# Hybrid clustering and classification methods to improve intrusion detection

Mazhar B. Tayel, Mohamed R. M. Rizk, Sherine K. Mohamedeen.

**Abstract**— The principle of Intrusion Detection System (IDS) is the most critical part of networks security infrastructure, because there are diverse ways to compromise stability and security of the network. Different soft-computing based methods have been proposed in recent years for development of intrusion detection systems. Many researchers have argued that Artificial Neural Networks (ANNs) can improve performance of IDS, when compared with traditional methods. However, detection precision, especially for low-frequency attacks, still inaccurate for ANN-based IDS mechanisms. Others used fuzzy clustering algorithms because of the uncertainty nature of such attacks. This paper presents a solution for low frequency attacks using hybrid approaches relying on Possibilistic Fuzzy C-Means clustering (PFCM) technique and ANN.

The proposed technique is a three step methodology, the first step is to perform fuzzy clustering; either using the PFCM technique, or the K-Means Clustering (KMC) technique. The second step is to perform classifying using ANNs, two types of neural networks are used; either Feed Forward Neural Network with back propagation (FFNN) or Radial Basis Neural Network (RBNN) to decide which is better in terms of precision detection. Finally, a fuzzy aggregation module is employed to aggregate these results. Experimental results on the KDD CUP 1999 dataset show that the proposed approach, PFCM-RBNN, outperforms KMC with ANN, c- means clustering and neural network FC-ANN, BPNN and other well-known methods such as Decision tree, the naïve Bayes in terms of detection precision.

**Index Terms**— Artificial Neural Network, Intrusion Detection System, K-means fuzzy clustering, Possibilistic Fuzzy C-Means clustering.

## 1 Introduction

Intrusion detection is an important issue in network security. Detection precision is the main key indicator to evaluate IDS referring to accuracy for each class of attack and stability of detection. Recently, there are exhaustive efforts for improving the existing detection techniques, due to high false alarm, moderate accuracy and lacking in the performance of the single classifier [1],[2].

Clustering is a widely used approach in IDS. Among the clustering algorithms, the fuzzy approaches are found to be efficient. Dunn in 1974 was first introduced Fuzzy C-Means clustering model (FCM) then Bezdek in 1983 extended and generalized this model[3]. Since then, different improvements of the method and model are suggested by researchers.

Clustering differs from classification. It deals with unsupervised learning of unlabeled data. Clustering algorithms can be safely used on a data set without much knowledge of it. Moreover detection and handling of noisy data and outliers are relatively easier using clustering. Clustering provides the ability to deal with data having different types of variables such as a continuous variable that requires standardized data, a binary variable, nominal variable, ordinal variable and mixed variables.

On the other hand in classification is done on unlabeled data after a supervised learning on pre-labeled data. Artificial Neural Networks (ANN) are widely used in solving many complex practical problems of classifying. However, the main drawback of ANN-based IDS; is the lower detection precision, especially for low frequency attacks, Remote to Local (R2L), User to Root (U2R) [1],[4].

The main objective of this paper is to test detection precision for low-frequency attacks. A proposed method based on using a hybrid model with a clustering algorithm and a classifying algorithm.

## 2 Design of Proposed Model

The proposed model is divided into four stages:

1. Dividing a dataset into training and testing sets,
2. Fuzzy clustering,
3. Artificial Neural Networks, and
4. Fuzzy Aggregation.

Fig 1 represents the four stages of the proposed model. The details of each stage will be discussed in the following subsections

- *Mazhar B. Tayel Electrical Engineering Department, Faculty of Engineering, Alexandria University. E-mail: profbasyouni@gmail.com*
- *Mohamed R. M. Rizk Electrical Engineering Department, Faculty of Engineering, Alexandria University. E-mail: mrmrizk@ieee.com*
- *Sherine K. Mohamedeen Electrical Engineering Department, Faculty of Engineering, Alexandria University. E-mail: en.sherine@gmail.com*

