PERFORMANCE COMPARISON OF INTRUSION DETECTION SYSTEM CLASSIFIERS USING VARIOUS FEATURE REDUCTION TECHNIQUES

V. VENKATACHALAM, S. SELVAN

Erode Sengunthar Engineering College, Erode,
H.O.D/IT P.S.G College of Technology, Coimbatore
E-mail: vv01062007@hotmail.com

Abstract: In this paper, we compare the performance of Intrusion Detection System Classifiers using various feature reduction techniques. To enhance the learning capabilities and reduce the computational intensity of competitive learning neural network classifiers, different dimension reduction techniques have been proposed. These include: Principal Component Analysis, Linear Discriminant Analysis, Independent Component Analysis. Many Intrusion Detection Systems are based on neural networks. However, they are computationally very demanding . In order to mitigate this problem, dimension reduction techniques are applied to a given dataset to extract important features. In the proposed research various classifiers are applied to the reduced feature dataset and their performance is compared. On the basis of these results, a technique is proposed which performs exceptionally well, in terms of both accuracy and computation time. When applied to the KDDCUP99 reduced feature dataset, this technique performs better than a standard learning schema based on the full featured dataset.

Keywords: Intrusion detection, Neural networks, PCA, LDA, ICA

1. INTRODUCTION

Intrusion Detection Systems (IDSs) are amongst the main tools for providing security in computer systems and networks. They detect intrusions and attacks through the analysis of TCP/IP packet data. Based on the data source, IDSs are classified into host-based and network-based. Also depending on the analysis approach, IDSs are categorized into misuse detection and anomaly detection systems. Misuse detection systems detect known attacks using pre-defined attack patterns and signatures. Anomaly detection systems detect attacks by observing deviations from the normal behaviour of the system. Supervised and unsupervised nets have been used in IDSs. Most supervised neural net architectures require retraining, to account for changes in the input data. Unsupervised nets offer an increased level of adaptability to neural nets, and have been used in intrusion detection systems.

To enhance the learning capabilities and reduce the computational intensity of competitive learning neural network, different dimension reduction techniques such as Principal Component Analysis, Linear Discriminant Analysis, Independent Component Analysis. Many Intrusion Detection Systems are applied to KDDCUP99 (Aapo and Oja, 2000; Balakrishnama, 1998), a well known dataset. Many Intrusion Detection Systems based on neural networks have been proposed. However, they are computationally very demanding and they face high misclassification rate. In order to mitigate these problems dimension reduction techniques are applied to the training dataset to extract important features. Various neural network classifiers such as Gaussian mixture, RBF, Binary tree, LAMSTAR, SOM, ART are applied to the reduced feature dataset and their performance is compared.

2. DIMENSION REDUCTION

In statistical terms, dimension reduction (Berchtold et al., 1998; Fodor, 2002; Gopi et al., 2004) is the process of reducing the number of random variables under consideration. This process can be divided into feature selection and feature extraction

The curse of dimensionality is a term coined by Richard Bellman to describe the problem caused by the exponential increase in volume associated to the addition of extra dimensions to a (mathematical) space.

The curse of dimensionality is a significant obstacle in machine learning problems that involve learning from few data samples in a high-dimensional feature space.

2.1 Predictive Data Mining

One of the predictive tasks of Data Mining is that of finding some form of classification of the items contained in the data mart from a set of raw data. When there is a finite set of classes that describe the domain of the data, the classification can be carried out by means of if-then rules that help users to classify a new item in one of such predefined classes. Such classification process is based on the values of some characteristics of the item itself and can be deterministic (e.g. there is no doubt about the membership of the item to the given class) or heuristic (e.g. the association of the item to one or more classes is given with a degree of certainty).

The association model can take the form of a decision tree, rather than a set of if-then rules, but the purpose of the model remains the same. When the classification domain is not finite (e.g. when the examined variable is a real number) the operation is called regression. The regression task models the set of data submitted to the task and can be used to predict new, not submitted, values.

2.2 Curse of Dimensonality

It is intuitive to think that increasing the dimension of the features should never reduce the recognizer's performance, since we are providing a larger, or at least the same, amount of information. Therefore the worst that could happen should be that performance would remain the same. As practice shows, this is unfortunately not the case; the performance can decrease even though we feed more data to the system. This behaviour is due to the finite amount of training data that can be presented to the model. In theory we normally assume the training data to be infinite and so the model could be perfectly trained under all circumstances. In practice this is not possible and if we chose a model that is too complex then it would be unlikely that all of our parameters could be well estimated. On the other hand the model should not be too simplifyed either.

2.3 Feature Reduction Techniques

Feature extraction (Graupe, 1997; KDDCUP99, 1999) applies a mapping of the multidimensional space into a space of fewer dimensions. This means that the original feature space is transformed by applying e.g. a linear transformation using principal components analysis.

