2013 International Conference on Computer Communication and Informatics (ICCCI -2013), Jan. 04 – 06, 2013, Coimbatore, INDIA

Intrusion Detection Technique by using K-means, Fuzzy Neural Network and SVM classifiers.

A. M. Chandrasekhar.

Department of Computer Science & Engineering,
Sri Jayachamarajendra college of Engineering (SJCE),
Mysore-570006, Karnataka, India.
E-mail: amblechandru@gamil.com.

K. Raghuveer.

Department of Information Science,
National Institute of Engineering (NIE),
Mysore, Karnataka-570008, India.
E-mail: raghunie@yahoo.com.

Abstract — **With the impending era of internet, the network security has become the key foundation for lot of financial and business web applications. Intrusion detection is one of the looms to resolve the problem of network security. Imperfectness of intrusion detection systems (IDS) has given an opportunity for data mining to make several important contributions to the field of intrusion detection. In recent years, many researchers are using data mining techniques for building IDS. Here, we propose a new approach by utilizing data mining techniques such as neuro-fuzzy and radial basis support vector machine (SVM) for helping IDS to attain higher detection rate. The proposed technique has four major steps: primarily, k-means clustering is used to generate different training subsets. Then, based on the obtained training subsets, different neuro-fuzzy models are trained. Subsequently, a vector for SVM classification is formed and in the end, classification using radial SVM is performed to detect intrusion has happened or not. To illustrate the applicability and capability of the new approach, the results of experiments on KDD CUP 1999 dataset is demonstrated. Experimental results shows that our proposed new approach do better than BPNN, multiclass SVM and other well-known methods such as decision trees and columbia model in terms of sensitivity, specificity and in particular detection accuracy.**

Keywords -- Intrusion Detection System; Neuro-fuzzy; Radial Support Vector Machine; K-means Clustering.

I. INTRODUCTION AND MOTIVATION

As defined by SANS institute, Intrusion detection is the art of detecting inappropriate, inaccurate, or anomalous activity. Intrusion detection is the process of monitoring and analyzing the events occurring in a computer system in order to detect signs of security problems. Intrusion detection is an important component of infrastructure protection mechanisms. Intrusion detection system (IDS) is the most essential part of the security infrastructure for the networks connected to the Internet, because of the numerous ways to compromise the stability and security of the network. IDS can be used to monitor computers or networks for unauthorized activities. Particularly, network based IDS analyze the network traffic coming into the network to detect, identify and track the intruders. Intrusion detection can be classified into two types: anomaly detection and misuse detection [1].

978-1-4673-2907-1/13/$31.00 ©2013 IEEE

Intrusion detection system has been an active area of research and development for the past few decades. This is primarily because of the escalating of attacks on computers and on networks in recent years and computerized scrutiny has become a compulsory addition to IT security [2]. Intrusion detection faces a number of challenges; an intrusion detection system must reliably detect malicious activities in a network and must perform efficiently to cope with the large amount of network. Intrusion detection systems are gauged based on its detection precision and detection stability. The majority of currently existing IDS face a number of challenges such as low detection rates and high false alarm rates, which falsely classifies a normal connection as an attack and therefore obstructs legitimate user access to the network resources. These problems are due to the sophistication of the attacks and their intended similarities to normal behavior. In recent years, more intelligence is brought into IDS by means of machine learning. Theoretically, it is possible for a machine learning algorithm to achieve the best performance by maximizing the detection accuracy. However, this normally requires infinite training sample sizes. This motivated the researchers to working to words enhancing the detection precision and stability.

Early in the decade, researchers focused on using rule based expert systems and statistical approaches. But when encountering larger datasets, the results of rule-based expert systems and statistical approaches become worse. Thus a lot of data mining techniques have been introduced to solve the problem. Artificial neural network (ANN) is one of the most widely used data mining and has been successful in solving many complex practical problems due to encounter of large traffic data set. Based on a study of latest research literatures, there are quite a lot of research that attempts to relate data mining and machine learning techniques to the intrusion detection systems so as to design more intelligent intrusion detection model. Currently the support vector learning technique is featuring superior [3].

The rest of the paper is organized as follows: The related research work on IDS is presented in section II. The proposed technique for intrusion detection is illustrated in section III. The detailed experimental setup, data set description,

evaluation criterion, results and comparative study are discussed in section IV. The conclusions are summed up in section V.