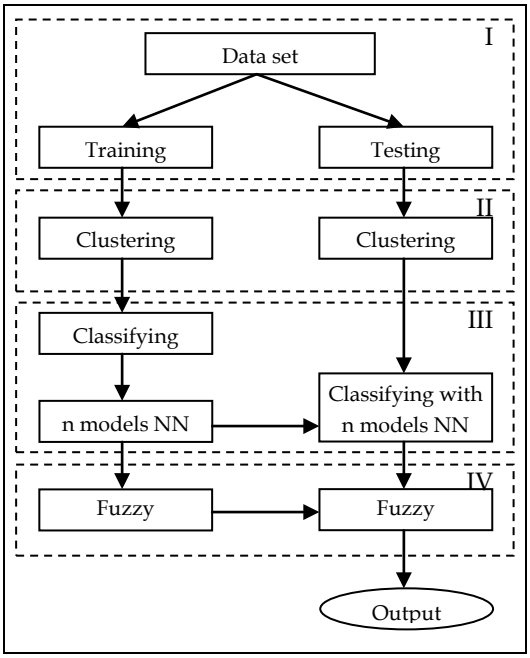



**Fig 1: Block diagram of the four stages proposed model.**

## 2.1 Dataset preparation:

The KDD Cup 1999 dataset is used for evaluating the intrusion detection methods. The KDD Cup 1999 training dataset consists of approximately 4,900,000 single connection vectors each of which contains 41 features and is labeled as either normal or an attack, with exactly one specific attack type. Also, the KDD Cup 1999 dataset contains a total number of 24 training attack types, with an additional 14 types of attacks in the test data. These attacks can be divided into 4 groups [5],[6]. Table 1 shows the list of attacks.

**Table 1: List of attacks.**

| The group of attacks | Types of attack |
|---|---|
| DoS | back, land, Neptune, pod, smurf, teardrop |
| R2L | ftp_write, guess_passwd, imap, multihop, phf, spy warezclient, Warezmaster |
| U2R | buffer_overflow, load module Perl, rootkit |
| Probe | IP_sweep, n_map, port_sweep, satan |

## 2.2 Fuzzy clustering

There are two main types of clustering, Hard Clustering and Soft Clustering [7]. Fuzzy clustering is a process of allocating membership levels and using them to assign data elements to one or more clusters.

The Hard Approaches algorithm has a drawback that clustering result is sensitive to selection of initial cluster centers and may converge to local optima. The cluster centers decide the local optimal solution in vicinity of the initial solution of K-means and the partition result of the dataset [8]. The Soft Approaches algorithm is a stochastic optimization technique generate good initial cluster centers to find an optimal optimal solution for numerical and qualitative problems. K-Means algorithm (KMC) is an example of Hard Clustering approach while Fuzzy C-Means (FCM) is an example of Soft Clustering approach [8].

### *2.2.1 Possibilistic Fuzzy C-Means clustering*

Pal[9] proposed a new and improved algorithm called PFCM. He relaxed the constraint on the typicality values but retain the column constraint on the membership values. The PFCM is a good clustering algorithm to perform classification tests because it possesses capabilities to give more importance to membership values. The PFCM is a hybridization of PCM and FCM that often avoids various problems of PCM, FCM, and FPCM.

## 2.3 Neural network:

ANN module aims to learn the pattern of every subset. In this paper, two types of neural networks are used; the Feed Forward Neural Network with back propagation, and The Radial Basis Neural Network to classify the clustered data.

### *2.3.1 Feed Forward Neural Network (FFNN)*

There are two types of phases used in multi-layer FFNN, the Forward Phase is used to fix the free parameter in the network and finish with the computation of an error signal. In the Backward Phase, the error signal is propagated through the network. During this phase, adjustments are applied to the free parameters of the network so as to minimize the error in a statistical sense [10].

### *2.3.2 Radial Basis Neural Network (RBNN)*

The Radial Basis Function (RBF) is applied to the distance to compute the weight (influence) for each neuron [6].

Weight = RBF (distance)

The following parameters are determined by the training process:

- The number of neurons in the hidden layer.
- The coordinates of the center of each hidden-layer RBF function.
- The radius (spread) of each RBF function in each dimension.
- The weights applied to the RBF function outputs as they are passed to the summation layer.

The RBF methods have been used to train the proposed networks [6].

## 2.4 Fuzzy aggregation

The aim of fuzzy aggregation module is to aggregate different ANN's result and reduce the detection errors. The errors are nonlinear, so another ANN is introduced to learn the errors as follows in order to achieve the objective:

**Step 1:** Let the whole training set TR as data to input every $ANN_i$ and get the outputs.

**Step 2:** Form the input for new ANN $Y_{input}$ by multiplying each output by the corresponding membership grade based on the corresponding cluster.