Feature extraction involves simplifying the amount of resources required to accurately describe a large set of data. When performing the analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power or a classification algorithm which over-fits the training sample and generalizes poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy.

We considered three dimension reduction techniques in this work

i. Linear Discriminant Analysis
ii. Independent Component Analysis
iii. Principal Component Analysis

2.4. Linear Discriminant analysis

LDA (Jing et al., 2006) finds the optimal transformation matrix as to preserve most of the information that can be used to discriminate between the different classes. Therefore the analysis requires the data to have appropriate class labels. In order to mathematically formulate the optimization procedure

$$\overline{x_j} = \frac{1}{N_j}\sum_{i=1}^{N_j} x_i \qquad (1)$$

$$\overline{W_j} = \frac{1}{N_j}\sum_{i=1}^{N_j}(x_i - \overline{x_j})(x_i - \overline{x_j})^{\mathrm{T}} \qquad (2)$$

$$\overline{x} = \frac{1}{N}\sum_{i=1} x_i \qquad (3)$$

$$\overline{T} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})(x_i - \overline{x}) \qquad (4)$$

we have to compute the mean vector and the covariance matrix for each class and for the complete data set (with all classes pooled together)

$$\sum_{j=1}^{J} N_j = N \qquad (5)$$

$$\hat{\theta} = \arg \max_{\theta_p} \frac{|\theta_p^T \overline{T} \theta_p|}{|\theta_p^T \overline{W} \theta_p|} \qquad (6)$$

Where

$$\overline{W} = \frac{1}{N}\sum_{j=1}^{J} N_j \overline{W_j} \qquad (7)$$

In the above formulas N denotes the total number of training tokens and Nj stands for the number of training tokens in class j. Naturally, the number of classes is J.

With these definitions, we can easily formulate the optimization criterion. Namely the numerator represents the covariance of the pooled training data in the transformed feature space. The denominator represent the average covariance within each class in the transformed feature space. Hence, the criterion really tries to maximize the 'distance' between classes, while minimizing the 'size' of each of the classes at the same time. This is exactly what we want to achieve because this criterion guarantees that we preserve most of the discriminant information in the transformed feature space. It turns out that the optimum matrix according to the above formula can be found in a fairly easy way. The result consists of those eigenvectors that correspond to the p largest eigenvalues. LDA is applied to the KDDCUP99 data and the features selected are shown in table 1. Tables 2 to 7 shows the performance of the various neural network classifiers using 17 features.

Table 1: Best 17 features selected after linear discriminant analysis

S.no	Feature	Description
0	duration	Continuous
1	protocol_type	Symbolic
2	Service	Symbolic
3	src_bytes	Continuous
4	land	Symbolic
5	wrong_fragment	Continuous
6	num_failed_logins	Continuous
7	logged_in	Symbolic
8	root_shell	Continuous
9	num_file_creations	Continuous
10	is_guest_login	Symbolic
11	count	Continuous
12	srv_count	Continuous
13	serror_rate	Continuous
14	srv_serror_rate	Continuous
15	diff_srv_rate	Continuous
16	dst_host_count	Continuous

Table 2 : Confusion matrix for Gaussian mixture IDS (17 features) CPE = 0.3309 FP=0.18 FN=10.31

Predicted Actual	Normal	Probe	DOS	U2R	R2L	% correct
Normal	60024	380	178	5	6	99.1
Probe	193	3882	91	0	0	93.2
DOS	17930	8966	202943	9	5	88.3
U2R	148	21	2	52	5	22.8
R2l	13815	613	6	30	1725	10.6
%correct	65.2	28.0	99.9	54.2	99.1	

Table 3: Confusion Matrix for RBF IDS (17 features) CPE = .4665 FP=.37 FN=14.74

Predicted Actual	Normal	Probe	DOS	U2R	R2L	%correct
Normal	59435	573	575	8	2	98.1
Probe	530	3554	78	3	1	85.3
DOS	37850	21191	170792	15	5	74.3
U2R	195	13	2	18	0	7.9
R2l	7280	7810	200	0	899	5.5
%correct	56.4	10.7	99.5	40.9	99.1	

Table 4: Confusion Matrix for SOM IDS (17 features) CPE =.2436 FP=1.23 FN=6.74

Predicted Actual	Normal	Probe	DOS	U2R	R2L	%correct
Normal	56750	3548	285	8	2	93.6
Probe	1290	2519	350	5	2	60.5
DOS	8193	1125	220530	4	1	95.9
U2R	102	69	9	47	1	20.6
R2l	11395	3026	7	0	1761	10.9
%correct	73.0	24.5	99.7	73.4	99.7	

