International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:1, 2009

Anomaly Detection using Neuro Fuzzy system

Fatemeh Amiri, Caro Lucas and Nasser Yazdani

***Abstract*—** As the network based technologies become omnipresent, demands to secure networks/systems against threat increase. One of the effective ways to achieve higher security is through the use of intrusion detection systems (IDS), which are a software tool to detect anomalous in the computer or network. In this paper, an IDS has been developed using an improved machine learning based algorithm, Locally Linear Neuro Fuzzy Model (LLNF) for classification whereas this model is originally used for system identification. A key technical challenge in IDS and LLNF learning is the curse of high dimensionality. Therefore a feature selection phase is proposed which is applicable to any IDS. While investigating the use of three feature selection algorithms, in this model, it is shown that adding feature selection phase reduces computational complexity of our model. Feature selection algorithms require the use of a feature goodness measure. The use of both a linear and a non-linear measure - linear correlation coefficient and mutual information- is investigated respectively

***Keywords*—**anomaly Detection, feature selection, Locally Linear Neuro Fuzzy (LLNF), Mutual Information (MI), liner correlation coefficient.

I. Introduction

With expansion of Networks and their applications, types and number of the attacks have been increased dramatically. Despite all efforts to construct the first line of defense for computer security such as user authentication, data encryption message encryption, secured network protocols, and firewalls, intruder still can bypass them. An anti-virus software can protect network users from malware (viruses, Trojan, horses, worms and spyware) that signature of them exist in their system database. Signature files of many anti-virus products only are updated on a weekly or daily basis. Then, computers are unsafe against new intrusions in the interval between updates.

One way to fill these gaps in network security is use of an Intrusion Detection System (IDS). IDS detects misuse of network or computer resources. It gathers and analyzes information from various areas within a computer or a network (users, processes) in order to identify suspicious patterns that may indicate a network or system attack from someone attempting to break into a system, and alerts the system or network administrator.

IDSs have been classified into two categories: *signature-based detection system* and *anomaly detection system*. A signature-based system models attack patterns or behavior of intruder and will alert once a match is detected. This is similar to the way which anti-virus system detects malware. The issue is that the signature of threats must be first detected by the IDS, diagnosed, and then the signature for detecting that attack must be applied to IDS. During that lag time, the IDS would be unable to detect the new threat.

On the other hand, an anomaly detection system first creates a baseline profile of the normal behavior of the network. The system alert the network administrator when traffic is detected which is anomalous or significantly different than the baseline profile. This system has a capability to detect previously unknown attacks without the use of signatures.

Machine learning and data mining techniques have recently been successfully applied to various problems in intrusion detection. Most data mining methods for IDS are good at detecting particular types of malicious activity. In this paper, PLLNF, an anomaly detection system, is developed which uses Locally Linear Neuro Fuzzy model (LLNF) for modeling system and LOLIMOT which is a powerful construction algorithm with many advantage over neural network or fuzzy systems. Higher dimensional its problem restricts the performance of LOLIMOT. Therefore, a preprocessing phase –feature selection phase- has been added to improve PLLNF performance. Of course this can be generalized to use with any IDS. Feature selection has reduced computational complexity of LLNF model.

Feature selection and ranking is an important issue in IDS because of a great number of features extracted from raw network data. The purpose of feature selection is to identify features which are truly useful, those are less important and those may be useless. Feature selection is useful to increase accuracy of learning algorithm, facilitate data understanding and data visualization and improve the generalization. Researchers have used many feature selection techniques in developing an IDS such as the Markov Blanket model [1], decision tree analysis [1], flexible neural tree model [2], hidden Markov model [3] are explained in Section 2.

Feature selection algorithms need a feature goodness measure. In this paper, two feature goodness measures are used: *correlation coefficient* and *Mutual Information* (MI). With these goodness measures, many feature selection algorithm are proposed from which three of those studied in this work: *liner correlation based feature selection* (LCFS), *forward feature selection Algorithm* (FFSA), and *mutual information based feature selection* (MIFS).

F. Amiri is with Center of Excellence: Control and Intelligent Processing, Faculty of Electrical and Computer Engineering, University of Tehran, Tehran, Iran (Corresponding author to provide phone: +98- 21- 80294195; fax: +98-21-82094214; e-mail: f.amiri@ ut.ac.ir).

