International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.1, January 2013

FORTIFICATION OF HYBRID INTRUSION DETECTION SYSTEM USING VARIANTS OF NEURAL NETWORKS AND SUPPORT VECTOR MACHINES

A. M. Chandrashekhar[1] and K. Raghuveer[2]

[1]Department of Computer **Science**, Sri Jayachamarajendra College of Engineering (SJCE), Mysore, Karnataka, 570006, India,
amblechandru@gmail.com
[2]Department of Information Science, National Institute of Engineering (NIE), Mysore, Karnataka 57008, India**.**
raghunie@yahoo.com

ABSTRACT

Intrusion Detection Systems (IDS) form a key part of system defence, where it identifies abnormal activities happening in a computer system. In recent years different soft computing based techniques have been proposed for the development of IDS. On the other hand, intrusion detection is not yet a perfect technology. This has provided an opportunity for data mining to make quite a lot of important contributions in the field of intrusion detection. In this paper we have proposed a new hybrid technique by utilizing data mining techniques such as fuzzy C means clustering, Fuzzy neural network / Neuro-fuzzy and radial basis function(RBF) SVM for fortification of the intrusion detection system. The proposed technique has five major steps in which, first step is to perform the relevance analysis, and then input data is clustered using Fuzzy C-means clustering. After that, neuro-fuzzy is trained, such that each of the data point is trained with the corresponding neuro-fuzzy classifier associated with the cluster. Subsequently, a vector for SVM classification is formed and in the last step, classification using RBF-SVM is performed to detect intrusion has happened or not. Data set used is the KDD cup 1999 dataset and we have used precision, recall, F-measure and accuracy as the evaluation metrics parameters. Our technique could achieve better accuracy for all types of intrusions. The results of proposed technique are compared with the other existing techniques. These comparisons proved the effectiveness of our technique.

KEYWORDS

Intrusion Detection System, Fuzzy C-Means Clustering, fuzzy neural network, Support Vector Machine

1. INTRODUCTION

Due to the sudden growth and extension of World Wide Web and network systems the computing world is witnessing enormous changes and challenges. Intrusion detection system has been an active area of research and development for the past few decades. This is primarily because of the mounting of attacks on computers and on networks in recent years and computerized scrutiny has become a compulsory addition to IT security [1]. Malicious ways like in the form of viruses, self-propagating worms, and denial of service attacks is a brutal threat to the internet and to the infrastructures using it for communication. The catastrophe of analysing illegitimate access of computer systems on the network which is to make out individuals who are using a computer system without permission (crackers) and those who have legal access to the system but are prevail over their privileges (insider threat) is known as intrusion detection. The system that detects and logs illegal access is called as intrusion detection system [2]. An intrusion detection system (IDS) typically consists of three major functional elements: an information source that provides a stream of event records, an analysis

DOI : 10.5121/ijnsa.2013.5106

Engine that determines signs of intrusions and a decision maker that applies a number of rules on the results of the analysis engine, and resolves what reactions should be done based on the outcomes of the analysis engine [3]. Based on a study on latest research literatures, there are quite a lot of result 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 [4].

There are three categories of intrusion detection systems which are host-based where information is found on a single or multiple host systems, network- based that examines the information captured from network communications and vulnerability assessment-based that identifies vulnerabilities in internal networks and firewall whereas based on the functionality intrusion detection can be classified into two as anomaly detection and misuse detection [5]. Intrusions are diagnosed by searching for actions that correlate to signatures of intrusions or vulnerabilities previously known in misuse detection based IDS. All signatures of possible attacks are incorporated in a pattern database that comprises the IDS. When the system matches the data with the attack pattern, the IDS look upon it as an attack [6]. It can be used very swiftly. The IDS don't have any prerequisite to "learn" the network activities before it is to be used. The overhead, which can turn out to be unacceptably high is one of the principal concerns of IDS. The operating system should keep track of all information concerning the actions performed for analysing system logs, which continuously results in huge amounts of data, requiring disk space and CPU resource [7]. On the other hand, anomaly detection based-IDS detect intrusions by searching for unusual network traffic [8]. Conventional techniques of network intrusion detection rely on the saved patterns of known attacks. Human experts present the attack patterns which are used to evaluate the network connection features for detecting intrusions [9]. Network administrators could take preventive measures by detecting planned and coordinated attacks through distributed monitoring [10].

In intrusion detection systems that relies on expert system, the rule-based detection using Denning's [11] profile model can be employed. A set of predefined rules that are supplied by an administrator or created by the system is employed for Rule-based examination. Kumar *et al.* [12] put forward a misuse detection system that uses pattern matching. The well-known intrusion signatures are determined as patterns in their system, which are then evaluated with the audit data commenced by the analysis component. An anomaly detection algorithm based on robust support vector machines (SVM) has been put forward by Hu *et al.* [13] in which, they could efficiently identified the intrusions although noise is there. The conventional SVM has been extended to Robust SVM and one-class SVM to be of online forms to get better training efficiency. A hierarchical hybrid intelligent system named DT-SVM that depends on decision trees (DT) and support vector machines to detect intrusions have been proposed by Peddabachigari *et al.* [14].

Chengjie GU *et al.* [15] proposed proximal support vector machine (PSVM) rather than SVM, which steer to an enormously fast and simple algorithm for generating a system of linear equations. The major drawback of traditional methods is that they cannot distinguish unknown intrusion and are incapable of adapting to new networks. A number of short-comings of the Anomaly-based Intrusion Detection are an elevated false positive rate and the capability to be fooled by a correctly delivered attack [16]. Intrusion detection systems that follow the misuse detection model need to be continuously updated to stay a step ahead of the hackers.

The motivation for our proposed intrusion detection technique is as follows: The dataset given as input for 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 also results in increasing the error rate and also negatively affects efficacy of the classifier system. In order to overcome this problem, our proposed technique we come 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. As typical machine learning framework consisting of Fuzzy C means and neuro-fuzzy (FC-NF) incorporates both the training phase and testing phase. Data set used to evaluate the validity of the proposed technique is prepared from the KDD CUP 1999 data set and explained in detail in section 3.

In the first step, Fuzzy C-means clustering is performed to separate the input data into various types of attacks and the normal data. Subsequently, neural network is trained, such that each of the data point is trained with the corresponding neural network associated with the cluster. As a result, 'K' neural networks are trained with the respective data points. Then, a vector for SVM classification is generated. For the vector formation, every data is passed through all of the 'K' trained neural networks. That is each data points receive 'K' attribute values after passing through 'K' neural networks. The membership function value of the data is found out and added to the attribute list in order to decrease the error probability and improve results. And in the final step, classification using RBF-SVM was performed to detect intrusion has happened or not. In order to improve the accuracy of our previous intrusion detection technique [43], we make use of the neuro-fuzzy rather than the neural network and the linear-SVM is replaced with the radial basis-SVM.

The rest of this manuscript is prearranged as follows: brief review of recent researches related to our proposed technique is presented in the section 2. The section 3 depicts the dataset taken for experimentation. The proposed technique for intrusion detection using neuro-fuzzy and radial SVM classifier is presented in section 4. The detailed experimental results and discussions are given in section 5. The conclusions are summed up in section 6.