Table 5: Confusion Matrix for Binary Tree Classifier IDS(17 features) CPE=.2283 , FP=.60, FN=5.44

Predicted Actual	Normal	Probe	DOS	U2R	R2L	%correct
Normal	58700	1300	583	7	3	96.9
Probe	569	3096	493	6	2	74.3
DOS	8204	1730	219915	3	1	95.7
U2R	99	99	3	26	1	11.4
R2l	8061	1003	7037	0	88	0.54
%correct	77.6	42.8	96.4	61.9	92.6	

Table 6: Confusion Matrix for ART IDS (17 features) CPE=.2295 FP=.59 FN=5.37

Predicted Actual	Normal	Probe	DOS	U2R	R2L	%correct
Normal	58750	773	1068	1	1	96.9
Probe	106	4001	58	0	1	96.0
DOS	4210	2806	222833	3	1	96.9
U2R	57	128	4	39	0	17.1
R2l	12360	1550	474	1	1804	11.1
%correct	77.8	43.2	99.3	88.6	99.8	

Table 7: Confusion Matrix for LAMSTAR IDS (17 features) CPE=.1304 FP=0.11 FN=3.40

Predicted Actual	Normal	Probe	DOS	U2R	R2L	%correct
Normal	60239	284	65	4	1	99.4
Probe	118	3982	65	0	1	95.6
DOS	2583	551	226715	3	1	98.6
U2R	98	52	5	72	1	31.6
R2L	7801	473	1810	0	6105	37.7
%correct	85.0	74.5	99.1	91.1	99.9	

2.5. Independent Component Analysis

ICA (Jolliffe, 2002) is very closely related to the method called blind source separation (BSS) or blind signal separation. ICA is one method, perhaps the most widely used, for performing blind source separation.

A relevant feature is defined as a feature whose removal deteriorates the performance or accuracy of the classifier, while an irrelevant or redundant feature is a not relevant one. Irrelevant features could deteriorate the performance of a classifier that uses all the features because irrelevant information is taken into account. Thus the motivation of a feature selector is (i) simplifying the classifier by the selected features; (ii) improving or not significantly reducing the accuracy of the classifier; and (iii) reducing the dimensionality of the data so that a classifier can handle large values of data.

ICA techniques provide statistical signal processing tools for optimal linear transformations in multivariate data. These methods are well-suited for feature extraction, noise reduction, density estimation and regression. The ICA problem can be described as follows, each of h mixture signal x1(k), x2(k),…,xh(k) is a linear combination of q independent components s1(k), s2(k),…,sh(k) , that is, X = AS where A is a mixing matrix. Given X, the problem is to compute A and S. Based on the following two statistical assumptions, ICA successfully obtains the results: 1) the components are mutually independent; 2) each component follows a non-gaussian distribution. By X = AS, we have S=A-1 S = A inverse of X=WX (where W = A inverse).

The task is to select an appropriate W which applied on X to maximize the non-gaussian behaviour of the components. This can be done through an iterative procedure. Given a set of n-dimensional vectors, the independent components are the directions (vectors) along which the statistics of the projections of the data vectors are independent of each other. Formally, if A is a transformation from the given reference frame to the independent component reference frame, then

X = AS indicates that

$$P(S) = \prod P_a(S_i), where P_a(.) \qquad (8)$$

is the marginal distribution and p(s) is the joint distribution over the n-dimensional vector s. Usually, the technique for performing independent component analysis is expressed as the technique for deriving one particular W, y = Wx, Such that the components of y become independent of one another. If the individual marginal distributions are non-gaussian then the derived marginal densities become a scaled permutation of the original density functions if one such W can be obtained. One general learning technique to find a suitable W is

$$W = \eta(I - \phi(y)y_T)W, \qquad (9)$$

Where $\phi(y)$ is a nonlinear function of the output vector y. ICA is applied to the KDDCUP99 data and the features selected are shown in table 8. Tables 9 to 14 show the performance of various neural network classifiers using 12 features.

Table 8: Best 12 features selected after independent component analysis

S.no	Feature	Description
0	service	Symbolic
1	src_bytes	Continuous
2	dst_bytes	Continuous
3	logged_in	Symbolic
4	count	Continuous
5	srv_count	Continuous
6	serror_rate	Continuous
7	srv_rerror_rate	Continuous
8	srv_diff_host_rate	Continuous
9	dst_host_count	Continuous
10	dst_host_srv_count	Continuous
11	dst_host_diff_srv_rate	Continuous

Table 9: Confusion Matrix for Gaussian mixture IDS (12 features) CPE = .3417 FP=.34 FN=10.53

Predicted ACTUAL	NORMAL	PROBE	DOS	U2R	R2L	% Correct
Normal	59510	692	380	5	6	98.2
Probe	295	3726	145	0	0	89.4
DOS	18445	10472	200922	9	5	87.4
U2R	153	27	1	42	5	18.4
R2l	13875	775	5	30	1504	9.3
%correct	64.5	23.7	99.7	48.8	98.9	

