CLUSTERING AND SAMPLE SELECTION TO ENHANCE THE PERFORMANCE OF THE LAMSTAR INTRUSION DETECTION SYSTEM

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 the present work, it is proposed to enhance the learning capabilities and reduce the training time of a competitive learning LAMSTAR neural network using Clustering and Sample Selection algorithm. KDDCUP99 reduced feature data set (Features reduced by PCA algorithm) is used for training and testing the various classifiers. KDDCUP99 dataset has five classes, DOS, PROBE, NORMAL, U2R, and R2L. The DOS class and Normal class have huge records, which in turn increase the training time of the classifiers. Sample records with high information gain are selected from KDDCUP99 DOS and Normal class using Clustering and sample selection algorithm. The resulting dataset is used to train the LAMSTAR intrusion detection system and its performance compared to other Intrusion Detection classifiers. The results obtained show that the proposed technique performs well in terms of detection rate and training time compared to the results obtained by the classifiers using the full dataset.

*Keywords***:** Intrusion Detection, LAMSTAR, SOM, Fuzzy-C-Means , Wavecluster.

1. INTRODUCTION

Intrusion detection is the process of monitoring computer networks and systems for the recognition of security policy violations. Intrusion detection augments the traditional audit, which was designed to occur at infrequent intervals, thus, making it a continuous process. Anomaly and misuse detection are two major areas of research in Intrusion Detection Systems. Misuse detection systems encode and match the sequence of “signature actions” of known intrusion scenarios. Anomaly detection systems establish normal usage patterns (profiles) using statistical measures on system features. Misuse Detection research is gaining importance due to the growing needs of network connectivity. A major issue in Misuse Detection research is the processing of a huge amount of data. The KDDCUP99 (URL: http://kdd.ics.cui.edu,99) dataset is a well known dataset typically used by researchers in this field.

The KDDCUP99 dataset has five classes, DOS, PROBE, NORMAL, U2R, and R2L. The DOS class and Normal class include huge records, which in turn increase the training time of the classifiers. In the present work, it is proposed to enhance the learning capabilities and reduce the training time of a competitive learning LAMSTAR neural network using Clustering and Sample Selection algorithm. The KDDCUP99 reduced feature data set (Features reduced by PCA algorithm) is used for training and testing the various classifiers. Sample records with high information gain are selected from KDDCUP99 DOS and Normal class using Clustering and sample selection algorithm. The resulting dataset is used to train the LAMSTAR intrusion detection system whose performance is compared with other Intrusion Detection classifiers.

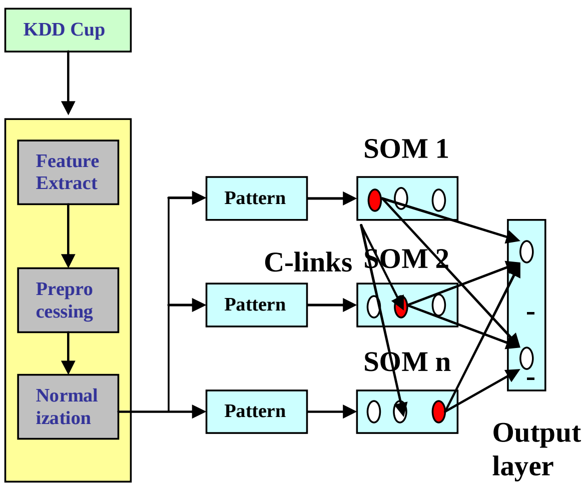

Figure 1: LAMSTAR IDS design

2. LAMSTAR IDS DESIGN

A modified LAMSTAR network used for intrusion detection is as shown in figure 1. The model reads in data from the KDDCUP99 set and sends it first to the feature extraction module, which extracts 41 features of the data for pre-processing by the designated module. The pre-processing module converts the 41 features into a standardized numeric representation. Normalization block reads the preprocessed data and normalizes the data into a format required by the SOM's. The normalized input pattern was split into sub patterns (basic features 9, content features 13, traffic 9, and others 10) (URL:http://kdd.ics.cui.edu,99). Each sub pattern is assigned to one SOM module. This SOM configuration yields very rapid matching with good error tolerance, and is capable of generalization.