2. SURVEY OF RELATED RECENT WORKS

As per the literature, lot of research work have been carried out in intrusion detection using various techniques. And some of them have inspired us to take up this research. Brief descriptions of some of those recent substantial researches in the area are presented below:

Mansour Sheikhan and Amir Khalili [17] proposed an algorithm which was utilized to develop IDS and classify the patterns of intrusion. To evaluate the performance of their system with other machine learning algorithms, multi-layer perception (MLP) with output weight optimization-hidden weight optimization (OWO-HWO) training algorithm was used with selected inputs based on the results of a feature significance analysis. Empirical results depicted the better performance of the IDS based on rule extraction from DCS, in recognizing hard-detectable attack categories, e.g. user to-root (U2R) and also permitting competitive false alarm rate (FAR). Although, MLP with 25 selected input features, instead of 41 standard features established by knowledge discovery and data mining group (KDD), performed better in terms of detection rate (DR) and cost per example (CPE) when compared with some other machine learning methods.

Kyaw Thet Khaing [18] proposed an improved SVM Model consisting of Recursive Feature Elimination (RFE) and a k-Nearest Neighbour (KNN) technique to carry out a feature ranking and selection job of the model. RFE could shrink redundant & recursive features and KNN could pick more accurately than conventional SVM. Experimentations and comparisons were carried out through intrusion dataset: the KDD Cup 1999 dataset. To grade the features of intrusion detection data, the model adopted Recursive Feature Elimination (RFE) and to improve extra accuracy in classification it takes on k-Nearest Neighbour (KNN). Only the important features would be counted when training an SVM. It was suggested that model was efficient for the KDD cup dataset. Even though the precision levels of conventional SVM and the model were not mostly different, the model's false negative rates were lower than the traditional SVM model. Additionally, the time taken to detect an intrusion in the model was

much less than the conventional SVM. It also had an advantage, i.e., the running time was much less as lesser numbers of features were used for classification.

Arvind Mewada *et al.* [19] explored the performance of MSVM for different categories of attacks. Statistical IDS triumph over many drawbacks present in signature based IDS. Statistical IDS uses models such as C4.5, NB etc for classification to discover intrusions. Multiclass Support Vector Machine was able to carry out multiclass classification. This paper uncovered the performance of MSVM (1-versus many, 1-versus-1 and Error Correcting Output Coding (ECOC)) and it's deviation for statistical NBIDS.

Snehal A *et al.* [20] proposed the decision tree based algorithm to build multiclass intrusion detection system. Support Vector Machines was the classifiers which were at first designed for binary classification. The classification applications could solve multi-class problems. Decision-tree-based support vector machine which combined support vector machines and decision tree could be an efficient way for solving multi-class problems. This method could diminish the training and testing time, increasing the efficiency of the system. The diverse ways to build the binary trees divides the data set into two compartments from root to the leaf in anticipation of every subset included of only one class. The construction order of binary tree had great influence on the classification performance.

Gang Wang *et al.* [21] proposed an approach, called FC-ANN, based on ANN and fuzzy clustering, to unravel the problem and help IDS achieve higher detection rate, less false positive rate and stronger stability. The general procedure of FC-ANN was as follows: first of all, fuzzy clustering technique was used to generate diverse training subsets. Consequently, based on different training subsets, dissimilar ANN models were trained to formulate diverse base models. Finally, a meta-learner, fuzzy aggregation module, was employed to aggregate these results. Experimental results on the KDD CUP 1999 dataset showed that their approach, FC-ANN, outperformed BPNN and other well-known methods such as decision tree, the naïve Bayes in terms of detection precision and detection stability.

Muna Mhammad T. Jawhar and Monica Mehrotra [22] presented an intrusion detection model derived from neural network and hybrid fuzzy logic. The key thought was to take benefit of different classification capacity of neural network and fuzzy logic for IDS. The model had capability to identify an attack, to distinguish one attack from another that is classifying attack and the essentially vital, to discover new attacks with the rate of high detection rate and low false negative. Required Training and testing data were acquired from the Defence Advanced Research Projects Agency (DARPA) intrusion detection assessment data set. Ghadiri A and Ghadiri N [23] permitted the design parameters to be determined dynamically by accepting layered hybrid architecture, hence resolving the abovementioned inadequacies. The first layer used FCM and GK fuzzy clustering to extract the features and the second layer used a set of RBF neural networks to perform the classification. The initial number of clusters, number of RBF networks and number of neurons inside each network which were determined with minimal input from the user were the flexible design parameters. The simulation result illustrate high detection rates as well as less false positives compared to previous methods.

Pohsiang Tsai *et al.* [24] suggested a Machine Learning (ML) framework in which various types of intrusions would be detected with different classifiers, containing different attribute selections and learning algorithms. Appropriate voting procedures were used to join the outputs of these classifiers. Experiments on the KDD-99 dataset designated that their approach obtained higher performance in comparison with other state-of-the-art detection methods, accomplishing low learning bias and enhanced generalization at a reasonable computational cost.

3. DESCRIPTION OF DATASET TAKEN FOR EXPERIMENTATION

For the testing and evaluation of proposed method and rating its performance, we have taken KDD CUP 99 dataset. DARPA in agreement with Lincoln Laboratory at MIT started the

DARPA 1998 dataset for estimating IDS [25]. The complicated version of DARPA dataset which encircle only network data (TCP-dump data) is named as KDD dataset [26]. KDD training dataset consists of moderately around 5 million vectors single correlation vectors and testing data set contains around 2 million vectors, where each single connection vector consisting of 41 features and is marked as a normal or an attack, through accurately one particular attack type [27]. These characteristics had all types of constant and symbolic with extensively altering ranges falling in to one of the four categories:

- First category in a connection consists of the inherent features which include the primary features of every individual TCP connections. Like for example, some of the features for each individual TCP connections are type of the protocol (TCP, UDP, etc.), duration of the connection and network service (http, telnet, etc.).
- The content features recommended by domain knowledge are employed to compute the payload of the original TCP packets, say for example the count of unsuccessful login attempts.
- The similar host features monitor the familiar connections having the identical target /destination host as present in past two seconds inside a connection and the statistics related to the protocol behaviour, service, etc are estimated.
- The related identical service features investigate the connections having the similar service as the existing connection in past two seconds.

The sample portion of network connection records are shown below in the table-1.

Table 1. Network connection records

timestamp	duration	service	src_host	dst_host	src_bytes	dst_bytes	flag	...
1.1	0	http	spoofed_1	victim	0	0	S0	...
1.1	0	http	spoofed_2	victim	0	0	S0	...
1.1	0	http	spoofed_3	victim	0	0	S0	...
1.1	0	http	spoofed_4	victim	0	0	S0	...
1.1	0	http	spoofed_5	victim	0	0	S0	...
1.1	0	http	spoofed_6	victim	0	0	S0	...
1.1	0	http	spoofed_7	victim	0	0	S0	...
...	...	...	...	...	...	...	...	...
10.1	2	ftp	A	B	200	300	SF	...
12.3	1	smtp	B	D	250	300	SF	...
13.4	60	telnet	A	D	200	12100	SF	...
13.7	1	smtp	B	C	200	300	SF	...
15.2	1	http	D	A	200	0	REJ	...
...	...	...	...	...	...	...	...	...

A wide range of attacks integrated in the dataset come beneath the following four main categories:

- Denial of Service (DOS) Attacks: DOS attack is an attack where as the attacker creates a few calculations or memory resource completely engaged or out of stock to handle authentic requirements, or reject justifiable users the right to utilize a machine.
- User to Root (U2R) Attacks: These are a category of attack where an attacker begins by accessing normal user account in the system (maybe attained by hunting the passwords, by social engineering or by attacking dictionary) and get advantage of several vulnerability to accomplish root entrée to the system.
- Remote to local (R2L) Attacks: R2L attack occurs when an intruder who has the potential to send packets to a system/machine over a network without having an account in that system/machine, makes use of a few vulnerability to accomplish local access as a client of that system/machine.
- Probes (PROBE) Attack: Probing is a collection of attacks where an attacker scrutinizes a network to gather information or to conclude prominent vulnerabilities.