Table 10: Confusion Matrix for RBF IDS (12 features) CPE = .3417 FP=0.34 FN=10.53

Actual	Normal	Probe	DOS	U2R	R2L	%correct
Normal	59400	593	590	8	2	98.03
Probe	550	3514	98	3	1	84.34
DOS	37875	21216	170742	15	5	74.28
U2R	196	15	2	15	0	6.57
R2l	7294	7820	200	0	875	5.4
%correct	56.4	10.6	99.5	36.6	99.1	

Table 11: Confusion Matrix for Binary Tree Classi- fier IDS (12 features) CPE=.2309 FP=0.67 FN=5.51

Predicted Actual	Normal	Probe	DOS	U2R	R2L	%correct
Normal	58501	1402	680	7	3	96.5
Probe	635	2950	573	6	2	70.8
DOS	8354	1523	219971	4	1	95.7
U2R	99	97	5	26	1	11.4
R2l	8063	1000	7055	0	71	0.43
%correct	77.3	42.3	96.4	60.5	91	

Table 12: Confusion Matrix for SOM IDS (12 12 features) CPE =.2485 FP=1.22 FN=6.86

Predicted Actual	Normal	Probe	DOS	U2R	R2L	%correct
Normal	56400	3773	410	8	2	93.0
Probe	1305	2474	380	5	2	59.4
DOS	8523	1324	220000	5	1	95.7
U2R	103	70	10	44	1	19.3
R2l	11410	3041	12	0	1726	10.6
% correct	72.5	23.2	99.6	71	99.6	

Table 13: Confusion Matrix for ART Classifier IDS(12 features) CPE=.2309 FP=.67 FN=5.51

Predicted Actual	Normal	Probe	DOS	U2R	R2L	%correct
Normal	58625	848	1118	1	1	96.7
Probe	166	3901	98	0	1	93.6
DOS	4460	2956	222433	3	1	96.8
U2R	58	129	5	36	0	15.8
R2l	12400	1580	484	1	1724	10.6
%correct	77.4	41.4	99.2	87.8	99.8	

Table 14: Confusion Matrix for LAMSTAR IDS (12 features) CPE = .3417 FP=.34 FN=10.53

Predicted Actual	Normal	Probe	DOS	U2R	R2L	% correct
Normal	60150	341	97	4	1	99.3
Probe	248	3700	217	0	1	88.9
DOS	2688	661	226500	3	1	98.5
U2R	99	59	9	60	1	26.3
R2L	7819	450	2920	0	5900	36.4
%correct	84.7	71.0	98.6	89.5	99.9	

2.6. Principal Component Analysis

Principal Component Analysis(PCA) (Khaled and Vemuri, 2002; Kordylewski, 1998; Morteza et al., 2004) is one of the most widely used dimensionality reduction techniques for data analysis and compression. This technique identifies patterns in the data, and expresses the data in a way that highlights their similarities and differences. Because patterns can be hard to find in data of high dimensions, PCA is a powerful analysis tool. Once patterns in the data are found, the data can be compressed reducing the number of dimensions without a significant loss of information.

Given the data, if each datum has N features represented for instance by x11 x12 … x1N , x21 x22….x2N, the data set can be represented by a matrix Xn×m.

The average observation is defined as

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (10)$$

The deviation from the average is defined as

$$\Phi_i = x_i - \mu \qquad (11)$$

The sample covariance matrix of the data set is defined as

$$c = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T = \frac{1}{n}\sum_{i=1}^{n}\Phi_i\Phi_i^T = \frac{1}{n}AA^T \qquad (12)$$

Eigen values and Eigen vectors of the sample covariance matrix C are usually computed by the Singular Value Decomposition. Suppose (λ1, u1), (λ2, u2)…. (λm, um) are m eigenvalue-eigenvector pairs of the sample covariance matrix C. The k eigenvectors having the largest eigenvalues are selected. The dimensionality of the subspace k can be determined by

$$\frac{\sum_{i=1}^{k}\lambda i}{\sum_{i=1}^{m}\lambda i} \geq \alpha \qquad (13)$$

Where α is the ratio of the variation in the subspace to the total variation in the original space. A m×k matrix U is formed

whose columns consist of the k eigenvectors. The representation of the data by principal components consist of projecting the data onto the k-dimensional subspace according to the following rules. PCA is applied to the KDDCUP99 data and the features selected are shown in tables 15. Tables 16 to 21 show the performance of various neural network classifiers using 13 features