II. RELATED RESEARCH WORK ON IDS

In the last decade various approaches have been developed and proposed in order to detect the intrusion. In the early stage, rule-based expert systems and statistical approaches were used to detect intrusion. A rule-based expert IDS can detect some well-known intrusions with high detection rate, but it is difficult to detect new intrusions and its signature database needs to be updated manually and frequently. Statistical-based IDS employs various statistical methods including Principal component analysis, cluster and multivariate analysis, bayesian analysis, and frequency and simple significance tests. But this type of IDS needs to collect enough data to build a complicated mathematical model, which is impractical in the case of complicated network traffic.

To overcome the restrictions of rule-based expert systems and statistical approaches, a number of data mining techniques have been introduced. Among these techniques, ANN is one of the most widely used and has been successfully applied to intrusion detection. Different types of ANNs are used in IDS like supervised, unsupervised, and hybrid ANN-based intrusion detection. Supervised ANN applied to IDS mainly includes multi-layer feed-forward (MLFF) neural networks and recurring neural networks. The MLFF neural networks are easy to reach the local minimum and thus stability is lower. Especially, for low-frequent attacks, the detection precision is very low. Some researchers have compared the effectiveness of supervised ANN with other methods such as support vector machine (SVM) and multivariate adaptive regression splines (MARS). Supervised ANN had been shown to have lower detection performance than SVM and MARS. Using unsupervised ANN in intrusion detection has an advantage of improvising the analysis of new data without retraining. The hybrid ANN combines supervised ANN and unsupervised ANN, or combines ANN with other data mining techniques to detect intrusion. The inspiration for using the hybrid ANN is to prevail over the limitations of individual ANN.

Jirapummin et al. [4] proposed employing a hybrid ANN for both visualizing intrusions using Kohenen's SOM and classifying intrusions using resilient propagation neural networks. Horeis [5] used a combination of SOM and radial basis function (RBF) networks. The system offers generally better results than IDS based on RBF networks alone. Han and Cho [6] proposed an intrusion detection technique based on evolutionary neural networks in order to determine the structure and weights of the call sequences. Chen, Abraham, and Yang [7] proposed hybrid flexible neural-tree-based IDS based on flexible neural tree, evolutionary algorithm and particle swarm optimization (PSO). Empirical results indicated that the proposed method is efficient. For ANN based intrusion detection, hybrid ANN has been the trend (Chen et al.[7]). But, different ways to construct hybrid ANN will highly influence the performance of intrusion detection. different hybrid ANN models should be properly constructed in order to serve different aims.

Following this torrent, we propose an approach for intrusion detection, which is an amalgamation of k-means, neural network and SVM techniques to enhance detection precision.

III. PROPOSED FRAMEWORK FOR IDS

In this section, we elaborate our proposed new approach. First of all, we present the whole framework of the new approach. Then we discuss the four main modules, i.e., k-means clustering module, neuro-fuzzy training module, SVM training vector module, and radial-SVM classification module. The proposed intrusion detection technique initially clusters the given training data set by using k-means clustering technique into k-clusters, where 'k' is the number of desired clusters. In the next step, neuro-fuzzy training is used to train 'k' neural networks, where each of the data in a particular cluster is trained with the respective neural network associated with each of the cluster. Subsequently, vector for SVM classification is generated. This vector consists of attribute values obtained by passing each of the data through all of the trained neuro-fuzzy classifiers, and an additional attribute which has membership value of each of the data. As a last step, classification is performed by using radial SVM to detect intrusion has happened or not. The block diagram of the proposed technique is given in fig.1.

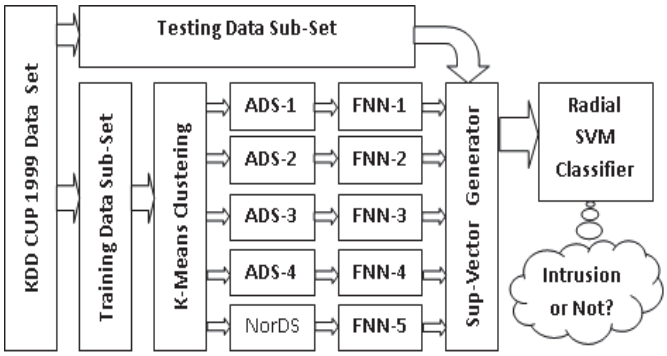

Fig. 1. Block diagram of proposed technique

The dataset given as input for our intrusion detection technique consists of large number of data, where each of the data considered has numerous attributes associated with it. Hence, to perform classification considering all these attributes is a hectic and time consuming task. Processing and executing this lump amount of data results in increasing the error rate and also negatively affects efficacy of the classifier system. In order to overcome this problem, our proposed technique comes up with a solution where the number of attributes defining each of the data is reduced to a small number through a sequence of steps. This process ultimately results in making the intrusion detection more efficient and also yields a less complex system with a better result. Data set

used to evaluate the validity of the proposed technique is prepared from the KDD Cup 1999 data set and the detailed explanation of it is given in section IV.

