# A Neuro-Fuzzy Classifier for Intrusion Detection Systems

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

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

## ABSTRACT

Computer networks have experienced an explosive growth over the past few years and have become the targets for hackers and intruders. An intrusion detection system's main goal is to classify activities of a system into two major categories: normal activity and suspicious or intrusive activity. The objective of this paper is to expose ANFIS as a neuro-fuzzy classifier to detect intrusions in computer networks. Our experiments and evaluations were performed with the KDD Cup 99 intrusion detection dataset which is a version of the 1998 DARPA intrusion detection evaluation dataset prepared and managed by MIT Lincoln Laboratories. This paper shows that our proposed method can be effective in detecting various intrusions.

## Keywords

ANFIS, Intrusion Detection, Neuro-Fuzzy, Fuzzy, Subtractive Clustering.

## 1. INTRODUCTION

During the past few years, the number of intrusions in computer networks has 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 [12].

Soft computing approaches have demonstrated their abilities in intrusion detection systems, and there are continual interests in utilizing them in intrusion detection systems [7,11,12,14]. Fuzzy logic as a robust artificial intelligent method has been successfully used for many intrusion detection systems [1,4,6,7,11].

Most Fuzzy systems use human experts to create sets of fuzzy rules as stated in [4], “We assume that security administrator can use their ***expert knowledge*** to help create a set of rules for each attack.” and [6] “The rules are described as follows, which are derived from ***experiments*** in detection TCP SYN flooding attack and the surveying of many hacking reports”. However, elicitation of fuzzy rules from experts is usually difficult. Moreover, the traditional fuzzy systems are not adaptive. Therefore, building fuzzy systems with learning and adaptation capabilities has gained much interest recently. Various methods have been suggested for automatic generation and adjustment of fuzzy rules without using the aid of human experts; the neural fuzzy [8,9] and genetic fuzzy are two most successful approaches in this regard [1].

ANFIS as an adaptive neuro-fuzzy inference system has the ability to construct models solely based on the target system sample data. This ability among others qualifies ANFIS as a fuzzy classifier for intrusion detection.

From the view point of classification, the main work of building an intrusion detection system is to build a classifier which can categorize normal and intrusion event data from the original dataset.

In order to promote the comparison of different works in this 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 intrusion detection systems [3].

The Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining collected and generated TCP dump data provided by the DARPA explained earlier 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 and will use it for our experiments [10].

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

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 fuzzy inference systems and neuro-fuzzy concepts in general and ANFIS (Adaptive Neuro-Fuzzy Inference System) particularly. Section 4 describes the subtractive clustering technique employed by ANFIS for automatic generation of the fuzzy inference system structure. Then the proposed system is explained and experimental results and evaluation of our approach is discussed. Finally, sections 7 makes some concluding remarks and proposed further areas for future research.

## 2. KDD CUP 99 DATASET

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 fall in four categories:

- 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.

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

Likewise, attacks fall into four main categories [10]:

- DOS (Denial of service): making some computing or memory resources too busy so that they deny legitimate users access to these resources.

- R2L (Root to local): unauthorized access from a remote machine according to exploit machine's vulnerabilities.

- U2R (User to root): unauthorized access to local super user (root) privileges using system's susceptibility.

- PROBE: host and port scans as precursors to other attacks. An attacker scans a network to gather information or find known vulnerabilities.

Total number of connection records in training dataset are about half million records. This is too large for our ends; 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 1.

**Table 1. Distribution of samples on the subset of 10% data of KDD Cup 99 dataset**

| Class | Samples | Samples Percent |
|---|---|---|
| Normal | 97277 | 19.69% |
| Dos | 391458 | 79.24% |
| U2R | 52 | 0.01% |
| R2L | 1126 | 0.23% |
| PROBE | 4107 | 0.83% |
|  | 492021 | 100% |

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 2 summarizes the distribution of normal and attack types of connection records in this test dataset.

**Table 2. Distribution of samples on the test data with corrected labels of KDD Cup 99 dataset**

| Class | Samples | Samples Percent |
|---|---|---|
| Normal | 60593 | 19.48% |
| Attack | 250436 | 80.52% |
|  | 311029 | 100% |

It is important to mention that we are only interested in differentiation between normal and intrusion behavior in this work, and multi-class classification and detection of intrusion type is the subject of future researches.