C.Lucas with Center of Excellence: Control and Intelligent Processing, Faculty of Electrical and Computer Engineering, University of Tehran, Tehran, Iran (e-mail: lucas@ipm.ir).

N.Yazdani is Associate Professor, ECE Department, University of Tehran, Tehran, Iran. (email:yazdani@ut.ac.ir).

The first algorithm is based on liner correlation coefficient and others are based on MI. A common property of them is that they are the *filter feature selection methods* applied before learning phase and does not depend on the learning process. Linear correlation removes features with near liner correlation to the class. The techniques based on a linear correlation coefficient cannot take care of arbitrary relation between the pattern coordinates and the different classes. Contrarily, the MI based can measure arbitrary relations between features and does not depend on transformation acting on the different features [4]. FFSA gives good result when all features are independent and no redundancy takes place Otherwise MIFS is generally a proper method for feature selection which has maximum relevancy and minimum redundancy.

The rest of paper is organized as follows: In Section II, related works in the field are studied. Then MI and how it is estimated are described in Section III. FFSA, MIFS are described in Sections IV and Section V. LCFS is described in Section VI. In Section VII, LLNF is described. Intrusion Dataset is introduced in Section VIII. Experiment and result are stated in Section IX. In Section X and Section XI, come in discussion and conclusion.

II. Related Work

In past decades, a great number of intrusion detection methods have been proposed to detect anomalies. Intrusion Detection Expert System (IDES) [5] was one of earliest intrusion detection systems which developed at the Stanford Research Institute. This system continuously monitored user behavior and detected suspicious events as they occurred. MIND (Minnesota Intrusion Detection System) [6] is one of the known data mining projects in anomaly detection which assigns a degree of outlierness to each data point called local outlier factor (LOF). The advantage of the LOF algorithm is capability to detect all forms of outliers.

Some techniques that widely applied for intrusion detection are statistic [7], artificial neural network [8],[9], genetic Algorithm and fuzzy logic [10], outlier detection schema [7] and rule learning [11].

Mukkamalaa et al [12] have examined the performance of Support Vector Machine, Multivariate Adaptive Regression Splines (MARS) and Artificial Neural Network (ANN). It has been demonstrated that an ensemble of ANN, MARS and SVM is preferable to individual approaches for intrusion detection according to classification accuracy. *Zhang et al*. [13] have formulated intrusion detection as a text processing problem which can be solved by SVM. Additionally, this system can employ some text processing techniques based on the characterization of frequency of system call executed by the privileged program.

Dickerson et al. [14] developed the Fuzzy Intrusion Recognition Engine (FIRE). FIRE creates and use fuzzy logic rules to the audit data to classify it as normal or anomalous. FIRE generates fuzzy set for every observed feature then applied them to define fuzzy rule. They understood this method is particularly effective against port scan and probes.

Saniee et al. [10] have applied fuzzy genetic-based learning method for intrusion detection problem. A new fitness function called SRPP is suggested in this paper. This fitness function increases the detection rate and increases the rate of false alarm as well.

Due to the curse of high dimensionality of network data, Several IDS which uses feature selection as a pre-processing have been developed. *Cherbrolu et al.* [1] discussed important and useful input features in building an IDS. Authors used Markov Blanket model and decision tree analysis in feature selection phase. Bayesian Network (BN) Classifier and Regression Trees (CART) have been applied to create an intrusion detection model.

Chena et al [2] have applied a flexible neural tree (FNT) model for IDS. The FNT model can reduce the number of features. Using 41 features, the best accuracy for the DoS and U2R is given by FNT model. The decision Tree classifier supplied the best accuracy for Normal and Probe classes which are a little better than the FNT classifiers.

Sung et al. [15] deleted one feature at a time to act experiment on SVM and Neural Network. KDD Cup 1999 dataset has been used for test this method. In five-class classification, it's found that by using only 19 of the most significant feature, rather than the all 41-feature set, the change in performance of intrusion detection was statistically unimportant.

III. Mutual Information and estimate it

In probability theory, especially in information theory, the mutual information is a natural measure of the dependency among random variables. MI between two random variables X and Y can be a measure of the amount of knowledge on Y supplied by X (or converse). If X and Y are independent, then, the MI between them will be zero.