Table 15: Best 13 features selected after principal component analysis

S.no	Feature	Description
0	duration	Continuous
1	flag	Symbolic
2	src_bytes	Continuous
3	dst_bytes	Continuous
4	land	Symbolic
5	wrong_fragment	Continuous
6	urgent	Continuous
7	num_failed_logins	Continuous
8	logged_in	Continuous
9	dst_host_serror_rate	Continuous
10	dst_host_srv_serror_rate	Continuous
11	dst_host_rerror_rate	Continuous
12	dst_host_srv_rerror_rate	Continuous

Table 16: Confusion Matrix for Gaussian mixture IDS (13 features) CPE = .2776 FP=.18 FN=10.32

Predicted ACTUAL	NORMAL	PROBE	DOS	U2R	R2L	% correct
Normal	60019	383	180	5	6	99.05
Probe	195	3876	95	0	0	93.03
DOS	17945	8972	202922	9	5	88.28
U2R	155	25	1	42	5	18.42
R2l	13825	625	5	30	1704	10.5
%correct	65.13	27.92	99.86	48.83	99.06	

Table 17: Confusion Matrix for RBF IDS (13 features) CPE = .3805 FP=.20 FN=14.40

Predicted Actual	Normal	Probe	DOS	U2R	R2L	%correct
Normal	59940	313	330	8	2	98.92
Probe	450	3704	8	3	1	88.91
DOS	36975	20116	172742	15	5	75.15
U2R	196	20	2	10	0	4.38
R2l	7294	7820	200	0	875	5.4
%correct	57.16	11.58	99.68	27.77	99.09	

Table 18: Confusion Matrix for SOM IDS (13 features) CPE = .3805 FP=.20 FN=14.40

Predicted Actual	Normal	Probe	DOS	U2R	R2L	% correct
Normal	56820	3508	255	8	2	93.77
Probe	1270	2549	340	5	2	61.18
DOS	7843	1025	220980	4	1	96.13
U2R	101	68	9	49	1	21.49
R2l	11391	3019	7	0	1772	10.94
%correct	73.38	25.06	99.72	74.24	99.66	

Table 19: Confusion Matrix for Binary Tree Classifier IDS (13 features) CPE=.1837 FP=0.61 FN=5.11

Predicted Actual	Normal	Probe	DOS	U2R	R2L	%correct
Normal	58683	1300	600	7	3	96.84
Probe	564	3102	492	6	2	74.45
DOS	7174	1003	221671	4	1	96.44
U2R	97	99	2	29	1	12.71
R2l	8063	1000	7054	0	72	0.44
%correct	78.68	47.69	96.45	64.04	91.13	

Table 20: Confusion Matrix for ART IDS (13 features) CPE=.2198 FP=.58 FN=5.49

Predicted Actual	Normal	Probe	DOS	U2R	R2L	%correct
Normal	58775	808	1008	1	1	96.99
Probe	136	3971	58	0	1	95.31
DOS	4210	2756	222883	3	1	96.96
U2R	57	126	4	41	0	17.98
R2l	12390	1560	454	1	1784	11.01
%correct	77.77	43.06	99.32	89.13	99.83	

Table 21: Confusion Matrix for LAMSTAR IDS (13 features) CPE=.2198 FP=.58 FN=5.49

Predicted Actual	Normal	Probe	DOS	U2R	R2L	%correct
Normal	60411	140	37	4	1	99.69
Probe	42	4118	5	0	1	98.84
DOS	1688	146	228015	3	1	99.20
U2R	99	54	5	69	1	30.26
R2L	7519	985	1020	0	6665	41.16
%correct	86.68	75.03	99.53	90.04	99.94	

3. IMPLEMENTATION

We focus our research on Misuse based Intrusion detection system using Neural network classifiers. Anomaly based systems are not suitable for Network environment, hence we focus on Misuse based systems. Misuse detection systems detect known attacks using priori defined attack patterns and signatures. We consider Six Neural Network classifiers (Gmix ,RBF, Binary tree, LAMSTAR, SOM, ART) in this paper. We tested the classifiers with three different reduced feature (LDA,ICA,PCA) KDDCUP99 dataset.

3.1 Gaussian Mixture

The Gaussian Mixture classifier (Nguyen, 2006) can perform better than a Gaussian classifier when the classifier distributions are not unimodal Gaussian. Different simulations were performed by changing various parameters: each class has its own Gaussian mixture, all classes share a single set of tied Gaussian mixtures, diagonal covariance, full matrices covariance, separate variance for each Gaussian distribution(?).