Table 2. List of continuous features in the connection vector of KDD cup 99 dataset

FI	Feature Name	Description
1	Duration	Length (no. of seconds) of the connection.
5	src_bytes	Number of data bytes from source to destination.
6	dst_bytes	Number of data bytes from destination to source.
8	wrong_fragment	Number of ``wrong'' fragments.
9	Urgent	Number of urgent packets.
10	Hot	Number of ``hot'' indicators.
11	num_failed_logins	Number of failed login attempts.
13	num_compromised	Number of ``compromised'' conditions.
14	root_shell	1 if root shell is obtained; otherwise 0.
15	su_attempted	1 if ``su root'' command attempted; otherwise 0.
16	num_root	Number of ``root'' accesses.
17	num_file_creations	Number of file creation operations.
18	num_shells	Number of shell prompts.
19	num_access_files	Number of operations on access control files.
20	num_outbound_cmds	Number of outbound commands in an ftp session.
23	Count	Number of connections to the same host as the current connection in the earlier period of two seconds.
24	srv_count	Number of connections to the same service as the current connection in the earlier period of two seconds
25	serror_rate	% of connections having ``SYN'' errors
26	srv_serror_rate	% of connections having ``SYN'' errors
27	rerror_rate	% of connections having ``REJ'' errors
28	srv_rerror_rate	% of connections having ``REJ'' errors
29	same_srv_rate	% of connections to the similar service
30	diff_srv_rate	% of connections to dissimilar services
31	srv_diff_host_rate	% of connections to dissimilar hosts
32	dst_host_count	count for target /destination host
33	dst_host_srv_count	srv_count for target /destination host
34	dst_host_same_srv_rate	same_srv_rate for target /destination host
35	dst_host_diff_srv_rate	diff_srv_rate for target /destination host
36	dst_host_same_src_port_rate	same_src_port_rate for target /destination host
37	dst_host_srv_diff_host_rate	diff_host_rate for target /destination host
38	dst_host_serror_rate	serror_rate for target /destination host
39	dst_host_srv_serror_rate	srv_serror_rate for target /destination host
40	dst_host_rerror_rate	rerror_rate for target /destination host
41	dst_host_srv_rerror_rate	srv_serror_rate for target /destination host

Table 3. List of symbolic features in the connection vector of KDD cup 99 dataset

F-Index	Feature Name	Description
2	protocol_type	type of the protocol, say TCP, UDP, etc.
3	Service	network service on the destination, say http, telnet, etc.
4	Flag	normal status or error status of connection
7	Land	1 if connection is to/ from the same port / host; otherwise 0.
12	logged_in	1 if successfully logged in; otherwise 0
21	is_hot_login	1 if the login is fit in to the ``hot'' list; otherwise 0.
22	is_guest_login	1 if the login is a ‘guest’; otherwise 0

These network examinations are reasonably significant for an attacker who is aiming an attack in future. An attacker having a proof, of which system/machine and facilities are accessible on a

given network, can exploit this information to search for subtle points. Here, table-2 presents list of continuous features and table-3 lists the symbolic features for the correlation records. Different attacks that come under four major categories are listed in table 4.

Table 4. Categories of attacks and its types

No.	Category of Attack	Types of Attacks
1	Denial of Service Attacks	neptune, back, land, pod, teardrop, smurf.
2	User to Root Attacks	Buffer_overflow, loadmodule, perl, rootkit.
3	Remote to Local Attacks	guess_passwd, ftp_write, imap, multihop, warezclient, phf, spy, warezmaster.
4	Probes	Portsweep, satan, ipsweep, nmap.

4. PROPOSED TECHNIQUES FOR INTRUSION DETECTION SYSTEM

In this section, we present the architecture of proposed system and techniques we used to implement the proposed system like the fuzzy C-means clustering (FCC), Neuro-fuzzy classifier (NF) and the RBF-SVM are explained. The detailed description of each step used in the proposed technique is also given in this section.

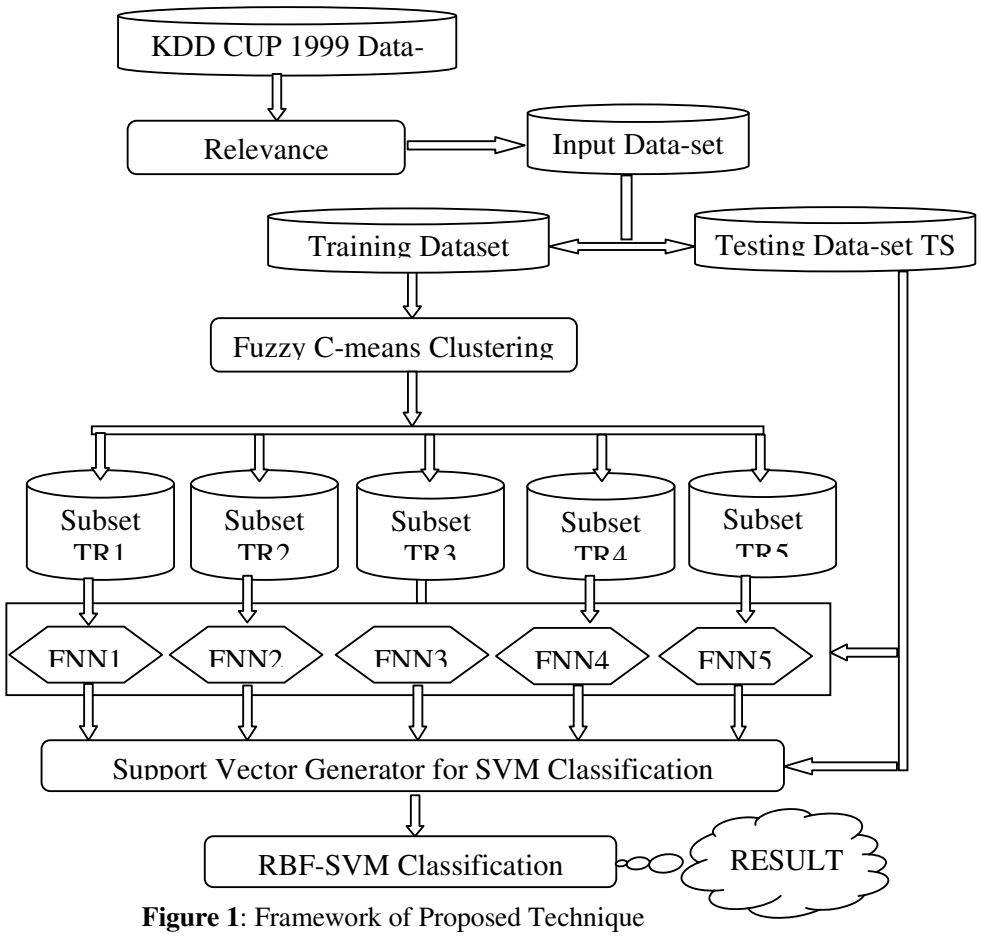

Figure 1: Framework of Proposed Technique

4.1 System Architecture of Proposed IDS

The proposed hybrid intrusion detection technique is a five step methodology. The block diagram of the proposed technique is given below in the figure-1.

