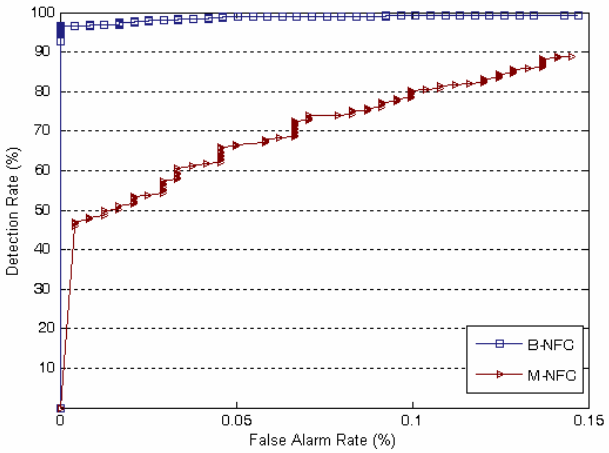

Fig. 2. ROC curve for the Neuro-Fuzzy classifiers; $0 \leq \Gamma \leq 0.5$

REFERENCES

[1] D. Song, M.I. Heywood, A.N. Zincir-Heywood, "Training Genetic Programming on Half a Million Patterns: An Example from Anomaly Detection," *IEEE Transactions on Evolutionary Computation*, 2005.

[2] J. Gomez, D. Dasgupta, "Evolving Fuzzy Classifiers for Intrusion Detection," *Proceeding Of 2002 IEEE Workshop on Information Assurance, United States Military Academy*, West Point NY, June 2001.

[3] S. Chavan, K. Shah, N. Dave, S. Mukherjee, A. Abraham and S. Sanyal, "Adaptive Neuro-Fuzzy Intrusion Detection System," *IEEE International Conference on Information Technology: Coding and Computing (ITCC' 04), USA, IEEE Computer Society,* Vol. 1, pp. 70-74, 2004.

[4] Z. Zhang, J. Li, C. Manikopoulos, J. Jorgenson and J. Ucles, "HIDE: a Hierarchical Network Intrusion Detection System Using Statistical Preprocessing and Neural Network Classification," *Proceedings of the 2nd Annual IEEE Systems, Mans, Cybernetics Information Assurance Workshop,* West Point, NY, 2001.

[5] M. S. Abade, J. Habibi, C. Lucas, "Intrusion detection using a fuzzy genetics-based learning algorithm," *Journal of Network and Computer Applications,* August 2005.

[6] J. E. Dickerson, J. E. Dickerson, "Fuzzy network profiling for intrusion detection," *Proceedings of NAFIPS 19th International Conference of the North American Fuzzy Information Processing Society*, pp. 301-306, Atlanta, USA, July 2000.

[7] M. Gao, M. C. Zhou, "Fuzzy intrusion detection based on fuzzy reasoning Petri nets," *Proceeding of the 2003 IEEE International Conference on Systems, Man and Cybernetics*, 5-8, pp. 1272-1277, Washington D. C., Oct. 2003.

[8] J.-S. R. Jang, "ANFIS: Adaptive-Network-based Fuzzy Inference Systems," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 3, pp. 665-685, May 1993.

[9] D. Nauck, R. Kruse, "NEFCLASS - A Neuro-Fuzzy approach for the classification of data," *presented at the Symposium on applied Computing*, Nashville, USA, 1995.

[10] H. Ishibuchi, T. Nakashima, T. Murata, "A fuzzy classifier system that generates fuzzy if–then rules for pattern classification problems," *Proceedings of second IEEE international conference on evolutionary computation*, Perth, Australia, November, pp. 759–64, 1995.

[11] J. Liu, J. Kwok, "An extended genetic rule induction algorithm," *Proceedings of the Congress on Evolutionary Computation Conference*, 2000.

[12] DARPA Intrusion Detection Evaluation: *http://www.ll.mit.edu/SSt/ideval/result/result_index.html*.

[13] KDD Cup 1999 Intrusion detection dataset: *http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html*.

[14] S. Chiu, "Fuzzy Model Identification Based on Cluster Estimation," *Journal of Intelligent & Fuzzy Systems*, Vol. 2, No. 3, September. 1994.

[15] R. Yager, D. Filev, "Generation of Fuzzy Rules by Mountain Clustering," *Journal of Intelligent & Fuzzy Systems*, Vol. 2, No. 3, pp. 209-219, 1994.

[16] W. Fan, M. Miller, SJ. Stolfo, Chan PK. "Using artificial anomalies to detect unknown and know network intrusions," *Proceedings of the first IEEE international conference on data mining*, 2001.

[17] K. Yamanishi, J. Takeuchi, G. Williams, "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms", *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pp. 320-324, 2000.