Intrusion Detection System Using Kernel FCM Clustering and Bayesian Neural Network

Karthik G1
Research Scholar, Department of Computer Science Engineering, V.M.K.V Engineering College, Vinayaka Missions Research Foundation Deemed University, Salem, Tamil Nadu, India.

Nagappan A2
Principal, V.M.K.V Engineering College, Vinayaka Missions Research Foundation Deemed University, Salem, Tamil Nadu, India.

Abstract- Data safekeeping and security has been a key concern in the rapidly growing computer systems and networks. One of the recent methods for identifying any abnormal activities staging in a computer system is carried out by Intrusion Detection Systems (IDS) and it forms a significant portion of system defence against attacks. Various methods based on Intrusion Detection Systems have been proposed in recent years. In this paper, Intrusion Detection System (IDS) based on Fuzzy Bisector- Kernel Fuzzy C-means clustering technique and Bayesian Neural Network is proposed. The system contains two steps namely clustering step and classification step. In clustering step, the input dataset is grouped into clusters with the use of Fuzzy Bisector- Kernel Fuzzy C-means clustering (FB-KFCM). In the classification step, the centroids from the clusters are taken for training in the Bayesian Neural Network. The proposed technique is implemented by JAVA PROGRAMMING using KDD CUP 99 dataset. The evaluation metric utilized is accuracy and comparative analysis is made to other techniques. The average accuracy value obtained is 93.91 which was better than other compared techniques. The high accuracy value shows the efficiency of the proposed technique.

Keywords:- Intrusion Detection System, Classification, Clustering, Fuzzy Bisector- Kernel Fuzzy C-means clustering (FB-KFCM), Bayesian Neural Network, KDD CUP 99.

I. INTRODUCTION

Due to the popularization of the Internet and local networks, intrusion events to computer systems are growing [3]. Because of increased network connectivity, computer systems are becoming increasingly vulnerable to attack. The general goal of such attacks is to subvert the traditional security mechanisms on the systems and execute operations in excess of the intruder's authorization. These operations could include reading protected or private data or simply doing malicious damage to the system or user files [4]. By building complex tools, which continuously monitor and report the activities, a system security operator can catch potentially malicious activities as they occur. Intrusion detection systems are becoming increasingly important in maintaining proper network security [5, 6 and 3]. An intrusion detection system (IDS) monitors networked devices and looks for anomalous or malicious behaviour in the patterns of activity in the audit stream [7]. Intrusion Detection System is used to monitor the events occurring in a computer system or network, analyse the system events, detect suspected intrusion, and then raise an alarm [3].

There are broadly two types of Intrusion Detection Systems namely Host-based Intrusion Detection System and Network based Intrusion Detection System. A Host based Intrusion Detection system has only host based sensors and a network based Intrusion detection system has network-based sensor [8]. Host-based technology examines events like what files were accessed and what applications were executed [9]. Network-based intrusion detection is the problem of detecting unauthorized use of computer systems over a network, such as the Internet [10]. A good intrusion detection system should be able to distinguish between normal and abnormal user activities [11]. This would include any event, state, content, or behaviour that is considered to be abnormal by a pre-defined standard [12]. Data mining-based intrusion detection systems can be classified according to their detection strategy. There are two main strategies such as misuse detection and anomaly detection [13]. Misuse detection, which uses patterns of well-known attacks or weak spots of the system to identify intrusions [14] and anomaly detection, which tries to determine whether deviation from the established normal usage patterns can be flagged as intrusions [14,7]. One major challenge in intrusion detection is that we have to identify the camouflaged intrusions from a huge amount of normal communication activities [15]. In order to detect intrusion activities, many Machine Learning (ML) algorithms, such as Neural Network [16], Support Vector Machine [17], Genetic Algorithm [18], Fuzzy Logic [19], and Data Mining [20], etc have been widely used for huge volume of complex and dynamic dataset to detect known and unknown intrusions. It is very important for IDSs to generate rules to distinguish normal behaviours from abnormal behaviour by observing dataset, which is the record of activities generated by the operating system that are logged to a file in chronologically sorted order [10].