The MI of two random variables X and Y is defined as:

$$I(X;Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$$
$$= H(X) + H(Y) - H(X;Y) \quad (1)$$

Where $H(.)$ is the entropy, $H(X \mid Y)$ and $H(Y \mid X)$ are the conditional entropies, and $H(X;Y)$ is the joint entropy of X and Y that are defined by:

$$H(x) = -\int_x p_X(x) \log p_X(x) dx \quad (2)$$

$$H(Y) = -\int_y p_X(y) \log p_Y(y) dy \quad (3)$$

$$H(X;Y) = -\int_x \int_y p_{X,Y}(x,y) \log p_{X,Y}(x,y) dx dy \quad (4)$$

Where $p_{X,Y}(x,y)$, $p_X(x)$ and $p_Y(y)$ are the joint probability density function and marginal density functions of X and Y, respectively. The marginal density functions are given by:

$$p_X(x) = \int_y p_{X,Y}(x,y)dy \quad (5)$$

$$p_Y(y) = \int_x p_{X,Y}(x,y)dx \quad (6)$$

By replacing (2) - (4) into (1), the MI equation will be:

$$I(X;Y) = \int_x \int_y p_{X,Y}(x,y) \log \frac{p_{X,Y}(x,y)}{p_X(x)\, p_Y(y)} dxdy \quad (7)$$

In discrete forms, the integration is substituted by summation over all possible values that appear in data. Therefore, it is only required to estimated $p_{X,Y}(x,y)$ in order to estimate the MI between X , Y by (5) to (7). Histogram- and kernel-based methods are widespread to estimate probability density function [16]. MI is the Kullback-Leibler distance between the joint distribution $p_{X,Y}(x,y)$ and the product distribution $p_X(x)$, $p_Y(y)$.

For estimating MI, a recent estimator based on entropy is used, that is estimated from k-nearest neighbor's statics. The basic idea is to estimate entropy from the average distance to the k-nearest neighbors (over all of data).

In practice, one has a set of N input-output pairs, $z_i=(x_i,y_i)$, $i=1,...,N$, which are assumed to be realizations of a random variable $Z=(X,Y)$ with density $p_{X,Y}(x,y)$. Either X and Y have values in R or in R^p. and the algorithm will use the Euclidean norm in those spaces.

Input-output pairs are compared through the maximum norm:

$$\left\| z - z' \right\|_\infty = \max\left\{ \left\| x - x' \right\|, \left\| y - y' \right\| \right\} \quad (8)$$

It can be considered that K is a fix positive integer; then $z_k(i) = (x_k(i), y_k(i))$ is the k-th nearest neighbor of z_i (with a maximum norm). Then it is denoted by:

$$\varepsilon_i/2 = \left\| z_i - z_{k(i)} \right\|_\infty \quad (9)$$

$$\varepsilon_i^x/2 = \left\| x_i - x_{k(i)} \right\|, \varepsilon_i^y/2 = \left\| y_i - y_{k(i)} \right\| \quad (10)$$

$\varepsilon_i/2$ is the distance from z_i to its k-th neighbor and $\varepsilon_i^x/2$ and $\varepsilon_i^y/2$ are the distances between the same points projected into X and Y subspaces. Obviously, $\varepsilon_i = \max\left\{ \varepsilon_i^x, \varepsilon_i^y \right\}$

n_{ix} and n_{iy} are the numbers of sample points with $||x_i - x_j|| \le \varepsilon_i^x/2$ and $||y_i - y_j|| \le \varepsilon_i^y/2$. Estimation of MI is then:

$$\hat{I}(X;Y) = \psi(K) - \frac{1}{k} - \frac{1}{N}\sum_{i=1}^{N}\left[\psi(n_i^x) + \psi(n_i^y)\right] + \psi(N) \quad (11)$$

Where ψ is the digamma function:

$$\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)} = \frac{d}{dx}\ln\Gamma(x), \psi(1) \approx -0.5772156 \quad (12)$$

Where:

$$\Gamma(x) = \int_0^\infty u^{x-1} e^{-u} du \quad (13)$$

With a small value for K, this estimator has a large variance and a small bias, whereas a large value of K leads to a small variance and a large bias. In this paper, $k=6$ is used

IV. Forward Feature Selection Algorithm

In this algorithm, each feature is inserted to the set of selected features in order to maximize MI among selected feature and output. This procedure is iterated until n input features have been selected where n is determined a priori. The algorithm can be described totally by the following procedure [17]:

1) Initialization: Set L ←'initial set of f features', S ←'empty set', O ← 'class-labels'.
2) Computation of the mutual information with the class- labels: For each feature l ϵ L compute I (O; l).
3) Choice of the first feature: Find the feature f that maximizes I (O, l); Set L← L-{l}, S← {l}.
4) Greedy selection: Repeat until desired numbers of features are selected:
   a. Computation of the mutual information between features: For all couples of features (l, s) with l ϵ L, s ϵ S, if it is not already available, compute I (l; s).
   b. Selection of the next feature: Chose the feature l ϵ L as the one that maximizes I (O,l); seO L← L-{l}, S← S U {l}
5) Output the set containing the selected features

V. Mutual Information Based Feature Selection Algorithm

MIFS is originally proposed by Battiti [4]. The objective is to maximize the relevance between the features and the output and minimize the redundancy of the selected features. The original algorithm computes I (O;*l*) and I(*l*; *l'*) and. The algorithm chooses one feature at a time; the one maximizing the information with class-labels.

This mutual information expression is corrected by subtracting a quantity proportional to the average mutual information within the selected features. The algorithm can be described by the following procedure:

1) *Initialization*: Set L ←'initial set of *f* features', S ←'empty set', O ← 'class-labels'.
2) *Computation of the mutual information with the class-labels*: For each feature *l* ϵ *L* compute I(O; *l*).
3) *Choice of the first feature*: Find the feature *l* that maximizes I(O,*l*); Set L← L-{*l*}, S← {*l*},.
4) *Greedy selection*: Repeat until desired number of features is selected:
   a. *Computation of the mutual information between features*: For all couples of features (*l*, *s*) with *l* ϵ L, *s* ϵ S, if it is not already available, compute I (*l*; *s*).
   b. *Selection of the next feature*: Choose the feature *l* ϵ L as the one that maximizes $I(O;l)$; set L← L-{*l*}, S← S U {*l*}
5) Output the set containing the selected features.

β is a parameter to adjust the relative importance of MI between the candidate feature and the already selected features with respect to the MI with the output. If β = 0 the algorithm only attempts to maximize mutual information with output, so the dependency between feature is never considered.

VI. Linear Correlation Based Feature Selection Algorithm

Linear correlation coefficient is very popular in statistic. It is able to show the strength and direction of a linear relationship between random variable. For Feature X with values x and classes Y with values y treated as random variables it is defined as:

$$corr(X,Y) = \frac{E(XY) - E(X)E(Y)}{\sqrt{\delta^2(X)\delta^2(Y)}} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_j (y_j - \bar{y})^2}} \quad (14)$$

If *X* and *Y* are linearly dependent, the *corr (X, Y)* is equal to ±1 and zero if they are completely independent. For feature selection, the followed algorithm is proposed [17]:

1) Initialization: Set F← 'initial set of f features', S ←'empty', O ← 'class-labels'
2) Computation of the correlation coefficient with the output: For each feature f ϵ F compute corr(O; *f*).
3) Choice of the first feature: Find the feature f that maximizes corr(O, *f*); Set F← F-{*f*}, S← {*f*}.
4) Greedy selection: Repeat until desired number of features is selected:
   a. Computation of the correlation coefficient between features: For all couples of features (*f*, s) with *f* ϵ F, *s* ϵ S, if it is not already available, compute *corr (f; s)*.
   b. Selection of the next feature: Choose the feature f ϵ L as the one that maximizes $corr(T;f) - \frac{\beta}{|S|}\sum_{s\in S} corr(f;s)$; set F← F-{*f*}, S← S U {*f*}
5) Output the set containing the selected features

VII. Locally Linear Neuro Fuzzy Models

The fundamental idea for application of the locally linear neuro fuzzy model is to divide the input space into small linear subspaces with fuzzy validity functions which show the validity of each linear model in its region. The structure of LLNF is shown in Fig. 1. In this study, the validity function is the Gaussian function as:

$$\mu(x) = \exp\left(-\frac{(x-c)^2}{2\delta^2}\right) \quad (15)$$

The number of validity function and their parameter, the center c and the standard deviation δ define the partitioning of input space. In each iteration, a new Local linear model is added to the model. Any produced linear part with its validity function can be described as a fuzzy neuron.