- The input data set DS needed for experimentation is prepared by conducting relevance analysis on KDD Cup 1999 data set in order to reduce the irrelevant attributes / features which will not contribute for intrusion detection.
- The input dataset DS is divided into Training Data set T_R and Testing Data set T_S. The Training data is clustered using Fuzzy C-Means Clustering into subsets say $\{T_{R1}, T_{R2}, T_{R3}, ..., T_{Rk}\}$, where, k is the number of clusters desired.
- Neuro-fuzzy (FNN) training NF_i is given to each of the T_{Ri}, where each of the data in a particular cluster is trained with the respective neural network associated with each of the cluster.
- Generation of vector for SVM classification, $S = \{D_1, D_2, ..., D_N\}$ which consists of attribute values obtained by passing each of the data through all of the trained Neuro-fuzzy classifiers, and an additional attribute μ_{ij} which has membership value of each of the data.
- Classification using RBF-SVM to detect intrusion has happened or not.

The detailed description of each of the steps is elaborated in the following sub-sections.

4.2 Input dataset preparation

Attributes in the KDD datasets had all forms of data like symbolic, continuous and discrete with considerably diverging resolution and ranges. Most pattern classification process is not able to route data in such a format. For this reason, pre-processing before building classification models the required. Pre-processing is having two steps: first step accomplishes the mapping of symbolic-valued attributes to numeric-valued attributes and scaling is implemented in the second step.

Attack names (like guess-passwd, buffer-overflow, etc.) were initially mapped in to one of the five classes (say 0 for Normal, 1 for Probe, 2 for DOS, 3 for U2R, and 4 for R2L) as explained by C. Elkan in [28]. The symbolic features like protocol-type, service, and flag were mapped to integer values ranging from 0 to N-1 where N is the number of symbols. Afterwards, each of these features were linearly scaled to the range {0.0, 1.0}. Features containing smaller integer value ranges like wrong-fragment {0, 3}, duration {0,58329},urgent {0, 14}, num-failed-logins {0, 5}, hot {0, 101},num-compromised {0, 9}, su-attempted {0, 2}, num-file-creations {0, 100}, num-shells {0, 5}, num-root {0, 7468}, num-access-files {0, 9}, count {0, 511}, dst-host-count {0, 255}, srv-count {0,511}and dst-host-srv-count {0,255} were also scaled linearly to the range {0.0, 1.0}. Two features spanning over a very large integer range, say dst-bytes {0, 1.3 billion} and src-bytes {0, 1.3 billion}. Logarithmic scaling with base 10 was applied to these features to condense the range to {0.0, 9.14}. All other features were either continuous, like diff-srv-rate, in the range {0.0, 1.0} or Boolean, like logged-in, having values (0 or 1). Therefore, scaling for these attributes are not necessary for these attributes as per the literature of Mahesh kumar sabhanani and gursel Serpen in [29].

4.3 Clustering using fuzzy C-means clustering algorithm

Examining and learning the behaviour 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 the 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.

Our input data set consists of the normal data and different types of intrusions like DOS, PROBE, R2L and U2R. Therefore, clustering results in grouping input data into clusters based on the type of intrusions. In our technique, initially the input data is grouped into clusters by use

of Fuzzy C-Means algorithm (FCM). We have employed Fuzzy C-Means clustering as time incurred is less when compared to hierarchical clustering and yields a better result when compared to K-Means clustering. Here, the input data set is clustered into K clusters, where K is the number of clusters desired which depends on the number of types of intrusion in the input data. Each cluster contains data points which are identical and is useful in identifying the intrusion detection pattern.

Fuzzy c-means [30] is a clustering technique which allows a piece of data to two or more clusters. It depends on minimization of the objective function given below:

$$J_m = \sum_{i=1}^{N}\sum_{j-1}^{C} \mu_{ij}^{m} \| x_j - c_j \|^2 \qquad 1 \le m < \infty$$

where m is any real number greater than 1, μ_{ij} is the degree of membership of x_i in the cluster j, x_i is the ith of d-dimensional measured data, c_j is the d-dimension center of the cluster and $\| * \|$ is any norm stating the similarity among any measured data and the center.

Fuzzy partitioning is achieved through an iterative optimization of objective function given above, with the update of membership μ_{ij} and the cluster centers c_j are given by the equation shown below:

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

This iteration will terminates when his iteration will stop when $\max_{ij}\{| \mu_{ij}^{k=1} - \mu_{ij}^{k} | < \xi\}$, where ξ is a termination criterion between 0 and 1, whereas k is the iteration step. This procedure congregates to a local minimum of J_m. So, the clustering results in forming K number of clusters where each cluster will have data which are alike in a way.

In our case, the data set is given as an input to the clustering process. Clustering proceeds as in the process described above. After FCM clustering, it capitulate K clusters wherein each cluster will be a type of the intrusion except for the one with normal data.

4.4 Training the Fuzzy neural networks

In the proposed technique, we use neuro-fuzzy system and radial SVM in contrast to the neural network and linear SVM which were used in the previous work [43]. Neural networks are a significant tool for classification. But it has many disadvantages of having impossible interpretation of the functionality and also faces difficulty in deciding the number neurons and the number of layers [31]. These disadvantages can be overcome by incorporating Fuzzy into neural networks and results in better results and outcomes. Neuro-fuzzy hybridization is generally termed as Neuro-Fuzzy System (NFS) or Fuzzy Neural Network (FNN) in the literature [32]. Neuro-fuzzy incorporates fuzzy sets and a linguistic model which contains of a set of IF-THEN fuzzy rules. The key strength of neuro-fuzzy systems is that they are universal / general approximators with the capability to solicit understandable IF-THEN rules. Neuro-fuzzy refers to the amalgamation of fuzzy set theory and neural networks having the advantages of both. 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 previous knowledge of relationships of data imitate human decision making process. It can perform fast computation using fuzzy number operations.

Fuzzy C-Means clustering results in the formation of K-clusters where each cluster will be a type of cluster or the normal data. For every cluster, we have a neuro-fuzzy associated with it. That is, there will be K number of neuro-fuzzy classifiers for K number of clusters formed. Each neuro-fuzzy classifier is trained with the data in the respective cluster.

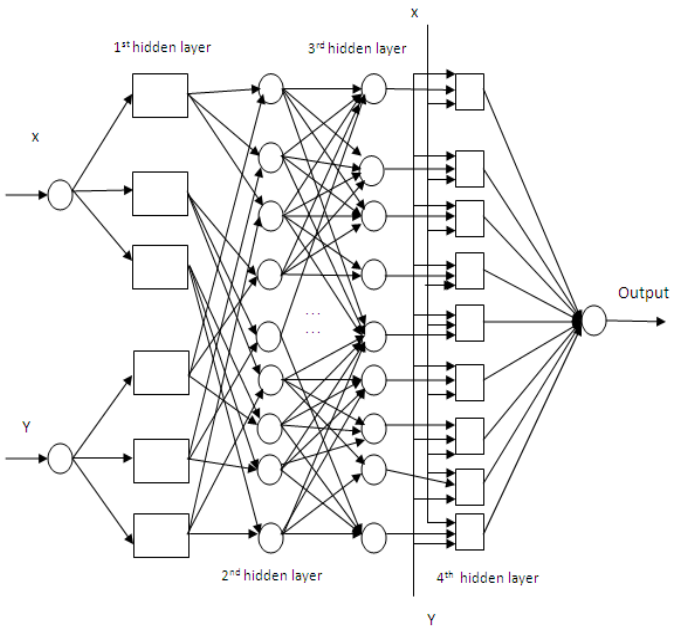

Figure 2: Neuro-fuzzy architecture

Figure-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 rules are found out. The output layer computes the output or result as the summing up all the signals that reach to this layer. Neuro-fuzzy makes use of back-propagation learning to find out the input membership function parameters and 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 disseminated and the parameters of the consequents are computed making use of the iterative minimum squared technique algorithm, whereas the parameters of the premises are taken to be fixed. In the other part, input patterns are propagated all over again and in every iterative step, the learning algorithm back propagation is made use of on order to alter the parameters of the premises, whereas the consequents stay fixed.