A. K-means Clustering module

The clustering algorithms are used to group unlabeled data. In our proposed technique, we are intended to group our input data set into different clusters based on types of intrusions. Since our input data set consists of the normal data and four different types of attacks, training data set is grouped into 5 clusters using k- means clustering technique. Examining and learning the behavior and characteristics of a single data point within a cluster can give hints and clue on all other data points in the same cluster. This is because of the fact that all data points inside a cluster differ only by a small amount and usually follow a more or less similar structure. Hence, clustering the data and then classifying is a simpler method and is less time consuming.

K-means is a prototype-based, partitional clustering technique that attempts to find a user-specified number (k) of clusters, which are represented by their centroids. K-means is one of the simplest unsupervised clustering algorithms that solve the well known problems in many fields. The k-means algorithm takes the input parameter 'k' and partitions a set of 'n' data points into k clusters so that the resulting intra-cluster similarity is high but the inter-cluster similarity is low. The aim of k-means cluster module is to partition a given set of data into clusters, where data belonging to different clusters should be as different as possible. The k-means algorithm [8] proceeds as follows.

1. Define number of clusters 'k'.
2. Initialize the k cluster centroids. This can be done by arbitrarily dividing all objects into k clusters, computing their centroids, and verifying that all centroids are different from each other. Alternatively, the centroids can be initialized to k arbitrarily chosen, different objects.
3. Iterate over all data points in the data set and compute the distances to the centroids of all clusters. Assign each data point to the cluster with the nearest centroid.
4. Re calculate 'k' new centroids as par centers of the clusters resulting from the previous step.
5. Repeat step 3 until the centroids do not change any more.

The goal of clustering is typically expressed by an objective function that depends on the proximities of the points to one another or to the cluster centroids; e.g., minimize the squared distance of each point to its closest centriod. Considering data whose proximity measure is Euclidian distance, for our objective function, we use sum of squared error (SEE). Sum squared error (SEE) is defined as given in equation below.

$$J = \sum_{J=1}^{K} \sum_{i=1}^{n} \| x_i^{(j)} - C_j \|^2$$

Where $\| x_i^{(j)} - C_j \|^2$ is a chosen distance measure between a data point and cluster center Cj is an indicator of the distance of the *n* data points from their respective cluster centers. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function.

We have employed k-means clustering as time incurred is less when compared to hierarchical clustering and yields a better result. Through k-means clustering module, the training data set is clustered into 5 subsets wherein 4 subset will be a type of the intrusion called attack data set (ADS) and one with normal data type called normal data set (NorDS). Due to the fact that the size and complexity of every training subset is reduced, the efficiency and effectiveness of subsequent fuzzy neural network (FNN) module can be improved. The detection precision, especially for low-frequent attacks, can also be enhanced.

B. Neuro-fuzzy Training Module

Neural networks are a significant tool for classification. The ability of high tolerance for learning-by-example makes neural networks flexible and powerful in IDS. But it has many disadvantages of having impossible interpretation of the functionality and also faces difficulty in determining the number of layers and the number of neurons [9]. These disadvantages can be overcome by incorporating fuzzy into neural networks and consequences in better results and outcomes. Neuro-fuzzy hybridization is widely termed as FNN or neuro-fuzzy system (NFS) in the literature [10]. Neuro-fuzzy refers to the combination of fuzzy set theory and neural networks with the advantages of both. Neuro-fuzzy incorporates fuzzy sets and a linguistic model consisting of a set of IF-THEN fuzzy rules. The main strength of neuro-fuzzy systems is that they are universal approximators with the ability to solicit interpretable IF-THEN rules. Buckley et al [11] argued that Fuzzy systems and feed forward neural networks are able to a approximate each other to any degree of accuracy, implying the universal approximate property of fuzzy systems. The main advantages of using neuro-fuzzy are that it can handle any kind of information (numeric, linguistic, logical, etc.). It can manage imprecise, partial, vague or imperfect information. It can resolve conflicts by collaboration and aggregation. It has self-learning, self-organizing and self-tuning capabilities. There is no need of prior knowledge of relationships of data mimic human decision making process. It can perform fast computation using fuzzy number operations.