Table 22: Comparison of detection rate, false alarm rate, training time and testing time of various classifiers (17features)

	Cost/Example	FP		Normal	Probe	DOS	U2R	R2L
GMIx	0.3309	0.18	DR	99.1	93.2	88.3	22.8	10.6
			FAR	34.8	72.0	0.14	45.8	0.92
			Training Time	37s	15s	50s	7s	12s
			Testing Time	27s	11s	39s	7s	14s
RBF	0.4730	0.37	DR	98.1	85.3	74.3	7.9	5.5
			FAR	43.6	89.3	0.50	59.1	0.89
			Training Time	37s	15s	48s	9s	13s
			Testing Time	29s	11s	39s	8s	12s
BINARYTREE	0.2283	0.60	DR	96.9	74.3	95.7	11.4	0.54
			FAR	22.4	57.2	3.6	38.1	7.37
			Training Time	33s	14s	47s	8s	15s
			Testing Time	28s	15s	27s	9s	14s
LAMSTAR	0.1304	0.11	DR	99.4	95.6	98.6	31.6	37.7
			FAR	15.0	25.5	0.86	8.87	0.07
			Training Time	39s	15s	49s	10s	15s
			Testing Time	27s	15s	24s	8s	12s
SOM	0.2436	1.23	DR	93.6	60.5	96.0	20.6	10.9
			FAR	27.0	75.5	0.30	26.6	0.34
			Training Time	39s	15s	47s	8s	15s
			Testing Time	27s	13s	23s	7s	11s
ART	0.2295	0.59	DR	96.9	96.0	96.9	17.1	11.1
			FAR	22.2	56.8	0.7	11.4	0.17
			Training Time	38s	14s	48s	5s	12s
			Testing Time	25s	13s	28s	4s	9s

3.2 Basis Function

Radial Basis Function classifiers (Selvan and Venkatachalam, 2007) calculate discriminant functions using local Gaussian functions. A total of six simulations were performed using the RBF algorithm. Each simulation used initial clusters created using the K-means algorithm: there were 8,16,32,40,64 and 75 clusters each in different output classes. Weights are trained using the least-square matrix inversion method to minimize the squared error of the output sums, given the basis function outputs for the training patterns. During training and testing variances are increased to provide good coverage of the data. For each RBF simulation, the cost per example for the test dataset was calculated.

3.3 SOM

For SOM (Lippmann, 2003) the training algorithm can be summarized in four basic steps. Step 1 initializes the SOM before training. Step 2, identifies the best matching unit is determined. The best matching unit (BMU) is the neuron, which is the most similar to the input pattern. Step 3 adjusts the best matching neuron (or unit) and its neighbours so that the region surrounding the best matching unit more closely represents the input pattern. This training process continues until all the input vectors are processed. The convergence criterion utilized here is expressed in terms of training epochs, and defines how many times all the input vectors should be fed to the SOM for training purposes.

3.4 Binary Tree classifier

The binary decision tree classifier[15] trains and tests very quickly. It can also be used to identify the input features, which are most important for classification because feature selection is part of the tree-buliding process. Two different training options were used 1. Expansion of the tree until no more errors are found. 2. Early interruption of expansion. Two different testing options were used 1. Full tree for testing, 2.Maximum number of nodes during testing

3.5 ART

The stability and plasticity of ART (Shyu et al., 2003) nets and their ability to cluster input patterns based on their user-controlled mutual similarity, made such nets more appropriate for using in IDSs, rather than the other types of unsupervised nets including SOM, for classifying network traffic into normal and intrusive attack. Accordingly, we used two types of unsupervised ART nets, ART-1 and ART-2. For ART1 and ART2 the optimum value for the vigilance parameter and the number of epochs determine performance.

3.6 LAMSTAR

LAMSTAR stores the information in neurons as well as the correlation links created during training, which make LAMSTAR more suitable for IDs. Using the LAMSTAR (Terrran and Brodley, 1999; Wenke and Stolfo, 2000) algorithm, different clusters were specified and generated for each output class. Simulations were run having 2,4,8,16,32,40,64 clusters. Clusters were trained until the average squared error difference between two epochs was less than 1%.

4. RESULTS AND DISCUSSIONS

Tables 22, 23 and 24 show the comparison of the detection rate, false positive rate, and cost per example for different classifiers using 17-feature, 12-feature and 13-feature datasets. The results obtained show that the detection rate of various classifiers, when applied to the different classes of three different reduced KDDCUP99 datasets, has only a minor variation. The LDA, ICA and PCA show almost the same performance when the detection rate is considered, whereas there is a significant change in training time, testing time, cost per example, and false positive values. The features selected by Principal Component Analysis, when applied to the various classifiers give higher performance than the features selected by LDA and ICA. The detection rate performance of all the classifiers using the 13 features selected by PCA is almost the same on the 41-feature dataset, whereas the training time and testing time are significantly reduced. The performances of the LAMSTAR neural network for all the classes are higher when compared to other classifiers. The 13-feature dataset significantly reduces the training time of the LAMSTAR neural network, due to the reduced number of computations required, without degrading the detection rate. The LAMSTAR IDS gives the highest detection rate with the lowest cost per example.