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. 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. Executing the reduced amount of data also results in decrease of error rate and the improved performance of the classifier system.

4.5 Generation of SVM training vector

In our system, we are employing radial SVM for the final classification for the intrusion detection. SVM is used as it achieves enhanced results when contrasted to other classification techniques especially when it comes to binary classification. In the final classification, the data is binary classified to detect intrusion or not.

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. Here in the process, each of the data is fed into each of the neural classifier to get the output value. That is each of the data is fed into K number of neuro-fuzzy classifiers to yield K output values. So the data values gets distorted and after passing through the K neuro-fuzzy classifiers, attribute number of the data in consideration changes and diminishes to K numbers where each value will be the output of the data passing through the respective neuro-fuzzy.

$$S = \{D_1, D_2, ..., D_N\}$$

Where, S is the SVM vector array. D_i is the i^{th} data and N is the total number of input data. Here after training through the neuro-fuzzy the attribute number reduces to K numbers.

$$D_i = \{a_1, a_2, ..., a_K\}$$

Here the D_i data is governed by the 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 precise values. Hence so as to overcome this and have improved result we comprise 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 by the equation:

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

Hence, the SVM vector is modified as $S^* = \{D_1^*, D_2^*, ..., D_3^*\}$ 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}. $D_1^* = \{a_1, a_2, ..., a_K, \mu_{ij}\}$ Hence the attribute number is reduced to K+1 where K is the number of clusters. This results in simple processing in the final SVM classification. This is owing to the fact that input data which had 34 attributes is now constrained to K+1 attributes. This also diminishes time incurred.

4.6 Classification using radial basis function support vector machine (RBF-SVM)

Use of radial SVM results in obtaining better results from the classification process when compared to normal linear SVM. In linear SVM, the classification is made by use of linear hyper-planes where as in radial SVM, nonlinear kernel functions are used and the resulting maximum-margin hyper-plane fits in a transformed feature space. The corresponding feature space is a Hilbert space of infinite dimensions, when the kernel used is a Gaussian radial basis function. The Gaussian Radial Basics function is given by the equation:

$$k(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2, for\ \gamma > 0, \gamma = 1/2\sigma^2$$

In the two-class scenario, a support vector classifier [33,34] constructs a try to attain a hyper-plane that reduces the distance from the members of each class to the voluntary hyper-plane. A two-class classification problem can be defined in the following way: Suppose there are M

training samples that can be given by the set pairs $\{(x_i, y_i), i = 1,2,3.....,M)\}$ with x_i being the class label of value ± 1 and $y_i \in {}^n$ where feature vector with n components. The classifier is given by the function $f(y;\alpha) \rightarrow x$ with α, the parameter factors of the classifier. The figure-3 gives us an idea about the margins for an SVM trained with samples from two classes and the Maximum-margin hyper-plane. Samples on the margin are called as the support vectors.

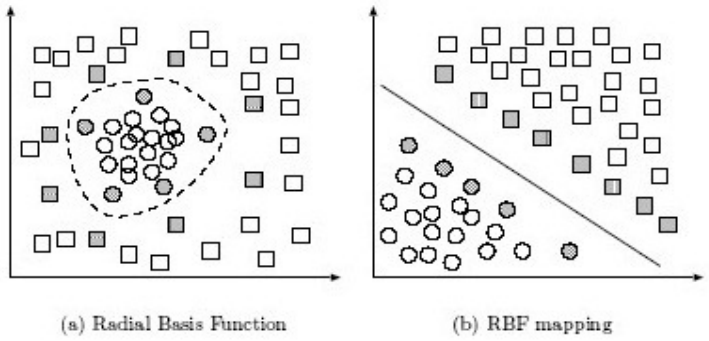

Figure 3: Showing the Radial Basis Fuction and Maximum-margin using RBF mapping.

An optimum separating hyper-plane is established out by the SVM algorithm such that:

1) Samples with labels ±1 are positioned on each side of the hyper-plane;
2) The distance of the nearest vectors to the hyper-plane in each side of maximum are called support vectors and the distance is the optimal margin.

The hyper-plane is given by the equation by $w.y + b = 0$ where (w,b) are the parameter factors of the hyper-plane. The vectors that are not on this hyper-plane guide to $w.y + b > 0$ and let the classifier to be given as $f(y;\alpha) = sgm(w.y + b)$. The support vectors lie on two hyper-planes, which are parallel to the optimal hyper-plane, of equation $w.y + b = \pm 1$.The maximization of the margin with the equations of the two support vector hyper-planes contributes to the following constrained optimization problem $\min \frac{1}{2} \| w \|^2$ with $x_i(w.y + b) \geq 1, i = 1,2,.....,M$ .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, getting 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. A Hilbert space of infinite dimensions is formed as the corresponding feature space when Gaussian radial basis function is the kernel used.

The input dataset having large number of attributes is changed into data having $K + 1$ attributes by performing the above steps. The resultant data having the constrained number of attributes associated with the data is given to the radial SVM classifier for the purpose of intrusion detection. The data with constrained number of attributes is given to the radial SVM, which is binary, classified to detect if there is intrusion or not.

5. RESULTS AND DISCUSSION

In this section, results obtained from the proposed technique are discussed and evaluated. Section 5.1 gives a brief description of the experimental setup, section 5.2 gives the outline of

the evaluation metrics used for the validity of the proposed technique, section 5.3 provides experimentation results and comparative analysis is explained in section 5.4.

5.1 Experimental set up

The proposed technique is implemented using MATLAB. It is very tough to perform the proposed technique on the KDD cup 99 dataset for estimating the performance, because it is of large scale. The subset of 10% of KDD Cup 99 dataset is made use for training and testing. The performance of the proposed technique is estimated by means of Precision, Recall, F-measure and Accuracy. The number of data points taken training for and testing phase is given in table 5.

Table 5. Training and Testing dataset taken for evaluation

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

5.2 Evaluation metrics

During testing phase, testing dataset is given as an input to the proposed technique and the obtained result is estimated with the evaluation metrics namely, precision, recall, F-measure and Accuracy [35]. In order to discover these metrics, first, we calculate confusion matrices like True positive (TP), False negative (FN), True negative (TN), and False positive (FP). The table 6 given below explains the confusion matrix and its definitions.

Table 6: Confusion matrix and its definitions

CONFUSION MATRIX				**DEFINATIONS**
		Predicted Class		**TP and TN**:-True positive and True negative are correct classifications.
		YES	NO	
Actual Class	YES	True Positive	False Negative	**FP**: - False Positive occurs when the result is envisaged as positive when it is actually negative.
	NO	False Positive	True Negative	**FN**: - False Negative occurs when the result is envisaged as negative when it is actually positive.

The performance of a binary classification test is statistically measured by precision and recall. The proportion of actual positives which are correctly recognized is calculated by Recall. The overall accuracy is calculated by using precision, recall and F-measure which are generally used to estimate the rare class prediction. It is advantageous to achieve a high recall devoid of loss of precision. Harmonic mean of precision and recall is called as F-measure. The equation used for Recall, precision, F-measure and overall accuracy is given below

$$\text{Precision} = TP/(TP+FP) \qquad \text{Recall} = TP/(TP+FN)$$

$$\text{F-Measure} = (2*\text{Recall}*\text{Precision})/(\text{Recall}+\text{Precision})$$

$$\text{Accuracy} = (TN+TP)/(TN+TP+FN+FP)$$

5.3 Experimentation and results

This section describes the experimental results and performance evaluation of the proposed technique. as discussed earlier, we used a subset of KDD cup data called 10% KDD cup 99 dataset for experimental assessment. In the training, we consider 26114 data points and in the testing we consider 27112 data points. The confusion metrics are computed for both training and testing dataset and the attained results are tabulated in table-7 and table-8. The evaluation metrics are calculated and the experimental results obtained for training data set and testing data

set are also tabulated in table 9 and table 10 respectively. These evaluation metrics gives an overall performance categorization of the proposed system. From the table, it is clear that our proposed system performs very well. We have achieved 98.94% accuracy in case of DOS intrusion and in other cases such as PROBE, RLA and URA; we got 97.11%, 97.78% and 97.80% respectively. Figure-4 and figure-5 illustrates the graphical representation of performance evaluation of our proposed technique for training and testing datasets.