Between SOM modules, connections are established using correlation links. The correlation links distribute information between various modules. The training data contains 22 attack patterns and normal patterns. The SOM modules are trained using this pattern. The coordinated activation of neurons between the various modules allows the network to detect intrusions. Detailed explanation of LAMSTAR IDs, Dataset KDDCUP99 used for evaluation, preprocessing, Normalization and performance metric used are available in (Venkatachalam et al., 2007)

3. KDDCUP99 DATA REDUCTION USING CLUSTERING

In the KDDCUP99 dataset the DOS class and the Normal class contain large number of records. The number of records in this classes are reduced using clustering and sample selection algorithms which in turn reduce the training and testing time without reducing the performance of the Intrusion detection system. KDDCUP99(PCA reduced data 13 features) (Nguyen et al., 2006; Shyu et al., 2003; Jolliffe, 2002) dataset is tested using SOM Clustering, Fuzzy C-means Clustering and Wavelet Clustering.

4. SELF-ORGANIZING MAP CLUSTERING

Self Organizing Map (SOM) (Lichodzijewski et al., 2002; Kohonen, 1998) by Teuvo Kohonen provides a data visualization technique as shown in figure 2, which helps to understand high dimensional data by reducing the dimensions of data to a map. SOM also represents a clustering concept to group similar data together. Therefore it can be said that SOM reduces data dimensions and displays similarities among data.

With SOM, clustering is performed by having several units compete for the current object. Once the data have been entered into the system, the network of artificial neurons is trained by providing information about inputs. The weight vector of the unit closest to the current object becomes the winning or active unit. During the training stage, the values for the input variables are gradually adjusted in an attempt to preserve neighborhood relationships that exist within the input data set. As the values of the input variables get closer to the input object, the weights of the winning unit are adjusted as well as its neighbors.

4.1 Data Similarity

Getting the Best Matching Unit is done by running through all right vectors and calculating the distance from each weight to the sample vector. The weight with the shortest distance is the winner. There are numerous ways to determine the distance, however, the most commonly used method is the Euclidean Distance

4.2 SOM algorithm

Each data from KDDCUP99 reduced feature dataset competes for representation in the SOM map. SOM mapping steps start from initializing the weight vectors. From there a sample vector is selected randomly and the map of weight vectors is searched to find which weight best represents that sample. Each weight vector has neighboring weights that are close to it. The weight that is chosen is rewarded by being able to become more like the randomly selected sample vector. SOM mapping is as shown in figure 2.

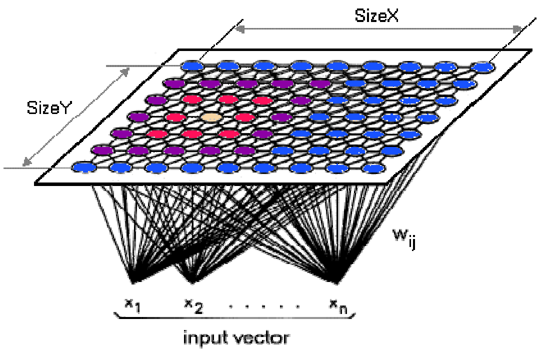

Figure 2: SOM Learning

The neighbors of that weight are also rewarded by being able to become more similar to the chosen sample vector. From this step the number of neighbors and how much each weight can learn decrease over time. This whole process is repeated a large number of times, usually more than 1000 times. In sum, learning occurs in several steps and over many iterations. :

1. The weights of all nodes are initialized.
2. A vector is chosen at random from the set of training data.
3. Every node is examined to identify the one whose weights are most similar to the input

vector. The winning node is commonly known as the Best Matching Unit (BMU).

4. Then the neighbourhood of the BMU is calculated. The amount of neighbors decreases over time.
5. The winning weight is rewarded with a pre determined value so that it looks like the sample vector. The neighbors also become more like the sample vector. The closer a node is to the BMU, the more its weights get altered and the farther away the neighbor is from the BMU, the less it learns.
6. Repeat step 2 for N iterations.

5. FUZZY C-MEANS CLUSTERING

In fuzzy clustering (Jonatan Gomez et al., 2001; Lampienen et al., 2002), each point has a degree of

$$\forall x \qquad \sum_{k=1}^{num.clusters} u_k(x) = 1 \tag{1}$$

membership in each cluster, as in fuzzy logic, rather than belonging completely to just one cluster.