K-means clustering results in the formation of 'k' clusters where each cluster will be a type of intrusion (ADS-1 to ADS-4) or the normal data (NorDS). For every cluster, we have a neuro-fuzzy classifiers (FNN-1 to FNN-5) associated with it i.e., there will be 5 number of neuro-fuzzy classifiers for 5 number of clusters formed. Each neuro-fuzzy classifier is trained with the data in the respective cluster. Neuro-fuzzy makes use of back-propagation learning to find out the input membership function parameters and the least mean square method to find out the consequent parameters.

Fig.2 shows the neuro-fuzzy architecture. The first hidden layer maps the input variable correspondingly to each membership functions. In the second hidden layer, T-norm operator is used to compute the antecedents of the rules. The rules strengths are normalized in the third hidden layer and subsequently in the fourth hidden layer the consequents of the

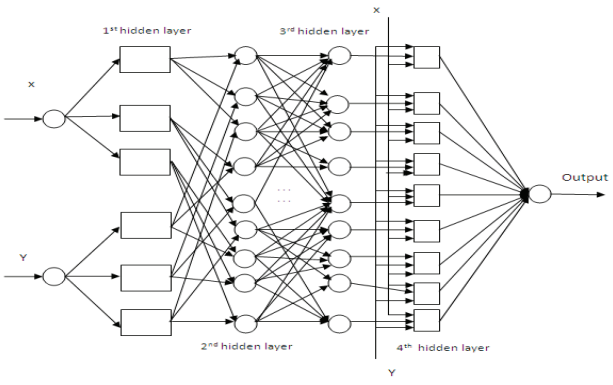

Fig.2. Neuro-fuzzy architecture.

rules are found out. The output layer computes the result or output as the summation of all the signals that reach to this layer. Neuro-fuzzy makes use of back-propagation learning to find out the input membership functions parameters and the least mean square method to find out the consequents parameters. Here in every iteration step, learning algorithm consists of two parts. In the initial part, the input patterns are propagated and the parameters of the consequents are computed making use of the iterative minimum squared method algorithm, while the parameters of the premises are taken to be fixed. In the other part, the input patterns are propagated again and in every iterative step, the learning algorithm back propagation is made use of on order to alter the parameters of the premises, while the consequents stay fixed.

For the every cluster formed in the previous step, we have a FNN associated with it. That is, there will be 5 numbers of FNNs; each FNN is trained with the data in the respective cluster. FNN module aims to learn the pattern of every subset.

C. SVM vector generation Module

Classification of the data point considering all its attributes is a very difficult task and takes much time for the processing, hence decreasing the number of attributes related with each of the data point is of paramount importance. The main purpose of the proposed technique is to decrease the number of attributes associated with each data, so that classification can be made in a simpler and easier way. Neuro-fuzzy classifier is employed to efficiently decrease the number of attributes.

The input data is trained with neuro-fuzzy after the initial clustering as we have discussed earlier, then the vector necessary for the SVM is generated. The vector array $S=\{D_1, D_2,\ldots D_N\}$ where, D_i is the i^{th} data and 'N' is a total number of input data. Here, after training through the neuro-fuzzy the attribute number reduces to 'k' numbers. $D_i = \{a_1, a_2 \ldots a_k\}$, here the D_i is the i^{th} data governed by attribute values a_i, where a_i will have the value after passing through the i^{th} neuro-fuzzy. Total number of neuro-fuzzy classifiers trained will be 'k', corresponding to the 'k' clusters formed after clustering. Initially, we have used the fuzzy c-means clustering which is error prone and does not yield the exact values. Hence, in order to overcome this and to have a better result we include a parameter known as membership value. Inclusion of the membership value into the attribute list results in a better performance of the classifier. Membership value μ_{ij} is defined as given by the equation below.