## 3. FUZZY LOGIC AND NEURO-FUZZY

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 so-called neuro-fuzzy systems. An effective method developed by Jang, et. al. for this purpose is called ANFIS (Adaptive Neuro-Fuzzy Inference System) [8].

The basic structure of the fuzzy inference system that we have seen thus far is a model that maps the input characteristics to the input membership functions. Nowadays, three well known types of fuzzy inference system are employed in various systems. The **Mamdani Fuzzy Model** was proposed as the very first attempt to map an input space to an output space on top of the experience of experts. In effort to develop a systematic approach to generate fuzzy rules from a given input-output dataset Takagi, Sugeno, and Kang proposed **TSK Fuzzy Model** (known as the **Sugeno Fuzzy Model**).

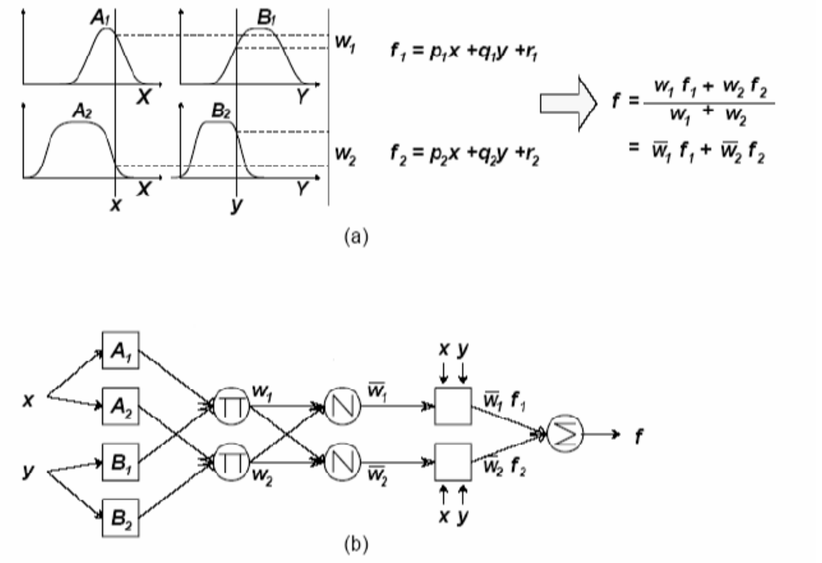

**Figure 1. (a) The Sugeno fuzzy model reasoning (b) Equivalent ANFIS structure [9]**

A fuzzy rule in a Sugeno fuzzy model has the form of,

*if* x *is* A *and* y *is* B *then* z=f(x, y)

where A and B are input fuzzy set in antecedent and usually z=f(x, y) is a zero or first order polynomial function in the consequent.

Fuzzy reasoning procedure for the first order Sugeno Fuzzy Model is shown in Figure 1.

Here, defuzzification procedure in the Mamdani Fuzzy Model replaces with operation of weighted average in order to avoid the time consuming procedure of defuzzification. (Defuzzification refers to the way a crisp value is extracted from a fuzzy set as a representative value)[9].

**Tsukamoto Fuzzy Model** replaces consequent function of Sugeno model with a monotonical membership function.

The ANFIS architecture used in this paper is equivalent to Sugeno Fuzzy Model. For more details on other fuzzy inference systems, the readers may refer to [8,9].

The rest of this section discusses the ANFIS structure as a class of adaptive network that is functionally equivalent with 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 should look like. Rather than choosing the parameters associated with a given membership function arbitrarily, these parameters could be chosen such that they tailor the membership functions 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+q1+r1.

*If* x *is* A2 *and* y *is* B2, *then* f2=p2x+q2+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 membership function. 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 membership function 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 membership functions will change through the learning process. ANFIS uses either back propagation or a combination of least square estimations and back propagation for membership function parameter estimations. The readers are referred to [8] for more details on these methods.

## 4. SUBTRACTIVE CLUSTERING

We use subtractive clustering to determine the number of rules, the membership functions and their initial points. Then ANFIS is applied for further fine-tuning of the membership functions.