Thus, points on the edge of a cluster, may be in the cluster to a lesser degree than points in the center of cluster. For each point x we have a coefficient

$$center_k = \frac{\sum_x u_k(x)^m x}{\sum_x u_k(x)^m} \tag{2}$$

giving the degree of membership in the kth cluster uk(x). Usually, the sum of those coefficients is defined to be 1.

With fuzzy c-means, the centroid of a cluster is the mean of all points, weighted by their degree of membership in the cluster. The degree of membership is related to the inverse of the distance to the cluster

$$u_k(x) = \frac{1}{d\ (center_k, x)}, \tag{3}$$

The coefficients are normalized and fuzzyfied with a real parameter *m* > 1 so that their sum is 1.

$$u_k(x) = \frac{1}{\sum_j \left(\frac{d\ (center_k, x)}{d\ (center_j, x)} \right)^{\frac{1}{(m-1)}}} \tag{4}$$

So for m equal to 2, this is equivalent to normalising the coefficients linearly to make their sum equal to 1. When m is close to 1, then the cluster center closest to the point is given much larger weight than the others, and the algorithm is similar to k-means.

The fuzzy c-means algorithm is very similar to the k-means algorithm:

- Choose a number of clusters.
- Assign randomly to each point coefficients of membership in the clusters.
- Repeat until the algorithm has converged (that is, the coefficients' change between two iterations is not larger than ε, the given sensitivity threshold) :
- Compute the centroid for each cluster, using the formula above.
- For each point, compute the coefficients of membership in the clusters, using the formula above.

The algorithm minimizes intra-cluster variance as well, but shows the same problems as k-means, the minimum is a local minimum, and the results depend on the initial choice of weights. The Expectation-maximization algorithm is a more statistically formalized method, which accounts for partial membership in classes. It has better convergence properties and is in general preferred to fuzzy-k-means.

6. WAVECLUSTER

The multi-resolution property of wavelet transforms inspires the researchers to consider algorithms that could identify clusters at different scales. WaveCluster (Gholamhosien et al., 1998 and 2000) is a multi-resolution clustering approach for very large spatial databases. Spatial data objects can be represented in an n-dimensional feature space and the numerical attributes of a spatial object can be represented by a feature vector where each element of the vector corresponds to one numerical attribute (feature). Partitioning the data space by a grid reduces the number of data objects while inducing only small errors. From a signal processing perspective, if the collection of objects in the feature space is viewed as an n-dimensional signal, the high frequency parts of the signal correspond to the regions of the feature space where there is a rapid change in the distribution of objects (i.e. the boundaries of clusters) and the low frequency parts of the n dimensional signal, which have high amplitude correspond to the areas of the feature space where the objects are concentrated (i.e., the clusters). Applying wavelet transform on the data decomposes it into different frequency sub-bands. Hence identifying the clusters is then converted to finding the connected components in the transformed feature space. Moreover, the application of wavelet transformation to feature spaces provides multiresolution data representation and hence the search for the connected components could be carried out at different resolution levels. In other words, the multi-resolution property of wavelet transforms enable the WaveCluster algorithm to effectively identify arbitrary clusters at different scales with different degrees of accuracy.

7. SAMPLE SELECTION ALGORITHM

After clustering the data, samples having high information gain are selected using the algorithm described below. Three different clustering algorithms SOM, Fuzzy- C-means and Wavecluster are used to cluster the data. The sample selection algorithm is applied to the clustered data separately and the results are compared. The algorithms are applied to the KDDCUP99 DOS class data and NORMAL class data, since these two classes include very large numbers of records.

The input classes are clustered separately in such a way as to produce a new dataset composed with the centroid of each cluster, and a set of boundary samples selected according to their distance from the centroid. The flow chart is as shown in figure 3.

Two parameters the number of fixed clusters K and Number of samples to be selected S are given as input parameter to the sample selection algorithm. The samples are taken from the clustered data according to their relative intra-class variance and their density (Matthias et al., 2003). The coverage factor is calculated using the intra-class variance and density for the DOS and NORMAL class separately. The number of samples taken from a given cluster is proportional to the computed coverage factor.