So the total model is a neuro fuzzy network with one hidden layer, and a linear neuron in the output layer which simply calculates the weighted sum of the outputs of locally linear neurons as follows:

$$\hat{y}_i = \omega_{i_0} + \omega_{i_1} u_1 + \omega_{i_2} u_2 + \cdots + \omega_{i_p} u_p \quad (16)$$

Where $[u_1 u_2 \ldots u_p]^T$ is the model input, M is the number of locally linear neurons and ω_{ij} denotes the linear estimation parameters of the i-th neuron. The validity Functions are selected as normalized Gaussians:

$$\emptyset_i(\underline{u}) = \frac{\mu_i(\underline{u})}{\sum_{j=1}^{M} \mu_j(\underline{u})} \quad (17)$$

$$\mu_i(\underline{u}) = \exp\left[-\frac{1}{2}\frac{(u_1 - c_{i1})^2}{\delta^2_{i1}}\right] \times \ldots \times \exp\left[-\frac{1}{2}\frac{(u_p - c_{ip})^2}{\delta^2_{ip}}\right] \quad (18)$$

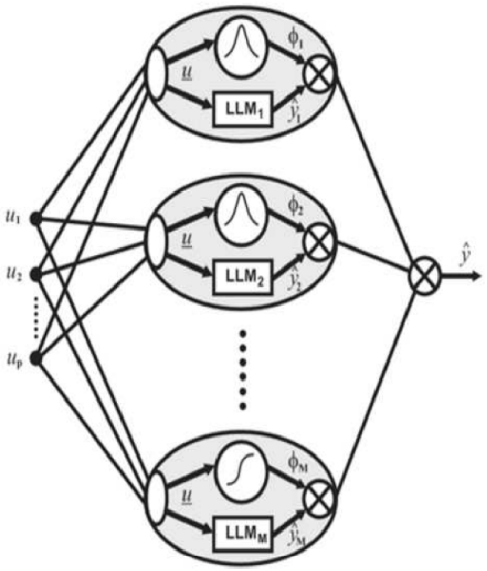

Figure 1: Topology of the Locally Linear Neuro Fuzzy model

M × P parameters of the nonlinear hidden layer are the parameters of Gaussian validity functions: center (c_{ij}) and standard deviation (σ_{ij}). Learning methods or Optimization are applied to tune two sets of parameters, parameters of local linear model (ω_{ij}) and the parameter of validity fuction (c_{ij}, σ_{ij}). Global optimization of linear parameters is simply got by least squares technique.

Locally Linear Model Tree (LOLIMOT) algorithm as an incremental tree-based algorithm is used to adjust the rule premise parameters (i.e. Determining the validation hypercube for each locally linear model) [18]. In each iteration, the worst performing locally linear neuron is determined to be divided.

All the possible divisions in the P dimensional input space are checked and the best is performed. The fuzzy validity functions for the new structure are updated.

The standard deviations are usually set as 0.7. For more detail refer to [18].

The computational complexity of LOLIMOT is $O(2MP^4)$. Hence the computational demand grows linearly with the model complexity, that is, with the number of Gaussian. Also, the computational demand grows exponentially with the input dimensionality.

VIII. Intrusion Dataset

The data set used in this experiment is the "KDD Cup 1999 data" [19]. The raw training data was about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic which processed into about five million connection records. A connection is a sequence of TCP packets starting and ending at some well defined times, during which data flows from a source IP address to a target IP address under some well defined protocol. Each record is unique in the data set with 41 continuous and nominal features plus one class label as either normal or as an attack. In this work, nominal features such as protocol (TCP/UDP/ICMP), service type (http/ftp/telnet/…) and TCP status flag (SF/REJ/…) have been converted into a numeric feature.

Derived Features can be classified into four categories. The first category, numbered 1-9, are basic Feature of individual TCP connections. The next category, numbered 10-22, are content features. The third category, numbered 23-31, are traffic features computed using a two-second time window and the forth category, numbered 32-41, are traffic features computed using a two–second time window from destination to host. The label of the features and their corresponding network data features are shown in Table 1.

Attacks have been classified into four categories: probing, denial of service (DoS), user to root (U2R), and remote to user (R2L). Details of these categories are as follows [20].