To detect newly encountered attacks, various researches have been undertaken, which use data mining as the key component [21]. Data mining is the analysis of data to establish relationships and identify hidden patterns of data, which otherwise would go unnoticed. Many researchers have dwelled into the field of database intrusion detection in
databases using data mining [22]. Several data mining techniques have been applied for intrusion detection, where, K-Mean Clustering [23] is unsupervised data mining techniques for intrusion detection. K-Means is a popular partitional clustering algorithm for its simplicity in implementation, and it is commonly applied in diverse applications. The main drawbacks of the \( k \)-means algorithm are: the choice of the value of \( k \), the cluster result is sensitive to the selection of the initial cluster centroids and convergence to the local minimum. In order to overcome the difficulties of K-Means clustering, several authors put modifications on the K-Means clustering. In [24], modification to K-Means clustering algorithm has been proposed for intrusion detection. This modified K-Means clustering algorithm is called as Y-Mean clustering that is extensively used for detecting the intrusion behaviour.

On the other hand, many researchers have argued that Artificial Neural Networks (ANNs) that can improve the performance of intrusion detection systems (IDS) when compared with traditional methods. Artificial Neural Network (ANN) is one of the widely used techniques and has been successful in solving many complex practical problems. However, for ANN-based IDS, detection precision, especially for low-frequent attacks, and detection stability are still needed to be enhanced. Furthermore, some of the researchers utilized Self-Organizing Map (SOM) or Self-Organizing Feature Map (SOFM) that is a type of artificial neural network, trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map. Self-organizing maps are different from other artificial neural networks in the sense that they use a neighbourhood function to preserve the topological properties of the input space. By providing the better detection accuracy, some of the researchers combined ANN with the data mining approaches to solve the problem and help IDS achieve higher detection rate, less false positive rate and stronger stability.

**Contribution of the paper**

- Intrusion detection system based on Fuzzy Bisector-Kernel Fuzzy C-means clustering technique and Bayesian Neural Network is proposed in this paper.
- The proposed technique is implemented using JAVA PROGRAMMING employing KDD CUP 99 dataset.
- The evaluation metric utilized is accuracy and comparative analysis is made to other techniques such as Multi-class SVM, Layered Conditional Random Fields, Columbia Model, Decision Tree, etc.

The rest of the paper is organized as follows: A brief review of researches related to the proposed technique is presented in section 2. The proposed intrusion detection technique is presented in Section 3. The detailed experimental results and discussions are given in Section 4. The conclusions are summed up in Section 5.

**II. LITERATURE REVIEW**

In recent times, intrusion detection has received a lot of interest among the researchers because it is widely applied for preserving the security within a network. Here, we present some of the techniques for intrusion detection. G. Gowrisona et al. [1] designed an intrusion detection system to classify the network behaviour with less computational complexity of \( O(n) \). The KDD Cup99 is a bench mark data used here to achieve promising classification rate. To achieve high detection rate in Intrusion Detection System (IDS), Shingo Mabu et al [2], described a fuzzy class association rule mining method based on Genetic Network Programming (GNP). GNP is used to enhance the representation ability with compact programs derived from the reusability of nodes in a graph structure. The combined method is evaluated with KDD99Cup and DARPA98 databases and showed that it provides competitively high detection rates.

However, to overcome the network based anomalies detection issue, Latifur Khanet al. [28] has proposed a method, which was the combination of SVM and DGSOT that starts with an initial training set and expanded it gradually using the clustering structure produced by the DGSOT algorithm. They compared the proposed approach with the Rocchio Bundling technique and random selection in terms of accuracy loss and training time gain by using a single benchmark real data set. Due to the necessity of misuse and anomaly detection in a single system, M. Bahrololumet al. [25] proposed an approach to design the system using a hybrid of misuse and anomaly detection for training of normal and attack packets respectively. The utilized method for attack training was the combination of unsupervised and supervised Neural Network (NN) for Intrusion Detection System. By misuse approach known packets were identified fast and unknown attacks were also be detected.

For the importance of an efficient Intrusion Detection System, K.S. Anil Kumar and V. NandaMohan [26] proposed a combination of three techniques comprising two machine-learning paradigms. K-Means Clustering, Fuzzy Logics and Neural Network techniques were deployed to configure an effective intrusion detection system. This approach revealed the advantage of converging K-Means-Fuzzy-Neural network techniques to eliminate the preventable interference of human analyst in such occasions. Also, to improve the accuracy as well as efficiency of the Intrusion Detection System, Shekhar R. Gaddamet al.[27] presented “K-Means+ID3,” a method to cascade k-Means clustering and the ID3 decision tree learning methods for classifying anomalous and normal activities in a computer network, an active electronic circuit, and a mechanical mass-beam system. Results showed that the detection accuracy of the K-Means+ID3 method was as high as 96.24 percent at a false-positive-rate of 0.03 percent on MSD; the total accuracy was as high as 80.01 percent on MSD and 79.9 percent on DED.