The relative variance of each cluster is computed using the following expression:

$$V_r(cl_i) = \frac{\frac{1}{Card(cl_i)} * \sum_{x \in cl_i} dist(x, c_i)}{\sum_{j=1}^{k} \left(\frac{1}{Card(cl_j)} * \sum_{x \in A} dist(x, c_j) \right)}$$

Cli –clusteri (5)

Where Card(X) gives the cardinality of a given set X, and dist(x,y) gives the distance between the two points x and y. Generally, the distance between two points is taken as a common metric to assess the similarity among the components of a samples set. The density value corresponding to the same cluster cli is computed as follows:

$$Den(cl_i) = \frac{Card(cl_i)}{Card(A)} \quad (6)$$

The coverage factor is then computed using equation

$$Cov(cl_i) = \frac{(Vr(cl_i) + Den(cl_i))}{2} \quad (7)$$

we can clearly see that: 0 ≤ Vr(cli) ≤ 1 and 0 ≤ Den(cli) ≤ 1 for any cluster cli. So the coverage factor Cov(cli) belongs also to the [0,1] interval. Furthermore, it is clear that:

$$\sum_{i=1}^{k} Vr(cl_i) = 1 \quad and \quad \sum_{i=1}^{k} Den(cl_i) = 1 \quad (8)$$

We can easily see that

$$\sum_{i=1}^{k} Cov(cl_i) = 1 \quad (9)$$

Hence the number of samples selected from each cluster is determined using the following expression: Num_samples(cli)= Round(S*Cov(cli))

7.1 Sample Selection Using SOM Clustering algorithm

SOM clustering is applied to the KDDCUP reduced feature data. The sample selection algorithm has two input parameters, K the number of clusters and S the number of samples to be selected from each cluster. Different values for K and S are given and the detection rate and training time are calculated. The best results are obtained with K=10 for the DOS class and K= 7 for the normal class. The results obtained with SOM are shown in tables 1, 2 and 3

Table 1: Detection rate and Training Time comparison for 5% of the data (DOS 19573, Normal 4864)

Classifier	Detection Rate		Training Time	
	DOS	Normal	DOS	Normal
Gaussian Mixture	86.30	96.40	15s	9s
Binary Tree	94.70	94.90	17s	8s
ART	86.30	96.40	15s	9s
LAMSTAR	98.90	97.10	18s	10s

Table 2 : Detection rate and Training Time comparison for 10% of the data (DOS 39145, Normal 9727)

Classifier	Detection Rate		Training Time	
	DOS	Normal	DOS	Normal
Gaussian Mixture	87.34	97.30	19s	11s
Binary Tree	95.38	95.40	19s	10s
ART	90.12	89.10	21s	13s
LAMSTAR	99.06	99.12	18s	11s