**Step 3:** Train the new ANN. Using $Y_{input}$ as input and use the whole training set TR's class label as output to train the new ANN.

Through above three steps, the introduced ANN can learn the errors which caused by the individual $ANN_i$ in ANN

module.

During the stage of testing, work procedure of ANN module and fuzzy aggregation module is similar to the above. Firstly, calculate the membership grade, based on the cluster centers$C^{TR}$. Then, respectively, the output, $Y^{TS}$, can be gotten using ANN module and fuzzy aggregation module.

## 3 EXPERIMENT

### 3.1 Data preparation

In this experiment, randomly select 18,285 records, similar to prior research [1]. The Probe, R2L, and U2R attack classes were totally selected because of their low portion in the KDD dataset. Three-thousand normal connections (records) and 10,000 DOS connections were randomly selected. For the testing step, the KDD testing set was used.

### 3.2 Evaluation criteria

Several metrics have been designed to measure the effectiveness of IDS. These metrics are defined in terms of types of correct or erroneous classifications that IDS can make. The confusion matrix represents true and false classification results

- A true positive indicates that the intrusion detection system detects precisely a particular attack having occurred.
- A true negative indicates that the intrusion detection system has not made a mistake in detecting an abnormal condition.
- A false positive indicates that a particular attack has been detected by the intrusion detection system but that such an attack did not actually occur.
- A false negative indicates that the intrusion detection system is unable to detect the intrusion after a particular attack has occurred.

In spite of representational power of the confusion matrix in classification, it is not a very useful tool for the sake of comparison of the IDSs. To solve this problem, different performance metrics are defined in terms of the confusion matrix variables [11].

- **Precision (PR):** It is the fraction of data instances predicted as positive that are actually positive [12].

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad (1)$$

- **Recall:** This metric is the proportion of intrusive actions that are classified as intrusive; namely, the percentage of the real attack instances covered by the classifier. Consequently, it is desired for a classifier to have a high recall value [12].

$$Recall = \frac{TP}{TP + FN} \qquad (2)$$

- **F-value (FV):** For a given threshold, the FV is the harmonic mean of the precision and recall at that threshold [13].

$$FV = \frac{(1 + \beta^2) * Recall * Precision}{\beta^2 * (Recall + Precision)} \qquad (3)$$

where β corresponds to the relative importance of precision versus recall and is usually set to 1.

### 3.3 Results and discussions

In the following experiments, each item is described by 41 features which form a vector. In the data set, some features are continuous and some are nominal. Since the clustering and classification algorithms require continuous values, these nominal values were first converted to continuous values. For the fuzzy clustering module, divide the training set into six subsets twice; the first time using PFCM clustering, while the other using k-means clustering. A stand three-layer network is used for ANN module and fuzzy aggregation module in the given experiments. There are 41 nodes for ANN module, in the input layer. Classify each subset of the clustered data twice, one time by FFNN, while the other time by RBNN.

For fuzzy aggregation module, five nodes are monitored, equivalent to the number of attacks i.e., Normal, DoS, Probe, R2L and U2R. The number of hidden nodes (hn) was determined by the empirical formula

$$\text{hn} = \sqrt{I + O} + \alpha \qquad (\alpha = 1 - 10), \qquad \textbf{(5)}$$

where $I$ is the number of input nodes, $O$ is the number of the output nodes and α is a random number [14]. In the given experiment, the complexity of intrusion detection α is considered equal to 10. Thus the structure of ANN in ANN module and fuzzy aggregation module are referred as [41; 18; 5] and [5; 13; 5], respectively. A sigmoid transfer functions are used for the input and hidden nodes, while a linear transfer function is used for the output node.

Ten experiments are performed by randomly selecting data according to the sampling rules, and then we compared the results with the BPNN and other well-known methods such as the decision tree and the naïve Bayes. The average results of the experiments are shown in Table 2 and the bar diagram in Figure 2.