To overcome network security issues and to find better method than SVM, M. Ektefaet al. [29] have presented intrusion detection using data mining techniques such as classification tree and support vector machines. Their result indicated, C4.5 algorithm is better than SVM in detecting
network intrusions and false alarm rate in KDD CUP 99 dataset. Rasha G. Mohammed Helali [30] has presented a survey on data mining based network Intrusion Detection System (IDS). They presented the features of signature based NIDS in addition to the current state-of-the-art of Data Mining based NIDS approaches. Intruder was one of the most publicized threats to security. Network Intrusion Detection Systems (NIDS) had become a standard component in network security infrastructures. They provided general guidance for open research areas and future directions. The intention of their survey was to give the reader a broad overview of the work that had been done at the intersection between intrusion detection and data mining.

III. PROPOSED INTRUSION DETECTION SYSTEM USING FB-KFCM CLUSTERING AND BAYESIAN NEURAL NETWORK

The main objective of this research is to develop effective network intrusion detection system by utilizing data mining and artificial intelligence techniques. In this paper, intrusion detection system, which uses Fuzzy Bisector- Kernel Fuzzy C-means clustering technique and Bayesian Neural Network, is proposed to have effective distinction between the relevant data and intruded data. The system contains two steps namely clustering steps and classification steps. The block diagram of the proposed intrusion detection system is given in figure 1.

![Block diagram of the proposed intrusion detection system](image)

Figure 1: Block diagram of the proposed intrusion detection system

3.1 Clustering Step

The input dataset given to the intrusion detection system normally comprise huge quantity of data, which makes the processing very complex, hectic and time consuming. Executing this large number of data can also lead to having poor results by the increase of errors. Hence, it will have marked effect on the efficiency of the system and ultimately leading to reduced quality intrusion detection system. To compact this problem, clustering technique is employed prior to classification. We have employed proposed Fuzzy Bisector- Kernel Fuzzy C-means clustering as the clustering technique, which have resulted in having good results.

a) Fuzzy C-Means

The proposed FB-KFCM is an extension to KFCM which itself is an extension to normally used FCM. Let the input data is represented by $z$, number of input data by $n$ and $\sigma$ be a real number greater than 1 representing the weighting coefficient. The center of the cluster is represented by $x_i$ and the number of clusters by $N_c$. Let $\mu_{ij}$ represent the degree of membership of $z_i$ in the cluster $j$. Fuzzy C- Means (FCM) clustering has the minimization objective function defined in eq.1,

$$F_m = \sum_{i=1}^{n} \sum_{j=1}^{N_c} \mu_{ij}^m \| z_i - x_j \|^2$$

(1)

In the process, initially arbitrary data points are assigned as centroids and subsequently, membership values of the data points with respect to the centroids are found out. The generalized formula for finding membership function value is given in eq.2,

$$\mu_{ij} = \frac{1}{\sum_{m=1}^{N_c} \left( \frac{\| z_i - C_m \|}{\| x_j - C_m \|} \right)^{2(\frac{1}{\sigma} - 1)}}$$

(2)

Afterwards, the updated centroid values are computed with the use of found out membership values. The centroid updation equation is given in eq.3,

$$x_j = \frac{\sum_{i=1}^{n} \mu_{ij} z_i}{\sum_{i=1}^{n} \mu_{ij}}$$

(3)

Based on the updated centroid values, membership values are again found out. This process is repeated in a loop process to have the final clusters. The loop contains updating the membership value $\mu_{ij}$ and center of the cluster centers $x_j$. The loop condition is defined in eq.4,

$$\text{max} \{\mu_{ij}^n - \mu_{ij}^m \mid < \lambda\}$$

(4)

Here, $\lambda$ has the value between 0 and 1. Hence, FCM would converge to a local minimum or a saddle point of $F_m$.

b) KFCM

The negative aspect of FCM is the fact that it does not come up with high-quality accurate results. This is overcome with the use of BF-KFCM. BF-KFCM employs KFCM with additional steps. KFCM differs from normal FCM with the use of kernel functions, which yield better results. Hence in KFCM, though the process is same as that of FCM, it differs in the objective function and the updation equations.