TABLE 1: NETWORK DATA FEATURE LABELS

Category 1		Category 2		Category 3		Category 4	
Label	network feature	Label	network feature	Label	network feature	Label	network feature
1	duration	10	hot	23	count	32	srv-diff-host-rate
2	protocol-type	11	num-falied-logins	24	srv-count	33	dst-host-srv-count
3	service	12	logged-in	25	serror-rate	34	dst-host-same-srv-rate
4	flag	13	num-compromised	26	srv-serror-rate	35	dst-host-diff-srv-rate
5	src-bytes	14	root-shell	27	rerror-rate	36	dst-host-same-src-port-rate
6	dst-bytes	15	su-attempted	28	srv-rerror-rate	37	dst-host-srv-diff-host-rate
7	land	16	num-root	29	same-srv-rate	38	dst-host-serror-rate
8	wrong-fragment	17	num-file-creations	30	diff-srv-rate	39	dst-host-srv-serror-rate
9	urgent	18	num-shells	31	srv-diff-host-rate	40	dst-host-rerror-rate
		19	num-access-files			41	dst-host-srv-rerror-rate
		20	num-outbound-cmds				
		21	is-host-login				
		22	is-guest-login				

IX. Experiment And Result

The training and test data comprises of 6480 and 6703 randomly generated records from five classes. Most of the selected connections are normal, which is generally the case in real-world networks. In this data set, there are 52 records of U2R attack type, which 32 of them are used for training and remaining records are used for testing. The LLNF has been used for binary classifications to represent normal activity (Normal). This classifier separate Normal from non-Normal class.

Experiments have four phases: data normalization, data reduction, training, and testing phases. First, train and test data normalized into the unit interval [-1 1]. Then, in terms of important and useful, the input features for intrusion detection are sorted by feature selection algorithm. In the training, LLNF construct a model using the training data to give maximum generalization accuracy on the unseen data. The test data are then passed through the saved trained model to detect intrusions in the testing phase. With considering to the LNFF is architecture for system identification and its output is continuous, the ROC curves are computed for the best threshold separating Normal Class from non-Normal (Fig.2). The optimal performance is obtained when output which is more than zero is labeled *Normal* and otherwise is labeled *Attack.*

The feature selection algorithm used in this paper can only rank feature in terms of their importance and cannot show how many features are optimal. So, for determining the best number of features, this work is started with the best feature and incrementally adds features by their importance to PLLNF. The optimal number of features in each algorithm is ones which have shown best result of classification on the training and test data. The best selected features for each class are shown in Table 2.

Results of classification with 99% confidence interval are shown in Table 3. Results have been shown that feature selection improves the classification accuracy in comparison to not using this phase. Due to low computational complexity of LCFS than the other feature selection algorithms, the PLLNF performance with LCFS is used for comparison.

In Table 4, performance our IDS and another related IDSs are shown, it has been represented the PLLNF have a good performance respect to another IDSs.

X. Discussion

While LCFS and FFS can be useful in particular cases, MIFS have the capability to measure a general dependence between features and to rank them. Since MIFS and FFSA use more information to select features, these techniques usually lead to optimal result.

The experiment results indicate that FFSA and MIFS have best results in anomaly detection and perform similarly to one another. Thus, the features of this class are almost independent from one another. But, experiment has shown that LCFS prefer to use due to its lower computing complexity than the other.

It's found that by adding preprocess phase for determining the most significant feature, rather than the all 41-feature set, the change in performance of intrusion detection was statistically important and complexity of LLNF model have reduced, it means that with the lower number of Gaussian has been shown the better result and faster. In addition, the false positive rate of PLLNF is very low with respect to the other IDS.

XI. Conclusion and Future Work

Our model called PLLNF, introduced a new IDS which has used a feature selection phase which applied either an information theatric or statistical criterions. Our IDS use LLNF model for classification however this architecture originally is use for system identification. LOLIMOT algorithm is applied for learning LLNF model parameters. The high computational efficiency of the LOLIMOT algorithm is to a great part a consequence of the utilization of linear parameter estimation methods. Experiments on KDD CUP 99 showed the PLLNF anomaly detection have improved the classification accuracy. The feature selection method used in this experiment has reduced complexity LLNF model. Having very low false Positive rate which is important in design IDS is the advantage of PLLNF anomaly detection respect to other IDSs.