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\| x_i - c_i \|}{\| x_i - c_k \|} \right)^{\frac{2}{m-1}}}$$

Hence the training vector is modified as $S^*= \{D^*_1, D^*_2,.. D^*_N\}$ where S* is the modified SVM vector which consists of modified data D^*_i, which consists of an extra attribute of membership value μ_{ij}. $D_1^* = \{a_1, a_2,\ldots. a_k, \mu_{ij}\}$, hence the attribute number is reduced to k+1 where 'k' is the number of clusters formed in the start on. This results in easy processing in the final SVM classification. This is due to the fact that input data which had 34 attributes is now constrained to 6 attributes. This also reduces the system complexity and time incurred.

D. Radial *SVM classifier Module*

SVM classifier is used as it produces better results for binary classification when compared to the other classifiers. But use of linear SVM has the disadvantages of getting less accuracy result, over fitting results and robust to noise. These short comings are effectively suppressed by the use of the radial SVM where nonlinear kernel functions are used and the resulting maximum-margin hyper-plane fits in a transformed feature space. Hilbert space of infinite dimensions is formed as the corresponding feature space when the kernel used is a Gaussian radial basis function. In our proposed technique, nonlinear kernel functions are used and the resulting maximum-margin hyper-plane fits in a transformed feature space. When the kernel used is a Gaussian radial basis function, the corresponding feature space is a Hilbert space of infinite dimensions. The Gaussian Radial Basics function is given by the equation below.

$$\phi(x - x_j) = \exp\left(-\frac{1}{2\sigma_j^2} \| x - x_j \|^2 \right) J = 1,2,\ldots N$$

The 'j'th input data point xj defines the center of radial basis function, the vector 'x' is the pattern applied to the input. σ_j is a measure of width of 'j'th Gaussian function with center xj.

The input dataset having large number of attributes is changed into data having k+1 attributes by performing the above steps. The data with constrained number of attributes is given to the radial SVM, which is binary, classified to detect if there is any intrusion or not.

IV. EXPERIMENTAL SETUP AND RESULTS

To evaluate the performance of our proposed approach, a series of experiments on KDD CUP 1999 dataset were conducted. We carried out these experiments by implementing proposed IDS in MATLAB version R2011a on a Windows PC with core to duo 1.83 GHz CPU and 4GB RAM.

A. Data Preparation

We used KDD CUP 1999 dataset for experiments. This dataset is a version of the original 1998 DARPA intrusion detection evaluation program, which is prepared and managed by the MIT Lincoln Laboratory. This dataset is one of the most rational publicly available data set that include actual attacks [12]. For that reason, researchers have been using this dataset to design and evaluate their intrusion detection systems. Furthermore it is a common dataset that allows researchers to compare their experimental results.

KDD CUP 1999 data set was obtained from raw TCP-dump data for a length of nine weeks. It is made up of a large number of network traffic activities that include both normal and malicious connections. KDD Cup 1999 dataset contains about five million connection records as training data and about two million connection records as test data. Each instance in the KDD Cup 1999 datasets contains 41 features that describe a connection and is marked as either normal or an attack, with accurately one particular attack type. Features 1-9 stands for the basic features of a packet, 10-22 for content features, 23-31 for traffic features and 32-41 for host based features. There are 38 different types attack in training and test data together and these types of attack fall into four main categories: PROBE(sometimes called Probing), denial of service(DOS), remote to local(R2L) and user to root(U2R) [13].

DOS and PROBE attacks come with greater frequency and can be easily separate from normal activities. In contrast, U2R and R2L attacks are embedded in the data portions of the packet and hence it is difficult to achieve detection accuracy for these two attacks. A complete listing of the set of features given in KDD Cup 99 dataset defined for the connection records and types of attacks falling into four major categories are given in [14].

Since KDD cup 99 dataset is of huge size, it is very tough and requires very high-end machines to perform experimentation. Because of this, a subset of 10% of KDD Cup 99 dataset is made use for our experimentation. The number of data points taken for training and testing phase is given in table 1. Totally, in training, we considered 26114 data points and in testing we considered 27112 data points.

B. Evaluation measure

The measurements used for evaluation of our proposed techniques are true positive (TP), true negative (TN), false positive (FP) and false negative (FN). True positive indicates the number of attacks records that are correctly classified. A true positive be a sign of properly detecting the occurrence of attack in IDS. True negative indicates the number of valid records that are correctly classified.

A true negative specifies that the IDS have not made a mistake in detecting a normal condition. False positive indicates records that were incorrectly classified as attacks whereas in fact they are valid activities. A false positive specifies the wrong detection of a particular attack by IDS. A false positive is often produced due to loose recognition conditions and it represents the accuracy of the detection system. False negative indicates records that were incorrectly classified as valid activities whereas in fact they are attacks. A false negative stipulate that the IDS is unable to detect the intrusion after a particular attack has occurred.