Suppose we do not have a clear idea of how many clusters there should be for a given set of data. Subtractive Clustering [2] is a fast, one-pass algorithm for estimating the number of clusters and the cluster centers in a set of data. This method is used here, and it is an extension of the Mountain Clustering Method proposed by Yager [13].

Consider a collection of m data points $\{x_1 \dots x_m\}$ in an N-dimensional space. Subtractive clustering assumes each data point is a potential cluster center and calculates a measure of the potential for each data point based on the density of surrounding data points. Density measure at data point $x_j$ is calculated as follows:

$$D_j = \sum_{i=1}^{m} \exp(-\frac{|x_j - x_i|^2}{(r_a/2)^2})$$

Where $r_a$ is a positive constant and it defines the neighborhood radius. The algorithm selects the data point with the highest density measure as the first cluster center and then destroys the potential of data points near the first cluster center. The algorithm then selects the data point with the highest remaining potential (next highest density measure has been remained) as the next cluster center and destroys the potential of data points near this new cluster center. This process of acquiring a new cluster center and destroying the potential of surrounding points repeats until the potential of all data points fall below a threshold. The range of influence of a cluster center in each of the data dimensions is called cluster radius. A small cluster radius will lead to find many small clusters in the data (resulting in many rules) and vice versa.

The clusters' information obtained by this method is used for determining the initial number of rules and antecedent membership functions, which is used for identifying the Fuzzy Inference System (FIS).

In this study, we use Subtractive Clustering to determine the number of rules and antecedent membership functions. So we can obtain a FIS structure that contains a set of fuzzy rules to cover the feature space.

## 5. SYSTEM ARCHITECTURE

As pointed out in section 2, there are 41 features in KDD cup 99 dataset. We used all the above features as the inputs of our neuro-fuzzy classifiers.

These features had all forms continuous, discrete, and symbolic, with significantly varying resolution and ranges. Pattern classification methods are not able to process data in such format. So preprocessing was required before building classification models. Here, preprocessing involved mapping symbolic valued attributes to numeric ones. Symbolic features like protocol types, services and flags 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 - TCP, UDP, ICMP- were appropriately mapped to three discrete numeric values 0, 1, 2. All the other features were either discrete or continuous used as the original forms.

The patterns were also labeled to one of the two classes, 0 for normal and 1 for attack.

The 150000 randomly selected points of the subset of 10% of data is used as training. Randomly 40000 records of data selected from labeled test dataset as the checking data (used for validating model). The labeled test data exists in the KDD dataset with the description of corrected data.

Subtractive Clustering Method with $r_a$=0.5 (neighborhood radius) partitions the training data and generates an FIS structure. So two fuzzy rules and two Membership Functions (MFs) for each input were obtained. All the input MFs are the Gaussian function which are specified by four parameters. Then for further fine-tuning and adaptation of membership functions, training dataset was used for training ANFIS while the checking dataset was used for validating the model identified. The ANFIS architecture used in this paper is equivalent to Sugeno Fuzzy Model and is as the same as structure has been explained at section 3.

The basic idea behind using a checking dataset for model validation is that after a certain point in training, the model begins over fitting the training dataset. If over fitting does occur, we cannot expect the fuzzy inference system to respond well to other independent datasets.

The ANFIS used here contains 212 nodes and a total number of 284 fitting parameters, of which 164 are premise parameters and 84 are consequent parameters.

The average RMSE (Root Mean Squared Error) for the training and checking data after 50 epochs of learning is 0.0516 and 0.2836, receptively. Figure 2 displays the error measure (RMSE) as function of epoch number for training dataset. The difference between desired and predicted values for both training and checking data can be seen in Figures 3(a) and 3(b), respectively.

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, *0* for normal and *1* for attack.