**Table 2: Performance of previous methods in normal data analysis.**

| Method / Metric | Decision tree[1] | Naïve Bayes[1] | BPNN[1] | FC-ANN[1] |
|---|---|---|---|---|
| Precision % | 91.22 | 89.22 | 89.75 | 91.32 |
| Recall % | 99.41 | 97.70 | 98.20 | 99.08 |
| F-value % | 95.14 | 93.27 | 93.79 | 95.04 |

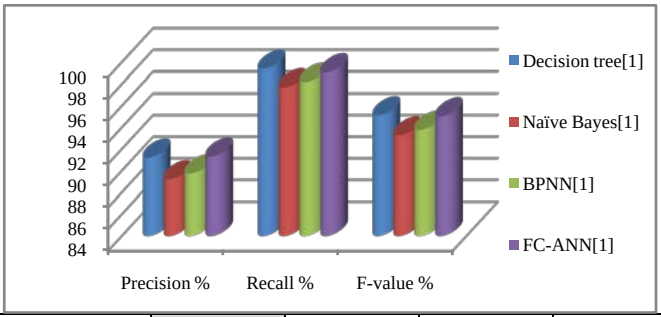



**Fig 2: Performance comparison of previous methods for normal data.**

From Fig 2 it is seen that the Decision tree method got the best results in terms of Recall and F- value. While the FC-ANN got the highest Precision percent than the Other 3 methods, but with no significance difference (0.1 %) compared to the Decision tree method. Accordingly, the Decision tree method is considered as a datum during the

normal data analysis.

**Table 3: The Performance of the used methods during normal data analysis.**

| Method / Metric | PFCM-FFNN | KMC-FFNN | PFCM-RBNN | KMC-RBNN | Decision tree[1] |
|---|---|---|---|---|---|
| Precision % | 99.46 | 98.04 | 99.56 | 99.26 | 91.22 |
| Recall % | 99.61 | 98.48 | 99.51 | 99.21 | 99.41 |
| F-value % | 99.23 | 98.26 | 99.54 | 99.17 | 95.14 |

Table 3 shows the results obtained during the experiment by applying the PFCM clustering, the FFNN, the RBNN and again by applying the KMC with the FFNN, and the RBNN while analyzing normal data.

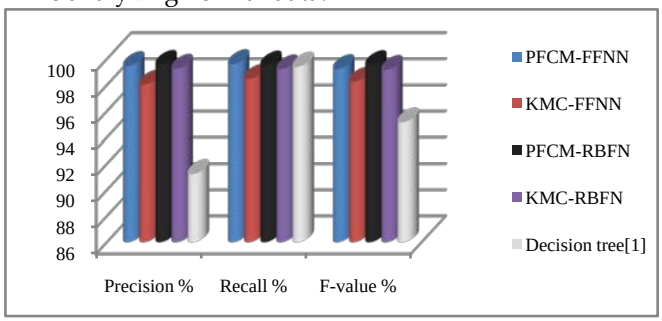



**Fig 3: Performance of the used methods in normal data analysis.**

From Fig 3 it is found that results were higher in the Precision, and F-value percent compared to the Decision tree method. While, there is a slight increase for the Recall percent. The PFCM-RBNN got the highest Precision and F-value percent, while the PFCM-FFNN got the highest Recall percent with a slight difference than the PFCM-RBNN 0.1%.

**Table 4 : The improvement percent in the used methods than the Decision tree method in the normal data analysis.**

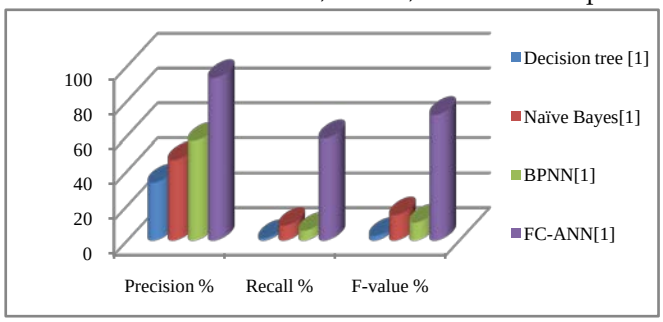
| Method / Metric | PFCM-FFNN | KMC-FFNN | PFCM-RBNN | KMC-RBNN |
|---|---|---|---|---|
| Precision % | 9.03 | 7.48 | 9.14 | 8.81 |
| Recall % | 0.20 | -0.94 | 0.10 | -0.20 |
| F-value % | 4.30 | 3.28 | 4.62 | 4.24 |

Table 4 shows the improvement percent of the proposed methods performance using the different metrics than the Decision tree method in the normal data analysis. The Precision percent increased by a minimum increase of 7.48% using the KMC-FFNN method, while the maximum increase was 9.14% by using the PFCM-RBNN method. The Recall percent decreased by 0.94% by using KMC-FFNN and decreased by 0.2% by using KMC-RBNN, while the Recall percent increased by 0.2 % by using the PFCM-FFNN method and 0.1% using the PFCM-RBNN method. The F-value generally increased with a minimum increase of 3.28% using the KMC-FFNN, and maximum increase of 4.62% using The PFCM-RBNN.