Due to the fact that the number of instances of U2R and R2L attacks in the training data set and testing data set is every low, these numbers are not adequate as a standard performance measure [15]. It could be biased if we use these numbers as a measure for the performance of the system. Hence, we used evaluation metrics like sensitivity, specificity and accuracy which are independent of the size of the training and the testing samples. They are defined as follows.

$$Sensitivity = TP/(TP + FN)$$

$$Specificity = TN/(TN + FP)$$

$$Accuracy = \frac{(\mathrm{TN} + \mathrm{TP})}{\mathrm{TN} + \mathrm{TP} + \mathrm{FN} + \mathrm{FP}}$$

So as to discover these metrics, we first compute confusion matrix for both training and testing data set and then we apply these values into the equations shown above to find sensitivity, specificity and accuracy. The obtained results for all attacks and normal data are given in table 2.

Table 1. Data Points taken for Training and Testing.

	Normal	DOS	PROBE	R2L	U2R
Training	12500	12500	1054	39	21
Testing	12500	12500	2053	38	21

Table 3. Accuracy comparison with existing methods

DIFFERENT METHODS	PROBE	DOS	U2R	R2L
KDD 99 Winner[17]	83.3	97.1	13.2	8.4
PNrule[18]	73.2	96.9	6.6	10.7
Multi-Class SVM[19]	75	96.8	5.3	4.2
Layered conditional random fields[20]	98.60	97.40	86.30	29.60
Columbia Model[21]	96.7	24.3	81.8	5.9
Decision tree[22]	81.4	60.0	58.8	24.2
BPNN[23]	99.3	98.1	89.7	48.2
Our Technique	**97.31**	**98.80**	**97.52**	**97.51**

The results are evaluated with the evaluation metrics namely, sensitivity, specificity and accuracy [16]. From the table, it is clear that we have achieved about 98.80% accuracy in case of DOS attack and reached heights of 97.31% accuracy in case of PROBE attack. In the case of R2L and U2R attacks it has attained 97.5% accuracy.

Table 2. Experimental results obtained for the training and testing dataset.

METRICS	TYPES OF ATTACKS							
	DOS		PROBE		R2L		U2R	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
True Negative (TN)	12487	12199	12487	12199	12487	12199	12487	12199
False Positive (FP)	13	301	13	301	13	301	13	301
True Positive (TP)	12500	12500	2025	1963	38	25	14	12
False Negative (FN)	0	0	29	90	1	13	7	9
Specificity	99.87	97.59	99.90	97.59	99.91	97.59	99.91	97.59
Sensitivity	1	1	98.59	95.62	97.44	65.79	66.67	52.38
Accuracy	**99.95**	**98.80**	**99.71**	**97.31**	**99.89**	**97.51**	**99.84**	**97.52**