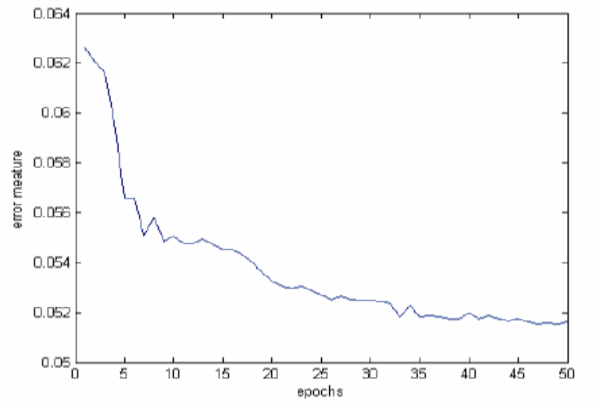

**Figure 2. Error measures vs. epoch numbers for the training dataset**

As it can be inferred from Figure 3, the output 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. Below, we will investigate the effect of Γ on performance evaluation parameters.

As mentioned earlier, training ANFIS causes further fine-tuning and adaptation of initial membership functions. Initial and final membership functions of some input features are illustrated in Figure 4.

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

Table 4 shows detection rate, false alarm, and classification rate for training and checking data after 50 epoch of training with Γ=0.5.

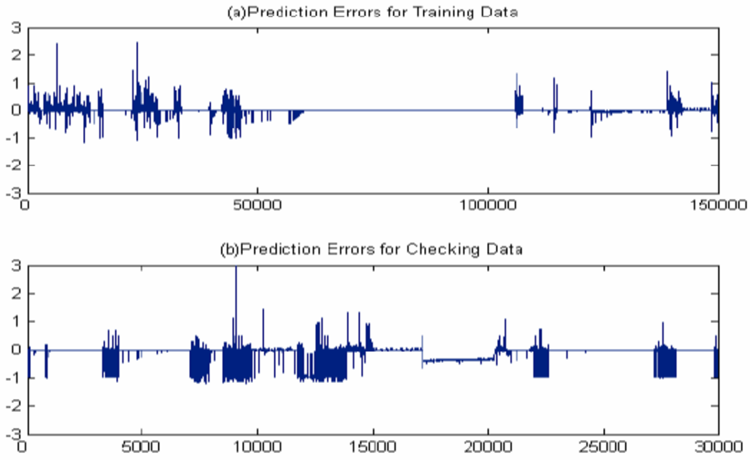

**Figure 3. The difference between desired and predicted values for (a) training data, (b) checking data.**

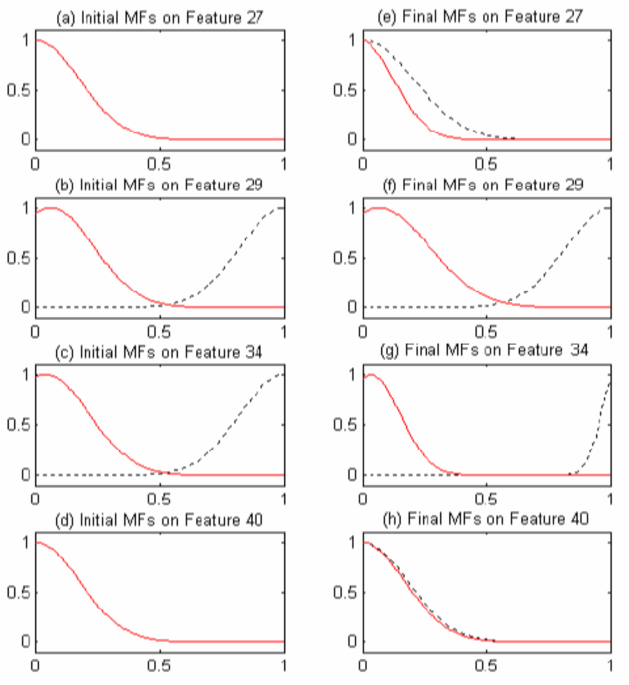

**Figure4. (a), (b), (c),(d) Membership Functions before training; (e), (f) ,(g) ,(h) Membership Functions after training**

Another metric used here is the *classification rate*. Classification rate is defined as ratio between number of test instances correctly classified and the total number of test instances classified.