Table 7: Training phase confession matrix

ATTACK TYPE	TRAINING PHASE			
	True Negative (TN)	False Positive (FP)	True Positive (TP)	False Negative (FN)
DOS	12491	9	12499	1
PROBE	12491	9	2023	31
R2L	12491	9	39	0
U2R	12491	9	14	7

Table 8: Testing phase confession matrix

ATTACK TYPE	TESTING PHASE			
	True Negative (TN)	False Positive (FP)	True Positive (TP)	False Negative (FN)
DOS	12235	265	12500	0
PROBE	12235	265	1897	156
R2L	12235	265	25	13
U2R	12235	265	11	10

Table 9. Experimental results obtained for Training data set.

EVALUATION MATRIX	TRAINING PHASE			
	DOS	PROBE	R2L	U2R
Precision	0.9993	0.9956	0.8125	0.6087
Recall	0.9993	0.9993	0.9993	0.9993
F-measure	0.9996	0.9902	0.8966	0.6364
Accuracy	**0.9996**	**0.9973**	**0.9993**	**0.9987**

Table 10. Experimental results obtained for Testing data set.

EVALUATION MATRIX	TESTING PHASE			
	DOS	PROBE	R2L	U2R
Precision	0.9793	0.8774	0.8621	0.3995
Recall	0.9788	0.9788	0.9788	0.9788
F-measure	0.9895	0.9001	0.1524	0.7407
Accuracy	**0.9894**	**0.9711**	**0.9778**	**0.9780**

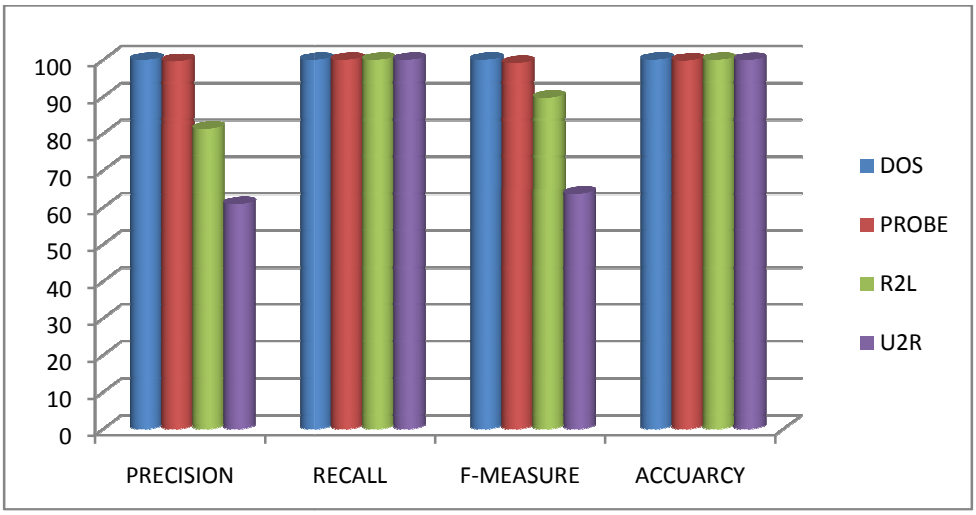

Figure 4: Performance evaluation of our proposed technique for Training dataset.

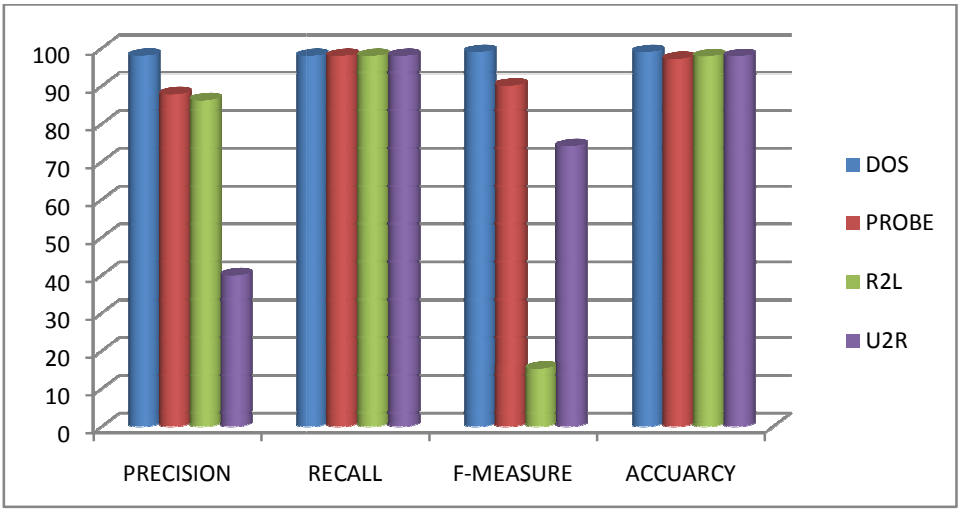

Figure 5: Performance evaluation of our proposed technique for Testing dataset.

5.4 Comparative analysis

This section illustrates the comparative analysis of the proposed technique with former methods. Table-11 demonstrates the comparison our praposed technique with the other state of art methods and shows the efficiency of our method. Our technique which use both the neural network and the SVM classifier performs well for all types of intrusions. From the table-11, it is apparent that the our technique has obtained a reliable peak scores for all types of intrusions especially for R2L and U2R attacks. In the case of DOS intrusion, we have attained 98.94% accuracy, for PROBE attack we reached 97.11% accuracy and for R2L and U2R attacks we have arrived at maximum accuracy value of 97.78% and 97.80% respectively, when compared to other methods. Figure-6 to Figure-9 shows the comparative results for all 4 types of attacks expressed in the graphical form. We have received the best results as we have employed fuzzy-neural networks to reduce the number of attributes of the data and by the utilization of radial basis function SVM in the final classification step.