TABLE 2: Selected Features for Normal Class

Method	# Feat.	Selected Features
FFSA	14	5, 2, 34, 6, 33, 40, 37, 27, 24, 23, 38, 32, 17, 35
MIFS	5	5, 2, 6, 12, 33, 37, 23, 34, 35, 3, 24, 29, 1, 30, 41, 36, 28, 38, 27, 39, 31, 25
LCFS	24	12, 23, 2, 34, 1, 23, 3, 36, 29, 27, 28, 39, 38, 24, 41, 26, 25, 30, 32, 10, 22, 37, 8, 31

TABLE 3: Performance of classification for anomaly Detection (DR: Detection Rate, FPR: false Positive Rate, DE: Detection Error)

class	Method	DR (%)	FPR (%)	Accuracy (%)	Complexity (# Gaussian)
Normal	**FFSA**	99.62±0.55	0.012±0.0014	99.85±0.15	14
	MIFS	99.5±0.2	0.012±0.022	99.86±0.05	15
	LCFS	99.49±0.11	0.02±0.05	99.85±0.05	14
	all features	99.05±0.83	0.032±0.0239	99.67±0.25	20

FIGURE 2_A: ROC CURVE FOR PLLNF WITH MIFS

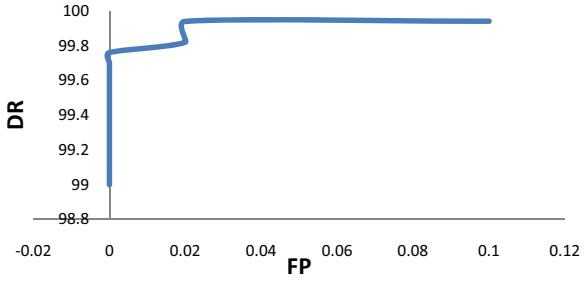

FIGURE 2_B: ROC CURVE FOR PLLNF WITH LCFS

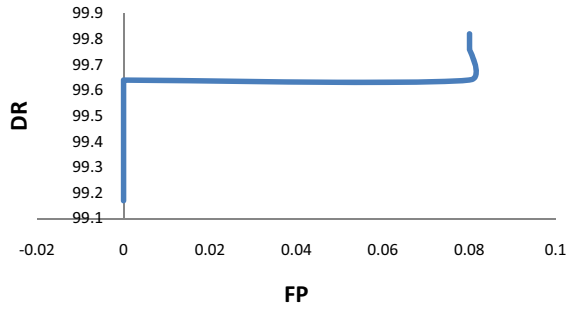

FIGURE 2_C: ROC CURVE FOR PLLNF WITH FFSA

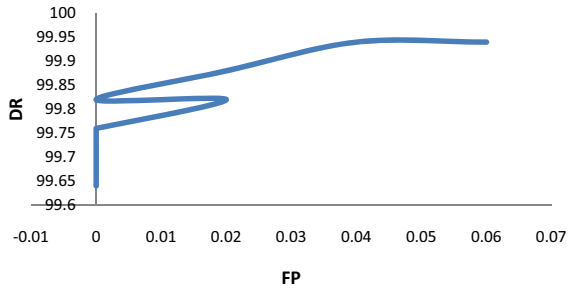

FIGURE 2_D: ROC CURVE FOR PLLNF (TOTAL FEATURE)

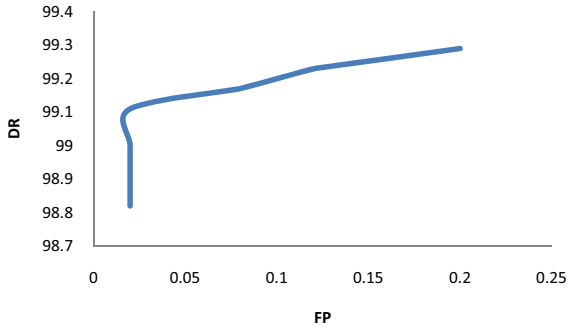

TABLE 4: THE PERFORMANCE COMPARISON WITH THE OTHER APPROACHES

Methods	Accuracy (%)	DR(%)	FP(%)
PLLNF	**99.85**	**99.49**	**0.02**
SVM [1]	99.55	-	-
Bayesian [2]	99.57	-	-
FNT [3]	99.19	65.0	0.98
SVM [4]	99.51	-	-
SRPP[10]	-	99.08	3.85

REFERENCES

[1] S. Chebrolu, A. Abraham, P. Thomas, "Feature deduction and ensemble design of intrusion detection systems, " *Computers & Security*, vol .24, issue 4, (2005) pp.295-307.