**Table 3. False Alarm, Detection and classification rate for training and checking data, Γ=0.5**

| Data | False Alarm Rate% | Detection Rate% | Classification Rate% |
|---|---|---|---|
| Training | 0.61 | 99.75 | 99.68 |
| Checking | 1.6 | 91.00 | 92.44 |

## 6. RESULTS

Two different experiments have been done on this paper; first we used all the records of labeled test dataset (corrected) as the testing data to evaluate our classifiers. Results are shown in Table 4. As can be seen, our classifier has good performance at intrusion detections with approximately low false alarm *rate* although only we have used 150000 records of 10% percent dataset. It is important to mention that unlike the other methods references here for comparison, our test data in the first experiment contains novel attacks (novel attacks which were not present in the training dataset).

In the second experiment, we randomly selected 40000 sampled connections from the source of 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 the average of the resulting value over 5 trials have been computed.

We compare our classifiers with two different fuzzy algorithm performances proposed in [1] and [7]. Also different algorithms performance referred at the above papers have been referenced here again.

**Table 4. False Alarm, Detection and Classification Rate for test data of first experiment; Γ=0.5**

| Data | False Alarm Rate % | Detection Rate% | Classification Rate% |
|---|---|---|---|
| Test | 1.6 | 91.07 | 92.48 |

Table 5 shows the comparison of different algorithms performance. Our classifier demonstrates better performance in reducing false alarm rate and increasing detection rate. Based on the results shown in the table, it can be seen easily that our approach has overall better performance than the other methods.

**Table 5. Comparing False Alarm, Detection and complexity of different algorithms**

| Algorithm | False Alarm Rate% | Detection Rate% | Complexity |
|---|---|---|---|
| Neuro-Fuzzy Classifier | 0.59 | 99.54 | O(n) |
| SRPP [1] | 3.58 | 99.08 | O(n) |
| EFRID [7] | 7 | 98.96 | O(n) |
| RIPPER[5] | 2.02 | 94.26 | $O(n \times \log^2 n)$ |

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 between 0 and 0.2 and plotted coordinate point $(FA, DR)_\Gamma$, where FA is the false alarm rate and DR is the detection rate[7].Figure 5 displays ROC curve for the classifier which has been used here with respect to Γ.

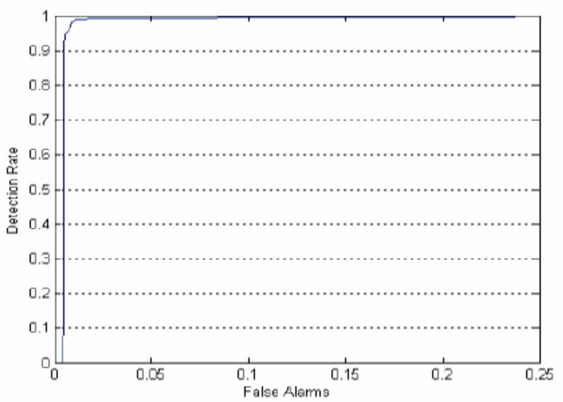

**Figure 5. ROC curve for Neuro-Fuzzy classifier; 0 ≤ Γ ≤ 0.2**

The ROC curve shows how the parameter Γ affects the false alarm rate and detection rate.

## 7. CONCLUTIONS

In this paper, we applied ANFIS as a neuro-fuzzy classifier for intrusion detection. Subtractive clustering determines the number of rules and membership functions with their initial locations. The method used here is capable of producing fuzzy rules without the aid of human experts. Results of experiments show these fuzzy rules are effective for detecting intrusion in a computer network. Also results illustrate the suggested method is capable of detecting novel attacks, and it makes this suitable for anomaly intrusion detection systems.

Our future work will focus on multi-class classification and detection of intrusion type. Also, we will continue to study on reducing fuzzy input variables by methods of feature selection.

## 8. ACKNOWLEDGMENTS

This work was partially supported by the Iran Telecommunication Research Center (ITRC), which is hereby acknowledged.

The first author also wishes to thank FUM Communication and Computer Research Lab. 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 and M. Khoshsaligheh.

## 9. REFERENCES

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

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

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

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

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

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

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

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

[9] Jang, J. S. R. and C. T. Sun, "Neuro-fuzzy modeling and control," Proceedings of the IEEE, March. pp. 378-406, March 1995.

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

[11] Shah K., Dave N., Chavan S., Mukherjee S., Abraham A. and Sanyal S., "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.

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

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

[14] Zhang Z., Li J., Manikopoulos C., Jorgenson J. and Ucles J., "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.