Table 11. Accuracy comparison with existing methods

DIFFERENT METHODS	PROBE	DOS	R2L	U2R
KDD 99 Winner[36]	83.3	97.1	8.4	13.2
PNrule[37]	73.2	96.9	10.7	6.6
Multi-Class SVM[38]	75	96.8	4.2	5.3
Layered conditional random fields[39]	98.60	97.40	29.60	86.30
Columbia Model[40]	96.7	24.3	5.9	81.8
Decision tree[41]	81.4	60.0	24.2	58.8
BPNN[42]	99.3	98.1	48.2	89.7
Our Technique	**97.11**	**98.94**	**97.78**	**97.80**

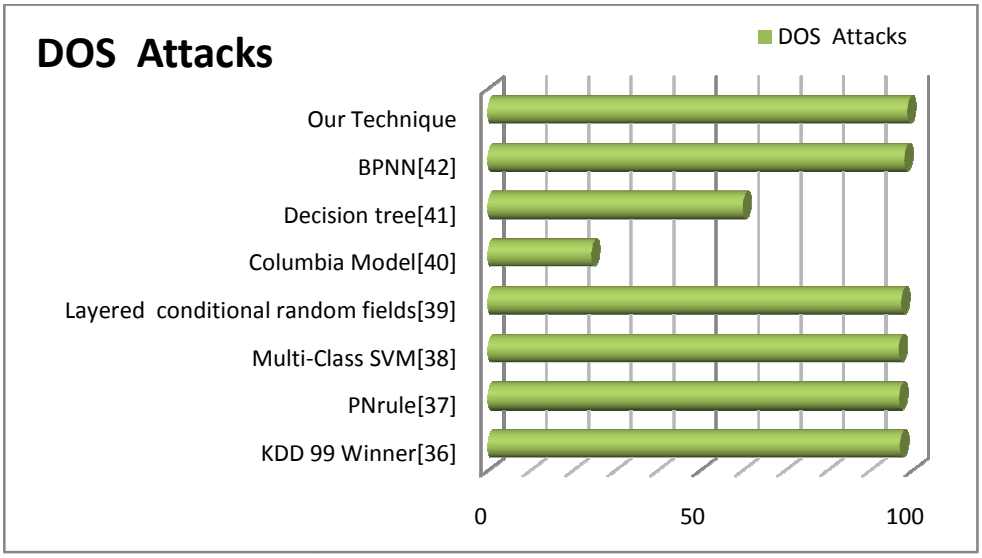

Figure 6: Performance comparison chart for DOS attack

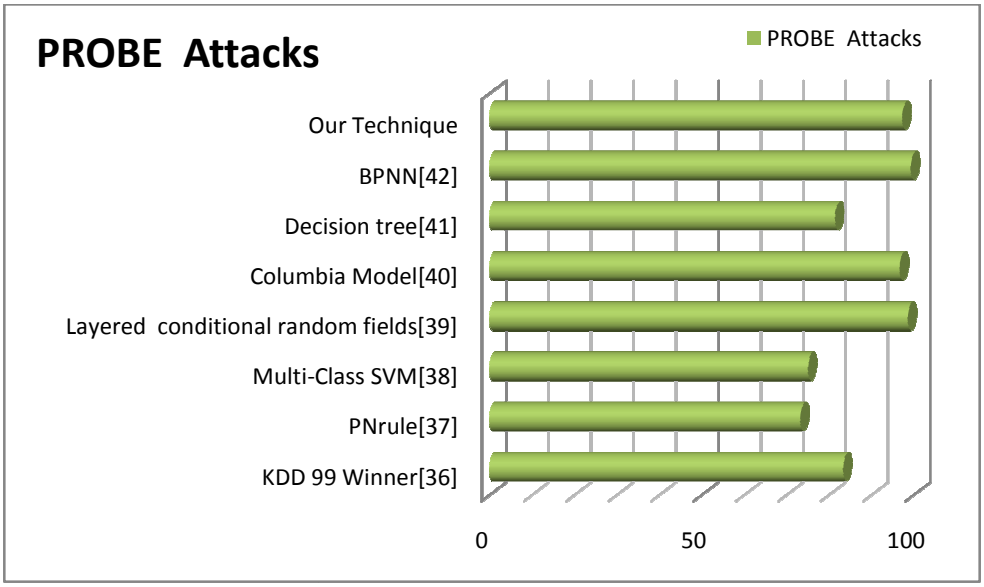

Figure 7: Performance comparison chart for PROBE attack

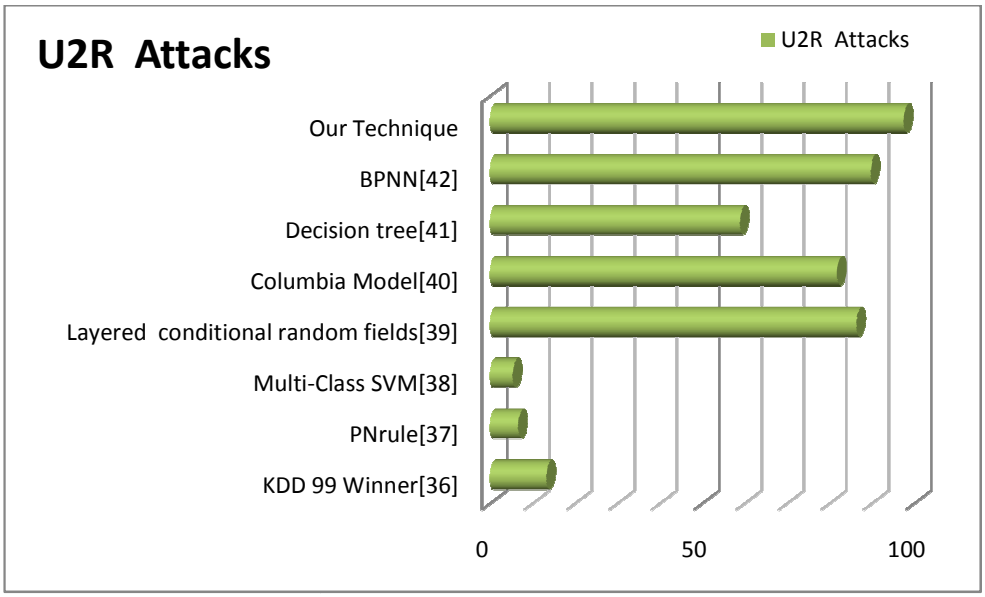

Figure 8: Performance comparison chart for U2R attack

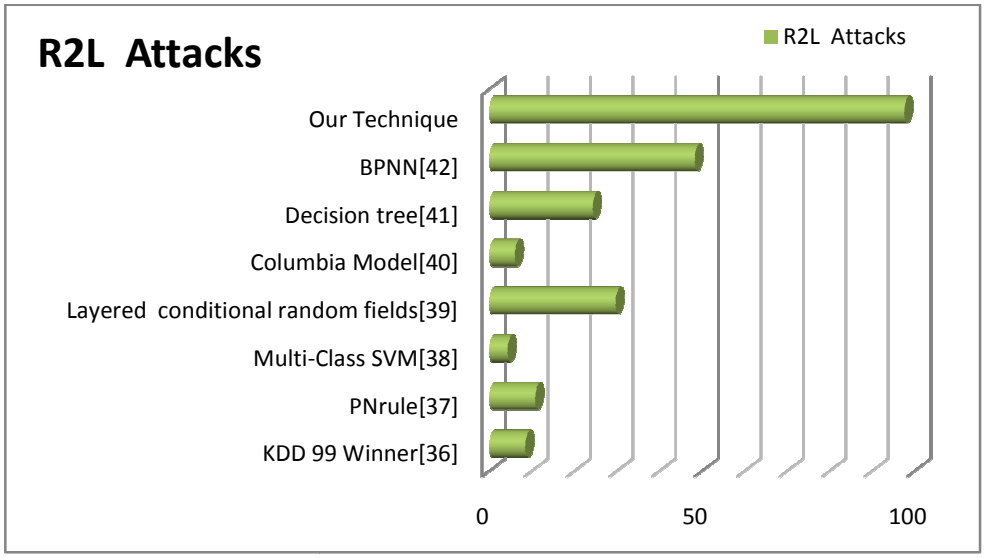

Figure 9: Performance comparison chart for R2L attack

6. CONCLUSIONS

This paper presents an efficient technique for intrusion detection by making use of fuzzy-neural networks and radial support vector machines. The proposed technique consists of initial clustering, fuzzy neural network, formation of SVM vector and the final classification using the radial SVM. KDD cup dataset 99 was used for experimental verification. Here, evaluation metrics consisted of three parameters, namely Precision, Recall, F-measure and Accuracy. For the metric calculation, we made use of True Positive, True Negative, False Positive and False Negative. For the evaluation results, we made use of about 10% of the full huge data set which amounted to about 27000 data points. The performance evaluation was made for both testing

and training for all types of intrusions such as DOS, PROBE, RLA and URA. Our technique achieved better results when compared to other existing techniques. The proposed technique has attained about 99% accuracy in the case of DOS attack and more than 97% accuracy for other types of attacks.