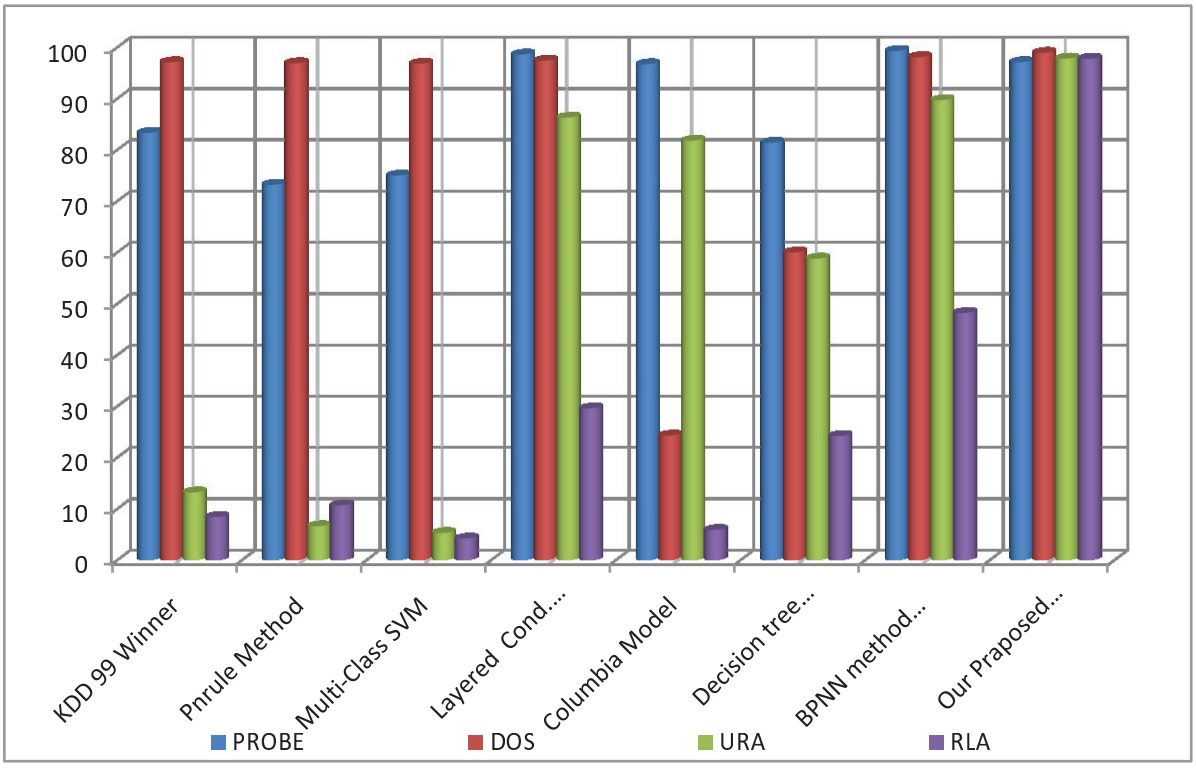

Fig. 2. Graphical representation of comparison of various other techniques with our proposed Technique

Table 3 demonstrates the comparison of our technique with other state of art techniques. From the table 3, it is apparent that the our technique has obtained a reliable peak scores for all types of intrusions. In the case of DOS intrusion, we have attained 98.80%, which is the maximum accuracy value when compared to other methods. In the case of the PROBE intrusion, we have attained a very good result of 97.31% accuracy.

For both U2R and R2L, once again we have reached the maximum value of 97.52% and 97.51% accuracy respectively when compared to others. Our technique which use both the neural network and SVM classifier performs well in all types of intrusions. We have received the best results as we have employed fuzzy-neural networks to diminish the number of attributes of the data and by the utilization of radial SVM in the final classification. Fig.2 shows the comparative results in graphical form.

V. CONCLUSION

It is impossible to prevent security violation completely by using the existing security technologies. Accordingly, intrusion detection is an important component in network security. IDS helps the information security community by increasing detection efficiency, reducing the manpower needed in monitoring and helping to learn new vulnerabilities by providing legal evidence.

In this paper we present an efficient technique for intrusion detection by making use of k-means clustering, fuzzy-neural networks and radial support vector machine. We took the help of k-means clustering technique to make large, heterogeneous training data set in to a number of homogenous subsets. As a result complexity of each subset is reduced and consequently the detection performance is increased. After initial clustering in the proposed technique, training will be given to fuzzy neural network and later SVM vector will be formed. At the end, the radial SVM will be used to perform final classification. The experimental results using the KDD CUP 1999 dataset exhibits the effectiveness of our new approach specially for low-frequent attacks, i.e., R2L and U2R attacks in terms of detection precision. KDD cup 99 dataset is used for experimental verification. Here, we have used confusion matrix for the purpose of evaluation of our proposed technique and the results are evaluated with the evaluation metrics namely, sensitivity, specificity and accuracy. We have received the best results and these are compared with results of other existing methods and from that, it is clear that our proposed technique outperformed all other state of art techniques.

REFERENCES

[1] David Wagner and Paolo Soto, "Mimicry Attacks on Host Based Intrusion Detection Systems" Proceedings of the 9th ACM conference on Computer and communications security, pp. 255 – 264, 2002.

[2] Ghanshyam Prasad Dubey, Neetesh Gupta and Rakesh K Bhujade, "A Novel Approach to Intrusion Detection System using Rough Set Theory and Incremental SVM", International Journal of Soft Computing and Engineering (IJSCE), vol.1, no.1, pp.14-18, 2011.

[3] Hansung Lee, Jiyoung Song, and Daihee Park, "Intrusion Detection System Based on Multi-class SVM", Dept. of computer & Information Science, Korea Univ., Korea, pp. 511–519, 2005.