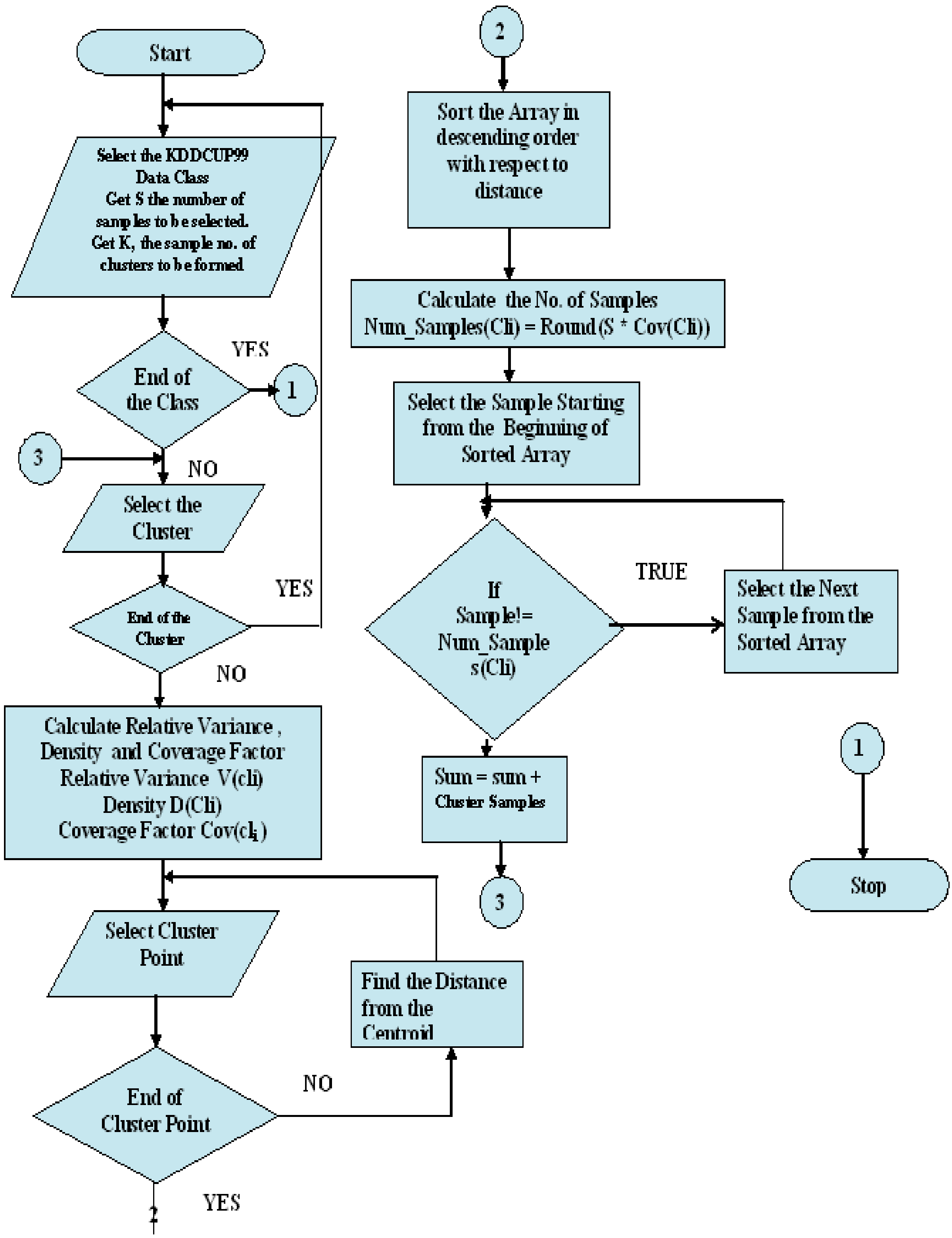

Figure 3 : Flow chart of the Sample Selection algorithm

Table 3 : Detection rate and Training Time comparison for 20% of the data (DOS 78390, Normal 19454)

Classifier	Detection Rate		Training Time	
	DOS	Normal	DOS	Normal
Gaussian Mixture	88.1	98.30	25s	16s
Binary Tree	95.70	96.40	27s	14s
ART	88.1	98.30	25s	16s
LAMSTAR	99.20	99.10	23s	15s

7.2 Sample Selection Using Fuzzy C-means clustering algorithm

Fuzzy C-means clustering is applied to the KDDCUP99 data with different values for K and S. The best results are obtained with K=9 for the DOS class and K= 7 for the normal class. The results are shown in tables 4, 5 and 6.

Table 4: Detection rate and Training Time comparison for 5% of the data (DOS 19573, Normal 4864)

Classifier	Detection Rate		Training Time	
	DOS	Normal	DOS	Normal
Gaussian Mixture	85.40	95.10	17s	11s
Binary Tree	94.10	94.20	18s	10s
ART	86.20	84.45	20s	11s
LAMSTAR	98.60	97.20	18s	12s

Table 5 : Detection rate and Training Time comparison for 10% of the data (DOS 39145, Normal 9727)

Classifier	Detection Rate		Training Time	
	DOS	Normal	DOS	Normal
Gaussian Mixture	86.30	97.20	21s	13s
Binary Tree	94.54	94.82	22s	12s
ART	89.20	87.10	23s	13s
LAMSTAR	99.05	99.14	20s	12s

Table 6 : Detection rate and Training Time comparison for 20% of the data (DOS 78390, Normal 19454)

Classifier	Detection Rate		Training Time	
	DOS	Normal	DOS	Normal
Gaussian Mixture	87.2	97.20	27s	17s
Binary Tree	94.70	95.70	27s	16s
ART	91.10	89.20	28s	18s
LAMSTAR	99.10	99.00	25s	16s

7.3 Sample selection using Wavecluster clustering algorithm

Wavecluster is applied to the KDDCUP99 data. The best results are obtained with K=9 for the DOS class and K= 6 for the normal class. The results are shown in tables 7,8 and 9.