REFERENCES

[1] Ghanshyam Prasad Dubey, Prof. 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.

[2] Iftikhar Ahmad, Azween Abdullah and Abdullah Alghamdi, (2010) "Towards the selection of best neural network system for intrusion detection", International Journal of the Physical Sciences, vol.5, no.2, pp.1830-1839.

[3] J.T. Yao, S.L. Zhao, L. V. Saxton (2005), "A study on fuzzy intrusion detection", Proc. of Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security SPIE. 5812, pp. 23-30.

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

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

[6] Rung-Ching Chen and Su-Ping Chen (2008), "Intrusion detection using a hybrid support vector machine based on entropy and tf-idf", International Journal of Innovative Computing, Information and Control, vol. 4, no. 2, pp. 41301-424.

[7] Srinivas Mukkamala, Andrew H. Sung, Ajith Abraham and Vitorino Ramos(2004), "Intrusion detection systems using adaptive regression splines", In Proceedings of the 6th International Conference on Enterprise Information Systems, ICEIS, vol.3, pp.26-33.

[8] Snehal A. Mulay, P.R. Devale and G.v. Garje (2010), "Intrusion Detection System using Support Vector Machine and Decision Tree", International Journal of Computer Applications, vol 3, no 3, pp. 40-43.

[9] Deepak Tinguriya and Binod Kumar (2010), "A Effect Approach for Intrusion Detection System using Incremental SVM", BLB-International Journal of Science & Technology, vol.1, no.2, pp.127-134.

[10] Ajith Abraham, Ravi Jainb, Johnson Thomas and Sang Yong Han (2007), "D-SCIDS: Distributed soft computing intrusion detection system", Journal of Network and Computer Applications, vol. 30, pp. 81–98.

[11] D. E. Denning (1987), "An intrusion detection model," IEEE Transactions on Software Engineering, vol. 13, no. 2, pp. 222–232.

[12] S. Kumar and E. Spafford (1995), "A software Architecture to Support Misuse Intrusion Detection", 18th National Information Security Conference, pp.194-204.

[13] Hu W, Liao Y and Vemuri V (2003). "Robust support vector machines for anomaly detection in computer security", In Proceedings of the International Conference on Machine Learning andApplications, pp. 23-24.

[14] Peddabachigari S, Abraham A, Grosan C (2007), "Modelling intrusion detection system using hybrid intelligent systems. Journal of Network and Computer Applications, vol.30, no.1, pp. 114–132.

[15] Chengjie Gu, Shunyi Zhang and Xiaozhen Xue (2011), "Network Intrusion Detection Based on Improved Proximal SVM", Advances in Information Sciences and Service Sciences. Vol 3, no. 4, pp.132-140.

[16] S.Ganapathy, N.Jaisankar, P.Yogesh and A. Kannan (2011), “An Intelligent Intrusion Detection System Using Outlier Detection and Multiclass SVM”, International Journal on Recent Trends in Engineering & Technology, vol.05, no.01, 2011.

[17] Mansour Sheikhan and Amir Khalili (2010), " Intrusion Detection Based on Rule Extraction from Dynamic Cell Structure Neural Networks", Majlesi Journal of Electrical Engineering, Vol. 4, No. 4, pp. 24-34.

[18] Kyaw Thet Khaing (2010), “Enhanced Features ranking and Selection using Recursive Feature Elimination (RFE) and k-Nearest Neighbour Algorithms in Support Vector Machine for Intrusion Detection System", International Journal of Network and Mobile Technologies, Vol.1, No.1, pp. 8-14.

[19] Arvind Mewada, Prafful Gedam, Shamaila Khan and M. Udayapal Reddy (2010), “Network Intrusion Detection Using Multiclass Support Vector Machine”, Special Issue of IJCCT for International Conference [ACCTA-2010], Vol.1, No.2,3,4, pp.172-175.

[20] Snehal A, Mulay P.R, Devale G.V, and Garje(2010), “Intrusion Detection System Using Support Vector Machine and Decision Tree”, International Journal of Computer Applications, Vol.3, No.3, pp.40–43.

[21] Gang Wang, Jinxing Hao, Jian Ma and Lihua Huang (2010), "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering", Expert Systems with Applications, pp. 1-8.

[22] Muna Mhammad T. Jawhar and Monica Mehrotra (2010), “Design Network Intrusion Detection System using hybrid Neuro-fuzzy”, International Journal of Computer Science and Security, Vol.4, No.3, pp.285-294.

[23] Ghadiri A and Ghadiri N (2011), " An Adaptive Hybrid Architecture for Intrusion Detection Based on Fuzzy Clustering and RBF Neural Networks", In Proceedings of the Ninth Annual Communication Networks and Services Research Conference (CNSR), pp. 123 - 129.

[24] Pohsiang Tsai, Tich Phuoc Tran, Tony Jan and Xiaoying Kong, “Network Intrusion Detection using Machine Learning and Voting techniques”, Machine Learning, pp.267-290, 2010.

[25] “DARPA Intrusion Detection Evaluation Data Set” from http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1998data.html

[26] “KDDCup1999Data”from http://www.sigkdd.org/kddcup/index.php?section=1999&method=data

[27] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu and Ali A. Ghorbani (2009), "A detailed analysis of the KDD CUP 99 data set", in Proceedings of the Second IEEE international conference on Computational intelligence for security and defence applications, pp. 53-58, Ottawa, Ontario, Canada.

[28] C. Elkan(2000),“Results of the KDD’99 Classifier Learning”, SIGKDD Explorations, ACM SIGKDD.

[29] Mahesh kumar sabhanani and gursel Serpen (2003), “Application of Machine learning algorithms to KDD intrusion detection dataset within misuse detection context” In Proceedings of the International Conference on Machine Learning, Models, Technologies and Applications (MLMTA), Vol. 1, pp. 209-215.

[30] J. C. Dunn (1973): "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", Journal of Cybernetics, Vol. 3, pp.32-57.

[31] Jose Vieira, Fernando Morgado Dias and Alexandre Mota (2004), "Neuro-Fuzzy Systems: A Survey", In proceedings of 5th WSEAS NNA International Conferenceon Neural Networks and Applications.

[32] R. Jang (1992), “Neuro-Fuzzy Modelling: Architectures, Analysis and Applications”, PhD Thesis, University of California, Berkley.

[33] Cristianini, Nello and Shawe-Taylor, John (2000), "An Introduction to Support Vector Machines and other kernel based learning methods", Cambridge University Press, Cambridge.

[34] Li Zhuo, Jing Zheng, Fang Wang, Xia Li, Bin Ai, Junping Qian (2008), "A Genetic Algorithm Based Wrapper Feature Selection Method For Classification Of Hyper spectral Images Using Support Vector Machine", The International Archives of the Photogrammetric, Remote Sensing and Spatial Information Science, Vol. XXXVII, No. B7, pp.397-402.

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

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

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

[38] T. Ambwani (2003), "Multi class support vector machine implementation to intrusion detection," in Proc. of IJCNN, pp. 2300-2305.

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

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

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

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

[43] A.M. Chandrasekhar and K. Raguveer (2012), "fusion of multiple data mining techniques for Effective network intrusion detection – A contemporary approach", in proceedings of 5th ACM International conference on security of information and networks (SIN 2012), pp. 33-37.