**Table 5: The performance of previous methods in R2L attack detection.**

| Method / Metric | Decision tree[1] | Naïve Bayes[1] | BPNN[1] | FC-ANN[1] |
|---|---|---|---|---|
| Precision % | 33.33 | 46.15 | 57.14 | 93.18 |
| Recall % | 1.43 | 8.57 | 5.71 | 58.57 |
| F-value % | 2.74 | 14.58 | 10.39 | 71.93 |

Table 5 shows the Precision, Recall, and F-value percent




using the Decision tree, Naive Bayes, BPNN, and FC-ANN for the R2L attack detection.

**Fig 4: Performance of previous methods in the R2L attack detection.**

From Fig 4 it is found that the FC-ANN got remarkably the highest results in detecting the R2L attack. In this experiment, the FC-ANN is referred to as the datum while analyzing the R2L attack detection.

**Table 6: Performance of the used methods in R2L attack detection.**

| Method / Metric | PFCM-FFNN | KMC-FFNN | PFCM-RBNN | KMC-RBNN | FC-ANN[1] |
|---|---|---|---|---|---|
| Precision % | 98.35 | 97.42 | 98.87 | 98.88 | 93.18 |
| Recall % | 99.84 | 99.30 | 99.98 | 99.92 | 58.57 |
| F-value % | 99.09 | 98.35 | 99.42 | 99.39 | 71.93 |

Table 6 shows the results obtained using The PFCM-FFFN, The PFCM-RBNN, The KMC-FFNN, and The KMC-RBNN compared to the FC-ANN and clearly, the introduced methods got significantly higher results than the FC-ANN.

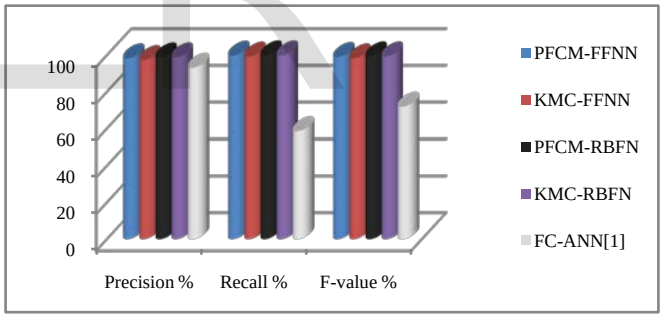



**Fig 5: Performance of the used methods in R2L attack detection.**

From Fig 5 it is found that there is a slight difference between the used methods in the R2L attack detection. The PFCM-RBNN got the highest Recall and F-value percent. The KMC-RBNN got the highest Precision percent with an insignificant difference (0.01%) compared to the PFCM-RBNN method.

**Table 7: The improvement percent in the used methods than the FC-ANN method in R2L attack detection.**

| Method / Metric | PFCM-FFNN | KMC-FFNN | PFCM-RBNN | KMC-RBNN |
|---|---|---|---|---|
| Precision % | 5.55 | 4.55 | 6.11 | 6.12 |
| Recall % | 70.46 | 69.54 | 70.70 | 70.60 |
| F-value % | 37.76 | 36.73 | 38.22 | 38.18 |

Table 7 shows the improvement percent of the detection performance of the used methods than the FC-ANN method in the R2L attack detection. The Precision percent is slightly increased by a minimum increase of 4.55% using the KMC-FFNN method, and a maximum increase of 6.12% using the KMC-RBNN.

The Recall percent is significantly increased by a min increase of 69.54% and a max increase of 70.70% using the PFCM-RBNN.

The F-value percent is also increased by a min increase of 36.73% using KMC-FFNN and a max increase of 38.22% using PFCM-RBNN.

**Table 8: Performance of previous methods in DoS attack detection.**

| Method / Metric | Decision tree[1] | Naïve Bayes[1] | BPNN[1] | FC-ANN[1] |
|---|---|---|---|---|
| Precision % | 99.84 | 99.69 | 99.79 | 99.91 |
| Recall % | 97.24 | 96.65 | 97.2 | 96.7 |
| F-value % | 98.52 | 98.15 | 98.48 | 98.28 |

Table 8 shows the performance of previous methods in the DoS attack detection.