[2] Y. Chena, A. Abrahama, B. Yanga, "Feature selection and classification using flexible neural tree," Journal of Neurocomputing 70 (2006) 305–313

[3] S. B. Cho, "Incorporating soft computing techniques into a probabilistic intrusion detection system," *IEEE Transactions on Systems, MAN, and Cybernetics part C: Applications and Reviews*, vol. 32, pp. 154–160, May 2002.

[4] Battiti, R.: "Using Mutual Information for Selecting Features in Supervised Neural Net Learning," *IEEE Transactions on Neural Networks*. 5 (1994), p. 537–550

[5] T.F. Lunt, A. Tamaru, F. Gilham, R. Jagannathm, C. Jalali,P.G. Neumann, H.S. Javitz, A. Valdes, T.D. Garvey, "A Real-time Intrusion Detection Expert System (IDES)," Computer Science Laboratory, SRI International, Menlo Park, CA, USA, Final Technical Report, February 1992.

[6] L. Ertoz, E. Eilertson, A. Lazarevic, P.-N. Tan, V. Kumar, J. Srivastava, P. Dokas, "The MINDS - Minnesota intrusion detection system, in: Next Generation Data Mining," MIT Press, Boston, 2004.

[7] A. Lazarevic, L. Ertoz,, V. Kumar, A. Ozgur and J. Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection, " *in Proc. of Third SIAM Conference on Data Mining* (May 2003).

[8] H. Debar, M. Becker and D. Siboni, "A neural network component for an intrusion detection system, " *in Proc. of IEEE Computer Society Symposium on Research in Security and Privacy* (Oakland, CA, May 1992) 240-250.

[9] M. Ramadas, S.O.B. Tjaden, "Detecting anomalous network traffic with self-organizing maps, " *in Proc. the 6th International Symposium on Recent Advances in Intrusion Detection*, Pittsburgh, PA, USA, 2003, pp. 36–54.

[10] M. Saniee Abadeh, J. Habibi, C. Lucas, "Intrusion detection using a fuzzy genetics-based learning algorithm, " *Journal of Network and Computer Applications*, Volume 30, Issue 1, January 2007, Pages 414-428

[11] W.W. Cohen, "Fast effective rule induction," *in Proc. of the 12th International Conference on Machine Learning*, Tahoe City, CA, 1995, pp. 115–123.

[12] S. Mukkamalaa, A.H. Sunga, A. Abrahamb, "Intrusion detection using an ensemble of intelligent paradigms," *Journal of Network and Computer Applications 28* (2005) 167–182.

[13] Z. Zhang, and H. Shen, "Application of online-training SVMs for real-time intrusion detection with different considerations," *Computer Communications*, vol. 28, issue 12, pp. 1428-1442, 2005.

[14] J.E. Dickerson, J.A. Dickerson, Fuzzy network profiling for intrusion detection, *in: Proc. 19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, Atlanta, GA, 2000, pp. 301–306.

[15] A. Sung, S. Mukkamala, Identifying important features for intrusion detection using support vector machines and neural networks, " *In: Proc. International Symposium on Applications and the Internet* (SAINT 2003); 2003. p. 209e17.

[16] M. Rezaei Yousefi, M. Mirmomeni, A. Vahabie, C. Lucas, C: "Near Optimal Feature Selection Using Mutual Information for Classification

Problems," *In Proc. the International Joint Conference on Knowledge Management for Composite Materials* (kcmc2007),

[17] F.Amiri, M. Rezaei Yousefi, C. Lucas, N.Yazdani, R.Rahmani, "Improved Feature Selection for Intrusion Detection System", unpublished.

[18] O.Nelles, *NonLiner System Identification from classical Approches to Neural Networks and Fuzzy Models*. New York, Springer-Verlag 2001, ch 13.

[19] http://www.esat.kuleuven.ac.be/sista/lssvmlab/
http://kdd.ics.uci.edu//databases/kddcup99/kddcup99. html

[20] S. Mukkamala, A. Sung, and A. Abraham, "Intrusion detection using ensemble of soft computing and hard computing paradigms, " *Journal of Network and Computer Applications*, Elsevier Science, vol. 28, issue 2, pp. 167-182, 2005