Table 23: Comparison of detection rate, false alarm rate, training time and testing time of various classifiers (12 features)

	Cost/Exmple	FP		Normal	Probe	DOS	U2R	R2L
GMIx	0.3417	0.34	DR	98.2	89.4	87.4	18.4	9.29
			FAR	35.5	76.3	0.27	51.2	1.06
			Training Time	29s	13s	45s	5s	9s
			Testing Time	24s	8s	35s	4s	9s
RBF	0.4730	0.38	DR	98.0	84.3	74.3	6.57	5.40
			FAR	43.6	88.4	0.52	63.4	0.91
			Training Time	32s	10s	41s	5s	10s
			Testing Time	26s	9s	32s	5s	8s
BINARYTREE	0.2309	0.67	DR	96.5	70.8	95.7	11.4	0.43
			FAR	22.68	57.7	3.65	39.5	8.98
			Training Time	28s	11s	39s	5s	8s
			Testing Time	22s	9s	21s	5s	8s
LAMSTAR	0.1470	0.14	DR	99.3	88.9	98.5	26.3	36.4
			FAR	15.3	29	1.42	10.4	0.07
			Training Time	35s	12s	45s	7s	12s
			Testing Time	22s	10s	21s	5s	8s
SOM	0.2485	1.22	DR	93.0	59.4	95.7	19.3	10.7
			FAR	27.5	76.8	0.37	29.0	0.35
			Training Time	36s	13s	44s	6s	11s
			Testing Time	25s	13s	23s	7s	9s
ART	0.2295	0.63	DR	96.7	93.6	96.8	15.8	10.6
			FAR	22.6	58.6	0.77	12.2	0.18
			Training Time	35s	14s	46s	5s	11s
			Testing Time	22s	11s	24s	4s	8s

Table 24: Comparison of detection rate, false alarm rate, training time and testing time of various classifiers (13 features)

	Cost/Example	FP		Normal	Probe	DOS	U2R	R2L
GMIx	0.2776	0.18	DR	99.0	93.0	88.3	18.4	10.5
			FAR	34.9	72.0	0.14	51.2	0.94
			Trainig Time	30s	12s	45s	4s	9s
			Testing Time	23s	8s	36s	4s	10s
RBF	0.3805	0.20	DR	98.9	88.9	75.1	4.38	5.40
			FAR	42.8	88.4	0.32	72.2	0.91
			Trainig Time	31s	11s	42s	5s	9s
			Testing Time	25s	8s	33s	5s	8s
BINARYTREE	0.1837	0.61	DR	96.8	74.4	96.4	12.7	0.44
			FAR	21.3	52.3	3.55	36.0	8.9
			Trainig Time	29s	11s	40s	5s	9s
			Testing Time	23s	10s	22s	5s	8s
LAMSTAR	0.1030	5.85e-4	DR	99.7	98.9	99.2	30.3	41.2
			FAR	13.3	25.0	0.47	9.96	0.06
			Trainig Time	36s	12s	46s	6s	12s
			Testing Time	23s	11s	22s	5s	8s
SOM	0.2404	1.21	DR	93.8	61.2	96.1	21.5	10.9
			FAR	26.6	74.9	0.28	25.8	0.34
			Trainig Time	37s	14s	47s	6s	11s
			Testing Time	26s	13s	24s	7s	10s
ART	0.2198	0.18	DR	97.0	95.3	97.0	18.0	11.0
			FAR	22.2	56.9	0.68	10.9	0.17
			Trainig Time	35s	14s	46s	5s	11s
			Testing Time	24s	12s	26s	4s	8s

Training time is also significantly reduced when PCA is used as a feature reduction technique, hence we propose to use the LAMSTAR IDS with PCA for our further research. PCA performs well because the features selected have high information gain.

5. CONCLUSION

Feature reduction techniques like Principal Component Analysis, Independent Component Analysis, and Linear Discriminant Analysis are applied to the KDDCUP99 data set to reduce its features. PCA selects 13 features, ICA selects 12 features, and LDA selects 17 features. The reduced features are used as input to different classifiers and the results are compared. The results show that the performance with 13 features, 12 features, and 17 features are comparable to the 41 features', with reduced training and testing times. Comparing these three algorithms, PCA gives better detection rate, false positive, cost per example, training and testing times than ICA and LDA. Comparing the various classifiers used, the LAMSTAR neural network shows better performance for all the classes with comparable training and testing times when 13 features selected by PCA are used as input data. The KDDCUP99 reduced dataset obtained with PCA shows promising results. Hence, we propose to consider PCA for our further research. To further improve performance we propose to reduce the numbers of samples in the PCA reduced dataset. This is achieved by means of a sample selection algorithm, which uses clustering to select the samples with high information gain.