[4] Jirapummin, C., Wattanapongsakorn, N., & Kanthamanon, P. "Hybrid neural networks for intrusion detection system".Proceedings of ITC–CSCC, pp 928-931, 2002.

[5] Horeis, T, "Intrusion detection with neural network – Combination of self organizing maps and redial basis function networks for human expert integration",a Research report 2003. Available in *http://ieee-cis.org/_files/EAC_Research_2003_Report_Horeis.pdf*

[6] Han, S. J., & Cho, S. B. "Evolutionary neural networks for anomaly detection based on the behavior of a program", IEEE Transactions on Systems, Man and Cybernetics (Part B), 36(3), pp. 559–570, 2005.

[7] Chen, Y. H., Abraham, A., & Yang, B, "Hybrid flexible neural-tree-based intrusion detection systems", International Journal of Intelligent Systems(IJIS), 22(4), pp. 337–352, 2007.

[8] A. M. Chandrashekhar and K. Raguveer. "Performance evaluation of data clustering techniques using KDD cup 99 intrusion data set" International journal of information and network security(IJINS), Vol 1(4), pp. 294-305, 2012.

[9] R. Jang. "Neuro-Fuzzy Modeling: Architectures, Analysis and Applications", Ph D Thesis, University of California, Berkley, 1992.

[10] Jose Vieira, Fernando Morgado Dias and Alexandre Mota. "Neuro-Fuzzy Systems, A Survey". Proceedings International Conference on Neural Networks and Applications, 2004.

[11] J.J. *Buckley*, Y. Hayashi and E. Czogala, "On the equivalence of neural nets and *fuzzy* expert systems, *Fuzzy* Sets & Systems", A Research Report , pp.129-134. *1993.*

[12] Aickelin, U., Twycross, J., Hesketh-Roberts, T "Rule generalization in intrusion detection systems using SNORT", International Journal of Electronic Security and Digital Forensics, 1 (1), pp. 101–116, 2007.

[13] T. G. Dietterich and G. Bakiri. "Solving multiclass learning problems via error-correcting output codes", Journal of Artificial Intelligence Research (JAIR) vol 2, pp. 263-286, 1995.

[14] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani. "A detailed analysis of the KDD CUP 99 data set", Proceedings IEEE international conference on Computational intelligence for security and defense applications, pp. 53-58, Ottawa, Ontario, Canada, 2009.

[15] Dokas, P., Ertoz, L., Lazarevic, A., Srivastava, J., & Tan, P. N. "Data mining for network intrusion detection", Proceeding of NGDM., pp.21–30, 2002.

[16] Wen Zhu, Nancy Zeng, Ning Wang. "Sensitivity, Specificity, Accuracy, Associated Confidence Interval and ROC Analysis with Practical SAS Implementations", Proceedingsof NESUG Health Care and Life Sciences, Baltimore, Maryland, 2010.

[17] B. Pfahringer, "Winning the KDD99 Classification Cup: Bagged Boosting," SIGKDD Explorations, vol. 1, pp. 65–66, 2000.

[18] R. Agarwal and M. V. Joshi, "PNrule: A New Framework for Learning Classifier Models in Data Mining," in A Case-Study in Network Intrusion Detection, 2000.

[19] T. Ambwani, "Multi class support vector machine implementation to intrusion detection," Procedings of IJCNN, pp. 2300-2305, 2003.

[20] K. K. Gupta, B. Nath, and R. Kotagiri, "Layered Approach using Conditional Random Fields for Intrusion Detection," IEEE Transactions on Dependable and Secure Computing, vol. 5, 2008.

[21] W. Lee and S. Stolfo, "A Framework for Constructing Features and Models for Intrusion Detection Systems," Information and System Security, vol. 4, pp. 227-261, 2000.

[22] J.-H. Lee, J.-H. Lee, S.-G. Sohn, J.-H. Ryu, and T.-M. Chung, "Effective Value of Decision Tree with KDD 99 Intrusion Detection Datasets for Intrusion Detection System," Procedings of 10th International Conference on Advanced Communication Technology. vol. 2, pp. 1170-1175, 2008.

[23] Tich Phuoc Tran, Longbing Cao , Dat Tran and Cuong Duc Nguyen , "Novel Intrusion Detection using Probabilistic Neural Network and Adaptive Boosting", International Journal of Computer Science and Information Security (IJCSI),Vol. 6, No. 1, pp.83-91, 2009.