In KFCM, input data ($z$) is mapped into a higher dimensional space ($S$) represented by non-linear feature map function $\phi: z \rightarrow \phi(z) \in Z$. The objective function of KFCM is given by eq.5,

$$F_m = \sum_{i=1}^{n} \sum_{j=1}^{N_c} \mu_{ij}^m \| \phi(z_i) - \phi(x_j) \|^2$$

(5)

Where,

$$\| \phi(z_i) - \phi(x_j) \|^2 = G(z_i, z_i) + G(x_j, x_j) - 2G(z_i, x_j)$$

(6)
Here, \( G(a,b) = \phi(a)^T \phi(b) \) which is the inner product kernel function and in our case, we are considering Gaussian kernel function. Hence, we have:

\[
G(z_i,z_j) = G(x_j,x_j) = e^{-\|x_j-x_j\|^2/2\sigma^2} = \phi(z_i)^T \phi(z_j)
\]

(7)

Hence, the objective function can be rewritten as in eq.9,

\[
F_m = \sum_{i=1}^{N} \sum_{j=1}^{c} \mu_{ij} [1 - G(z_i,z_j)]
\]

(9)

Minimizing the objective function with respect to \( \mu_{ij} \), we get the updation equations for finding membership value \( \mu_{ij} \) and centroids \( x_j \) is given in eq.10,

\[
\mu_{ij} = \frac{\left( \frac{1}{1-G(z_i,x_j)} \right)^{\frac{1}{1-\alpha}} \sum_{j=1}^{c} \mu_{ij} G(z_i,x_j) s_j}{\sum_{j=1}^{c} \left( \frac{1}{1-G(z_i,x_j)} \right)^{\frac{1}{1-\alpha}} \sum_{j=1}^{c} \mu_{ij} G(z_i,x_j) s_j}
\]

(10)

c) Fuzzy Bisector-Kernel Fuzzy C-means clustering (FB-KFCM)

In FB-KFCM, fuzzy bisector is incorporated into the KFCM to obtain better and more accurate results. Fuzzy bisector proceeds with the predefined rules and splits the selected cluster into two. Selection of the cluster is based on the parameters of Minimum Squared Error (MSE) and number of data points in the cluster. The cluster formation is carried out in various stages and in each stage; one existing cluster is further divided into two clusters. Let the input dataset be represented by \( \{z_1, z_2, ..., z_{N_d}\} \), where \( N_d \) is the number of input data. After clustering the data would be grouped to form clusters represented by \( FC = \{FC_1, FC_2, ..., FC_N\} \), where \( N \) is the number of clusters. Each cluster \( FC_i (0 < i \leq N) \) would have certain data \( z_i \in FC_i \) from the input data set. Let the data inside the \( i^{th} \) cluster be represented by \( FC_i = \{r_{i1}, r_{i2}, ..., r_{ni}\} \), where \( nci \) is the number of data in the \( i^{th} \) cluster. Illustration of proposed BF-KFCM clustering is given in figure 2.

The process of forming the final clusters is carried out in various stages. If the numbers of clusters are to be formed is \( N \), then FB-KFCM will consist of \( N + 1 \) stages. In stage 1, the input data is split into two clusters with the use of KFCM. Let the input data be represented as \( Z \), the formed clusters as \( A_1 \) and \( A_2 \). In the next stage, a particular cluster is taken and further divided to form two more clusters so as to make 3 clusters in total. Selection of cluster, which is to be divided using KFCM is based on certain rules. For rule formation, two parameters of MSE and number of data points in the respective cluster are found out.