**Fig 6: Performance of previous methods for the DoS attack detection.**

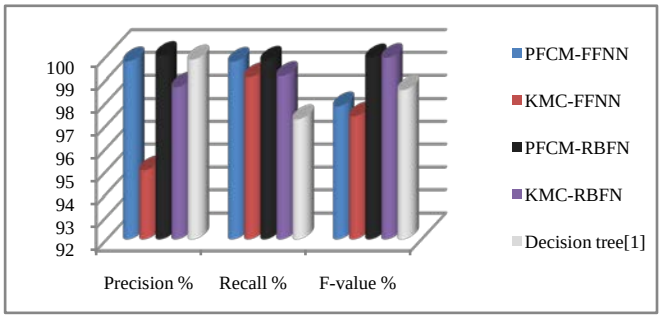
From Fig 6it is found that there are slight differences between the four methods. The Decision tree method got the Highest Recall and F-value percent. The FC-ANN got the highest Precision percent. Take the Decision tree method as the datum in this experiment due to the insignificant difference (0.07%) between the Decision tree and the FC-ANN in the Precision percent.

**Table 9: Performance of the used methods in DoS attack detection.**

| Method / Metric | PFCM-FFNN | KMC-FFNN | PFCM-RBNN | KMC-RBNN | Decision tree[1] |
|---|---|---|---|---|---|
| Precision % | 99.76 | 95.00 | 100 | 98.65 | 99.84 |
| Recall % | 99.74 | 99.07 | 99.79 | 99.12 | 97.24 |
| F-value % | 97.81 | 97.37 | 99.89 | 99.91 | 98.52 |

Table 9 shows results using the PFCM-FFFN, The PFCM-RBNN, The KMC-FFNN, and The KMC-RBNN methods compared to the Decision tree method.

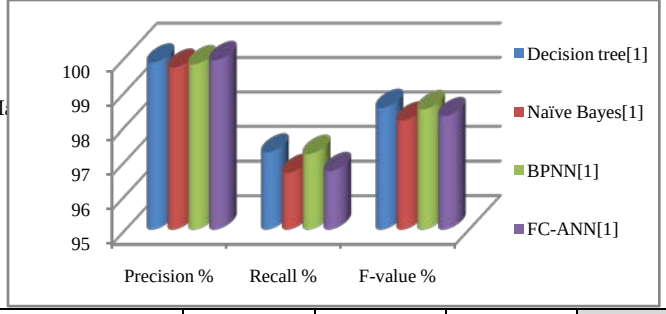
**Fig 6: Performance of the used methods in DoS attack detection.**




From Fig 7 it is found that the PFCM-RBNN method got the perfect Precision percent (100%) and the highest Recall percent. The KMC-RBNN got the highest F-value. Comparing the PFCM-RBNN and The KMC-RBNN methods, it is noticed that there is a slight difference in the F-value percent between the two methods.

**Table 10: The improvement percent in the used methods than the Decision tree method in DoS attack detection.**

| Method / Metric | PFCM-FFNN | KMC-FFNN | PFCM-RBNN | KMC-RBNN |
|---|---|---|---|---|
| Precision % | -0.08 | -4.85 | 0.16 | -1.19 |
| Recall % | 2.57 | 1.88 | 2.62 | 1.93 |
| F-value % | -0.72 | -1.17 | 1.39 | 1.41 |




As shown by Table 10, the Precision percent is slightly decreased by using KMC-FFNN by 4.85%, KMC-RBNN by 0.16% and PFCM-FFNN by 0.08%, while slightly increased by 0.16% using the PFCM-RBNN method.

The Recall percent is slightly improved by a min increase of 1.88% using KMC-FFNN and a max increase of 2.62% using PFCM-RBNN.

The F-value is slightly affected by a little decrease percent using KMC-FFNN, and PFCM-FFNN and by a little increase percent using PFCM-RBNN, and KMC-RBNN with the KMC-RBNN method scoring the max increase of 1.41%.