REFERENCES

Aapo Hyvärinen and Erkki Oja, (2000) Independent Component Analysis : Algorithms and Applications, Neural Networks, Volume 13 , Issue 4-5 pp 411 – 430.

Balakrishnama. A. Ganapathiraju, (1998) Linear Discriminant Analysis - A Brief Tutorial, Institute for Signal and Information Processing, Department of Electrical and Computer Engineering, Mississippi State University.

Berchtold, S., C. Böhm, H.-P. Kriegel, (1998) The pyramid- technique:towards breaking the curse of dimensionality. In: SIGMOD, pp. 142–153.

Fodor I.K. (2002). A survey of dimension reduction techniques. Technical Report UCRL-ID-148494, U.S. Department of Energy/Lawrence Livermore National Laboratory. .

Gopi K. Kuchimanchi, Vir V. Phoha, Kiran S. Balagani, Shekhar R. Gaddam, (2004) Dimension Reduction Using Feature Extraction Methods for Real-time Misuse Detection Systems, Proceedings of the 2004 IEEE Workshop on Information Assurance and Security T1B2 1555 United States Military Academy, West Point, NY,.

Graupe D., (1997) Principles of Artificial Neural Networks, pp. 191-222, World Scientific Publishing Co. Pte. Ltd., Singapore.

http://kdd.ics.uci.edu//databases/kddcup99/
kddcup99. html.

Jing Gao et al., (2006) "A Novel Framework for Incorporating Labeled Examples into Anomaly Detection", Proceedings of the Siam Conference on Data Mining.

Jolliffe I. T. (2002) "Principal Component Analysis", Springer Verlag, New York, NY, third edition. .

Khaled Labib and Rao Vemuri, (2002) " NSOM: A Real-Time Network-Based Intrusion Detection System Using Self-Organizing Maps " , Networks and Security, 21(1).

Kordylewski H., (1998) "A Large Memory Storage and Retrieval Neural Network for Medical and Engineering Diagnosis/Fault Detection ", Doctor of Philosophy's Thesis, University of Illinois at Chicago, TK-99999-K629.

Morteza Amini et al., (2004) "Network-Based Intrusion Detection Using Unsupervised Adaptive Resonance Theory (ART)", Published in the Proceedings of the 4th Conference on Engineering of Intelligent Systems (EIS 2004), Madeira, Portugal.

Nguyen D., A. Das, G. Memik, and A. Choudhary, (2006) "Reconfigurable Architecture for Network Intrusion Detection Using Principal Component Analysis" In Proc. ACM/SIGDA 14th international symposium on Field programmable gate arrays , pp. 235 – 235.

Lippmann R., (2003) "Passive Operating System Identification From TCP/IP Packet Headers" published in the Proceedings of the Workshop on Data Mining for Computer Security (DMSEC), Lincoln Laboratory ,Massachusetts.

Selvan S., V. Venkatachalam, (2007) Intrusion Detection using an Improved Competitive Learning Lamstar Neural Network, IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.2, pp 255-263, February 2007, ISSN 1708-7906.

Shyu M.-L., S.-C. Chen, K. Sarinnapakorn, and L. Chang, (2003) "A novel anomaly detection scheme based on principal component classifier", In Proc. IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third

IEEE International Conference on Data Mining (ICDM'03), pp 172–179.

Lane T. and C.E. Brodley (1999) Temporal Sequence Learning and Data Reduction for Anomaly Detection, ACM Transactions on Information and System Security, Vol. 2, No. 3, August 1999, Pages 295–331.

Wenke. L. and S. J. Stolfo, (2000) A Framework for Constructing Features and Models for Intrusion Detection Systems," S. ACM Transactions on Information and System Security, vol. 3, pp. 227.

AUTHOR BIOGRAPHIES

Srinivasan Selvan (M'87-SM'95) received the B.E degree in electronics and communication engineering and the M.E degree in communication systems from the University of Madras, Chennai, India,in 1977 and 1979, respectively, and the Ph.D. degree in computer science and engineering from the Madurai Kamaraj University, Madurai, India in 2001. He has 28 years of teaching experience. He is currently the principal of St Peters Engineering College, Chennai, India. He has published more than 100 papers in international and national journals and conference proceedings. His areas of research include intrusion detection systems, digital image processing, soft computing, digital signal processing, computer networks and data mining

V. Venkatachalam received the B.E. degree in Electronics and Communication from Bharathiyar University and M.S. degree in software systems from Birla Institute of Technology. He received M. Tech. Degree in Computer Science from National Institute of Technology. His Research interest includes Network Security and Pattern recognition. He is currently pursuing his PhD degree in Network Security. Presently working as Head of the Dept. CSE in Erode Sengunthar Engineering College.