Mean Squared Error (MSE) for a cluster is found out by finding the Euclidean distances between the data points and the centroid. Let the data points in the \( i^{th} \) cluster be represented by \( d_i \) and let the number of data points in the cluster be \( N_i \), centroid of \( i^{th} \) cluster be represented by \( c_i \) then MSE is given in eq.11,

\[
MSE_i = \frac{1}{N_i} \sum_{k=1}^{N_i} |d_i - c_i|^2
\]

(11)

By computing the MSE and number of data points for each of the cluster \( A_1 \) and \( A_2 \), selection of the cluster to be split is made. The selection condition is that the cluster should have maximum number of points and minimum MSE. Let the number of data points in \( A_1 \) and \( A_2 \) be represented by \( N_{A1} \) and \( N_{A2} \). Let the MSE value of \( A_1 \) and \( A_2 \) be represented by \( MA_1 \) and \( MA_2 \). Hence, the conditions can be written as in eq.12 and eq.13,

\[
If (N_{A1} > N_{A2}) AND (MA_1 < MA_2), Select A_1
\]

(12)

\[
If (N_{A2} > N_{A1}) AND (MA_2 < MA_1), Select A_2
\]

(13)

In other cases, arbitrary selection is carried out between \( A_1 \) and \( A_2 \). In our illustration, we have chosen \( A_1 \) and are split to form \( B_1 \) and \( B_2 \) by the use of KFCM. Hence, the clusters in consideration are \( A_2, B_1 \) and \( B_2 \). Subsequently, in stage 2, one among the three clusters is selected and the
selected cluster is further divided with the use of KFCM. The selection of the cluster to be divided is based on MSE and number of data points. Let the number of data points in B1 and B2 be represented by \( NB1 \) and \( NB2 \). Let the MSE value of B1 and B2 be represented by \( MB1 \) and \( MB2 \). The selection is based on the following conditions:

Select \( A2 \), if \( NA2 = \text{Maximum} (NA1, NB1, NB2) \)
\[
\text{AND } MB2 = \text{Minimum} (MB1, MB2) \]
Select \( B1 \), if \( NB1 = \text{Maximum} (NA1, NB1, NB2) \)
\[
\text{AND } MB1 = \text{Minimum} (MB1, MB2, MB2) \]
Select \( B2 \), if \( NB2 = \text{Maximum} (NA1, NB1, NB2) \)
\[
\text{AND } MB2 = \text{Minimum} (MB1, MB2, MB2) \]

For other cases, any of the three clusters is selected. In our illustration, we have selected A2 and are divided to clusters C1 and C2. Hence, the clusters in consideration are B1, B2, C1 and C2. In the third stage, respective cluster to be divided by KFCM is found out as in the earlier stages. In generalizing, suppose the clusters in the \( i^{th} \) stage are represented as \( C1, C2, ..., C_K \). The number of data points in the clusters is represented as \( N1, N2, ..., N_K \) and MSE of the clusters are represented as \( M1, M2, ..., M_K \), selection of the cluster which is to be divided can be defined by the rule:

Select \( C1 \), if \( N1 = \text{Maximum} (N1, N2, ..., N_K) \)
\[
\text{AND } M1 = \text{Minimum} (M1, M2, ..., M_K) \]
Select \( C2 \), if \( N2 = \text{Maximum} (N1, N2, ..., N_K) \)
\[
\text{AND } M2 = \text{Minimum} (M1, M2, ..., M_K) \]

In the illustration example, C2 is selected in stage 3 and divided to form D1 and D2. In stage 4, B2 is selected and subsequently, the process is repeated to have the required clusters. The process of dividing the selected cluster by the use of KFCM is carried out for all the \( N \) stages to form \( N+1 \) clusters represented in eq.14,

\[
FC_i, 0 < i \leq (N + 1) .
\]

After having the required number of clusters, the centroid from each of cluster is calculated and is taken for further process. That is instead of all the data inside the cluster, only the centroid is taken and given to learning process. As all data points inside the cluster are more or less the same, taking centroid will serve the purpose of representing all data inside a cluster. This would lessen the time of computation in further processes and also would reduce the complexity and risks. Let the data inside the \( i^{th} \) cluster be represented by \( FC_i = \{ ri_1, ri_2, ..., ri_{NCi} \} \).

Hence the centroid (\( \text{Cen}_i \)) of \( i^{th} \) cluster is found out in eq.15,

\[
\text{Cen}_i = \frac{1}{ \sum_{j=1}^{NCi} ri_j } .
\]

Where, \( ri \) is the represented \( i^{th} \) cluster.

Hence, we have converted to large bulky dataset into small number of data for better handling, learning and easier computation.