The comparison of detection rate and training time of the three clustering algorithms shows that the wave cluster gives the best detection rate and training time.

Table 7 : Detection rate and Training Time comparison for 5% of the data (DOS 19573, Normal 4864)

Classifier	Detection Rate		Training Time	
	DOS	Normal	DOS	Normal
Gaussian Mixture	86.60	96.10	16s	10s
Binary Tree	95.50	96.00	17s	9s
ART	84.30	86.10	18s	12s
LAMSTAR	98.90	98.20	16s	11s

Table 8: Detection rate and Training Time comparison for 10% of the data (DOS 39145, Normal 9727)

Classifier	Detection Rate		Training Time	
	DOS	Normal	DOS	Normal
Gaussian Mixture	88.30	98.20	19s	11s
Binary Tree	96.53	96.73	20s	10s
ART	92.32	90.48	20s	12s
LAMSTAR	99.48	99.54	19s	10s

Table 9 : Detection rate and Training Time comparison for 20% of the data (DOS 78390, Normal 19454)

Classifier	Detection Rate		Training Time	
	DOS	Normal	DOS	Normal
Gaussian Mixture	88.2	98.20	25s	15s
Binary Tree	95.70	96.40	25s	14s
ART	92.12	90.78	28s	16s
LAMSTAR	99.48	99.53	23s	14s

8. RESULTS AND DISCUSSION

The detection rate and training time of the various classifiers, using three different clustering methods, and a sample selection algorithm with 5 % of the data selected are shown in tables 10 and 11, respectively. The results obtained show that with only 5% of the data, the performance of all the classifiers are good except for RBF whose detection rate is smaller for the DOS class. The training time is also significantly shorter.

Table 10: Detection rate comparison for three different Clustering methods using 5% of the data (DOS 19573, Normal 4864)

Classifier	Detection Rate SOM Clustering		Detection Rate Fuzzy-C-Means		Detection Rate Wave Cluster	
	DOS	Normal	DOS	Normal	DOS	Normal
Gaussian Mixture	86.30	96.40	85.40	95.10	86.60	96.10
Binary Tree	94.70	94.90	94.10	94.20	95.50	96.00
ART	86.30	96.40	86.20	84.45	84.30	86.10
LAMSTAR	98.90	97.10	98.60	97.20	98.90	98.20

Table 11: Training time comparison of three different clustering methods using 5% of the data (DOS 19573, Normal 4864)

Classifier	Training Time SOM Clustering		Training Time Fuzzy-C-Means		Training Time Wave Cluster	
	DOS	Nor mal	DOS	Nor mal	DOS	Nor mal
Gaussian Mixture	15s	9s	17s	11s	16s	10s
Binary Tree	17s	8s	18s	10s	17s	9s
ART	15s	9s	20s	11s	18s	12s
LAMST AR	18s	10s	18s	12s	16s	11s

The detection rate and training time of the various classifiers using 10 % of the data selected are shown in tables 12 and 13, respectively. The results obtained show that the detection rate is increased compared to the results obtained with 5% of the data, but the training time is significantly also increased.

Table 12: Detection rate comparison of three different clustering methods using 10% of the data (DOS 39145, Normal 9727)

Classifier	Detection Rate SOM Clustering		Detection Rate Fuzzy-C-Means		Detection Rate Wave Cluster	
	DOS	Normal	DOS	Normal	DOS	Normal
Gaussian Mixture	87.34	97.30	86.30	97.20	88.30	98.20
Binary Tree	95.38	95.40	94.54	94.82	96.53	96.73
ART	90.12	89.10	89.20	87.10	92.32	90.48
LAMSTAR	99.06	99.12	99.05	99.14	99.48	99.54

Table 13 Training Time comparison of three different clustering methods using 10% of the data (DOS 39145, Normal 9727)

Classifier	Training Time SOM clustering		Training Time Fuzzy-C-Means		Training Time Wave Cluster	
	DOS	Normal	DOS	Normal	DOS	Normal
Gaussian Mixture	19s	11s	21s	13s	19s	11s
Binary Tree	19s	10s	22s	12s	20s	10s
ART	21s	13s	23s	13s	20s	12s
LAMSTAR	18s	11s	20s	12s	19s	10s

The detection rate and training time of the various classifiers using 20 % of the data selected are shown in tables 14 and 15, respectively. The results obtained show that the detection rate is increased compared to the results obtained for the previous cases, but the training time is also significantly increased. Among the three data sets, the 10 % data with wave clustering gives good detection rate as well as short training time, hence we propose to consider 10% data with wavecluster for our further research.

Table 14: Detection rate comparison of three different clustering methods using 20% of the data (DOS 78390, Normal 19454)

Classifier	Detection Rate SOM Clustering		Detection Rate Fuzzy-C-Means		Detection Rate Wave Cluster	
	DOS	Normal	DOS	Normal	DOS	Normal
Gaussian Mixture	88.1	98.30	87.2	97.20	88.2	98.20
Binary Tree	95.70	96.40	94.70	95.70	95.70	96.40
ART	88.1	98.30	91.10	89.20	92.12	90.78
LAMSTAR	99.20	99.10	99.10	99.00	99.48	99.53

Table 15: Training Time comparison of three different clustering methods using 20% of the data (DOS 78390 Normal 19454)

Classifier	Detection Rate SOM Clustering		Detection Rate Fuzzy-C-Means		Detection Rate Wave Cluster	
	DOS	Normal	DOS	Normal	DOS	Normal
Gaussian Mixture	25s	16s	27s	17s	25s	15s
Binary Tree	27s	14s	27s	16s	25s	14s
ART	25s	16s	28s	18s	28s	16s
LAMSTAR	23s	15s	25s	16s	23s	14s

9. CONCLUSION

Clustering and Sample selection algorithms are applied to the KDDCUP99 DOS and Normal class reduced feature dataset. Three different clustering algorithms are used. Different experiments are performed by selecting 5 % ,10 % and 20 % of the data from each clustered data set and applied to various classifiers.

The results obtained show that with only 10 % of the data the detection rate reaches a maximum level and the training time is also maintained at a reasonable level. The Detection rate of the LAMSTAR neural network is significantly higher compared to that of other classifiers. The training time of the LAMSTAR is shorter. Comparing the three different clustering algorithms used, wavecluster gives better performance with shorter training time.

REFERENCES

Gholamhosein.S, Surojit.C, Aidong.Z. (1998). WaveCluster : A Multi-Resolution Clustering Approach for Very Large Spatial Databases, Proceedings of the 24th VLDB conference. pp 428-439 Newyork,USA.

Gholamhosein S, Surojit.C. Aidong.Z. (2000) WaveCluster: A wavelet-based clustering approach for spatial data in very large databases. The VLDB Journal, Volume 8 , Issue 3-4 , pp 289 – 304.

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

Jonatan Gomez and Dipankar Dasgupta.(2001) Evolving Fuzzy Classifiers for Intrusion Detection. Proceedings of the 2002 IEEE Workshop on Information Assurance United States Military Academy, West Point, NY.

Kohonen T. (1998). Self-Organizing and Associative Memory. 2 nd Edition, Springer Vela, N.Y.

Lampinen. T, Koivisto. H , Honkanen. T.(2002) Profiling network applications with fuzzy c-means clustering and self- organizing map. International Conference on Fuzzy Systems and Knowledge Discovery, Computational Intelligence for the E-Age 2 Volumes. November 18-22, Orchid Country Club, Singapore.

Lichodzijewski P, A.N. Zincir Heywood, and M. I. Heywood.(2002). Host-based intrusion detection using self-organizing maps. Proceedings of the 2002 IEEE World Congress on Computational Intelligence. Honolulu, HI, pp. 1714-1719.

Matthias.K, Stefano.L, Gianluca.M.(2003). Distributed Clustering Based on Sampling Local Density Estimates. In Proceedings of International Joint Conference on Artificial Intelligence (IJCAI 2003), pages 485-490, Mexico.

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

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

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

AUTHORS BIOGRAPHIES

Dr. S. Selvan is Professor and Head, Department of Information Technology at PSG College of Technology, Coimbatore, India. He has 27 years of teaching experience. He has published more than 60 papers in international and national journals and conference proceedings. His areas of research include digital image processing, soft computing, digital signal processing and computer networking.

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, Erode.