3.2 Classification Module

The centroids obtained after the clustering process are used for the learning or training process of Bayesian Neural Network. The input to the Bayesian Neural Network would be centroids of the clusters given in eq.16,

\[
\text{Cen}_i; 0 < i \leq (N + 1) .
\]

a) Neural Network

Artificial Neural Networks provide a powerful tool for classification and has been used in a broad range of areas. The latest enormous research activities in neural classification have recognized that neural networks are a gifted substitute for a variety of traditional classification methods. The benefit of neural networks lies in the subsequent theoretical facets. First, neural networks are data driven self-adaptive methods in which they can fine-tune themselves to the data exclusive of any clear specification of functional or distributional form for the unique model. Second, they are universal functional approximators in which neural networks can approximate whichever function with random accuracy. Neural networks are nonlinear models, which makes them stretchable in modelling real world intricate relationships. Neural networks are able to approximate the subsequent probabilities, which offer the basis for setting up classification rule and performing statistical analysis.

In general, the neural network consists of three layers named as input layer, hidden layer and the output layer. The neural network works making use of two phases, one is the training phase and the other is the testing phase. In training phase, the network is trained under large data base. In our case, the centroids found out after the clustering is fed as the training data. Initially, the nodes are given random weights. As the output is already known in the training phase, the output obtained from the neural network is compared to the original and weights are varied using algorithms so as to reduce the error. Normally back-propagation algorithms are employed in Neural Networks. In the testing phase, the input test data is fed to the trained neural network having particular weights in the nodes and the output is calculated so as to find if intruded or not. Figure 3 shows the general block diagram of the neural network.
b) **Bayesian Neural Network**

Inclusion of Bayesian concept has the advantages of better learning for Neural Networks. Bayesian based learning is based on two properties. One is that background knowledge is utilised in selecting prior probability distribution for model parameters. Second is the fact that prediction are made with respect to the posterior parameter distribution obtained by updation of the prior function. These two properties are in built into the Neural Network to have the Bayesian Neural Network.

Considering a single hidden layer based Neural Network, we can see that the output can be mathematically written in eq.17,

\[ y_j(x) = h_i + \sum_k \omega_{ki} b_k(x) \]  

(17)

Where, \( b_k(x) = \tan \theta_j + \sum_j \sigma_{jk} x_j \)  

(18)

Here \( x \) represents the input vector, \( y_j(x) \) denotes the output value function, \( \omega_{ki} \) gives the weight from hidden layer \( k \) to output \( i \) and \( \sigma_{jk} \) gives the weight from input \( j \) to hidden layer \( k \). The network can be used to define probabilistic model for classification. This is carried out by using the network output to define the target \( z_i \), given the input vector \( x \). For classification, where target is a single discrete value for possible class outputs, the probability can be defined in eq.19,

\[ P(z = i | x) = \sum_j \gamma_{j(i)} \]  

(19)

The bias and the weights present in the Neural Network are based on the training inputs, which contains the input values and the corresponding output values. This can be represented by: \( (x^{(0)}, z^{(0)}), 0 < i < n \) where, \( n \) is the total number of inputs. The weights and the bias are updated based on the error in the network. This error is computed as the squared sum of difference between the network outputs and the target outputs. The updation is such way as to minimize the error in the system. This minimization is equivalent to likelihood estimation for Gaussian noise method where minus log of likelihood is proportional to the sum of squared error.

In Bayesian approach to Neural Network, the objective is to find the predictive distribution for the target values in a new test case, given the input for that case, the input and the targets in the training cases. Then, the predictive distribution can be written in eq.20,

\[ P(z^{(n+1)} | x^{(n+1)}, z^{(1)}, \ldots, z^{(n)}) = \int P(z^{(n+1)} | x^{(n+1)}, \theta) P(\theta|x^{(1)}, z^{(1)}, \ldots, x^{(n)}, z^{(n)}) d\theta \]  

(20)

Where, \( \theta \) gives the network parameters like weight and bias. Posterior density for the parameters is proportional to product of prior and likelihood function, which can be represented in eq.21,
4.2 Evaluation Metrics

In testing phase, the testing dataset is given to the proposed technique to detect intrusion and obtained results are evaluated with the evaluation metrics for accuracy. True positive, true negative, false negative and false positive are found out to find the accuracy measure. Table 3 defines the terms for these. Let true positive be represented as $\alpha$, true negative be represented as $\beta$, false positive be represented as $\gamma$ and false negative be represented as $\lambda$. Accuracy (represented as $\delta$) can be defined as the proportion of the true results ($\alpha$ and $\beta$) in total results. Mathematically, it can define as:

$$\delta = \frac{\alpha + \beta}{\alpha + \beta + \gamma + \lambda}$$  \hspace{1cm} (22)

<table>
<thead>
<tr>
<th>Flag</th>
<th>normal or error status of the connection</th>
<th>Symbolic</th>
</tr>
</thead>
</table>

4.3 Comparative Analysis

In this section, our proposed technique is compared with other prominent techniques. For the purpose of detailed comparison, we compare proposed technique (BF-KFCM + Bayesian network) with existing technique (FCM + Bayesian network). The detailed analysis is taken for three cases. In the first case (case 8:2), eight samples out of ten are taken for testing and rest 2 for testing purpose. In the second case (case 7:3), seven samples out of ten are taken for testing and rest 3 for testing purpose. And in final case (case 9:1), seven samples out of ten are taken for testing and rest 3 for testing purpose. Table 4, 5 and 6 gives the accuracy values for the three cases and the corresponding plots are given in figures 4, 5 and 6.
Inferences form tables 4-6 and figures 4-7

• Comparison of the proposed technique (BF-KFCM+ Bayesian network) is made with the existing technique (FCM+ Bayesian network).

• Figure 4 and table 4 gives the accuracy values for case 8:2 for varying cluster size. Figure 5 and table 5 gives the accuracy values for case 7:3 for varying cluster size. Figure 6 and table 6 gives the accuracy values for case 9:1.

• Accuracy values are taken for different cluster sizes of 140, 160, 180 and 200.

• In all cases we can observe that our proposed technique has achieved better accuracy value when compared with existing technique.

• Average accuracy value for case 1 was about 86.8 for existing and 93.9 for proposed technique. Average accuracy for case 2 was about 86.7 for existing and 95.0 for proposed technique. Average accuracy for case 3 was about 86.7 for existing and 92.2 for proposed technique.

• Average accuracy values for proposed and existing are given in figure 7. Total average accuracy for existing was 86.77 while for proposed technique it was 93.91.

• These values show the efficiency of the proposed technique by achieving better accuracy values.

We also compare our proposed technique with other techniques in the area. The comparison values are given in table 7 and figure 8. Comparison is made respect to KDD 99 winner, PN rule, Multi-class SVM, Layered Conditional Random Fields, Columbia Model, Decision Tree and BSPNN.

<table>
<thead>
<tr>
<th>Technique</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDD 99 winner</td>
<td>90.2</td>
</tr>
<tr>
<td>PNrule</td>
<td>85.6</td>
</tr>
<tr>
<td>Multi-class SVM</td>
<td>85.9</td>
</tr>
<tr>
<td>Layered Conditional Random Fields</td>
<td>90.1</td>
</tr>
<tr>
<td>Columbia Model</td>
<td>89.7</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>72.4</td>
</tr>
<tr>
<td>BSPNN</td>
<td>92.3</td>
</tr>
<tr>
<td>Our proposed technique</td>
<td>93.9</td>
</tr>
</tbody>
</table>

From table 7 and figure 8, we can infer that our proposed technique has performed well by obtaining high accuracy value.

V. CONCLUSION

Intrusion detection system based on Fuzzy Bisector-Kernel Fuzzy C-means clustering technique and Bayesian Neural Network is proposed in this paper. The system contains two steps namely clustering step, which uses Fuzzy Bisector-Kernel Fuzzy C-means clustering (FB-KFCM) and classification step, which uses Bayesian Neural Network. The proposed technique is implemented by JAVA PROGRAMMING using KDD CUP 99 dataset. The evaluation metric utilizes its accuracy and comparative analysis that is made for other techniques. Average accuracy value came about 93.91 which was better than other compared techniques. The high accuracy value shows the efficiency of the proposed technique.

REFERENCES


Ian Pei , Jiawei Han , Laks V. S. Lakshmanan, “Pushing Convertible Constraints In Frequent Itemset Mining”, Data Mining And Knowledge Discovery, Vol. 8, No.3, pp.227-252, May 2004.