**Table 11: the Performance of previous methods during U2R attack detection.**

| Method / Metric | Decision tree[1] | Naïve Bayes[1] | BPNN[1] | FC-ANN[1] |
|---|---|---|---|---|
| Precision % | 50 | 25 | 50 | 83.33 |
| Recall % | 15.38 | 7.69 | 23.08 | 76.92 |
| F-value % | 23.53 | 11.76 | 31.58 | 80 |

Table 11 shows the performance comparison of previous methods during the U2R attack detection.

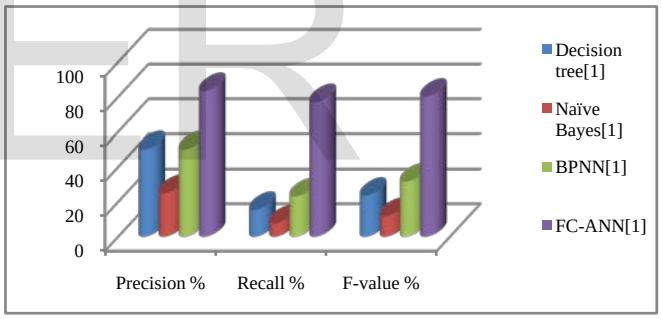



**Fig 7: Performance of previous methods in the U2R attack detection.**

From Fig 8 it is found that the outstanding performance of the FC-ANN method in detecting the U2R attack. Take the FC-ANN method as the datum level for the U2R attack detection in this experiment.

**Table 12: Performance of the used methods in U2R attack detection.**

| Method / Metric | PFCM-FFNN | KMC-FFNN | PFCM-RBNN | KMC-RBNN | FC-ANN[1] |
|---|---|---|---|---|---|
| Precision % | 98.32 | 95.16 | 100 | 97.86 | 83.33 |
| Recall % | 99.57 | 99.45 | 99.71 | 99.26 | 76.92 |
| F-value % | 98.17 | 97.26 | 99.85 | 98.88 | 80 |

Table 12 shows the results during the U2R attack detection compared to the FC-ANN method results.

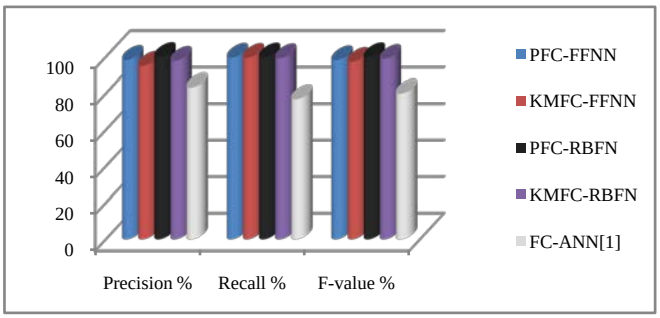



**Fig 8: Performance of the used methods in U2R attack detection.**

From Fig 9 it is found that the superior results of the methods used to the results of the FC-ANN method.

The PFCM-RBNN method again got the outstanding results with the perfect Precision percent and the highest Recall and F-value percent.

**Table 13: The improvement percent in the used methods than the Decision tree method in U2R attack detection.**

| Method / Metric | PFCM-FFNN | KMC-FFNN | PFCM-RBNN | KMC-RBNN |
|---|---|---|---|---|
| Precision % | 17.99 | 14.20 | 20.01 | 17.44 |
| Recall % | 29.45 | 29.29 | 29.63 | 29.04 |
| F-value % | 22.71 | 21.58 | 24.81 | 23.6 |

Table 13 shows that applying the proposed methods generally improves the ID metrics during the U2R attack detection. The Precision percent increased by a min increase of 14.20% using KMC-FFNN and a max increase of 20.01% using the PFCM-RBNN.

The Recall percent is considerably increased by a min increase of 29.04% using KMC-RBNN and a max increase of 29.63% using PFCM-RBNN.

The F-value is increased by a min increase of 21.58% using KMC-FFNN and a max increase of 24.81% using PFCM-RBNN.

**Table 14: Performance of previous methods in Probe attack detection.**

| Method / Metric | Decision tree[1] | Naïve Bayes[1] | BPNN[1] | FC-ANN[1] |
|---|---|---|---|---|
| Precision % | 50 | 52.61 | 60.94 | 48.12 |
| Recall % | 78.13 | 88.13 | 88.75 | 80 |
| F-value % | 60.98 | 65.89 | 72.26 | 60.09 |

Table 14 shows the comparison of previous methods in Probe attack detection.

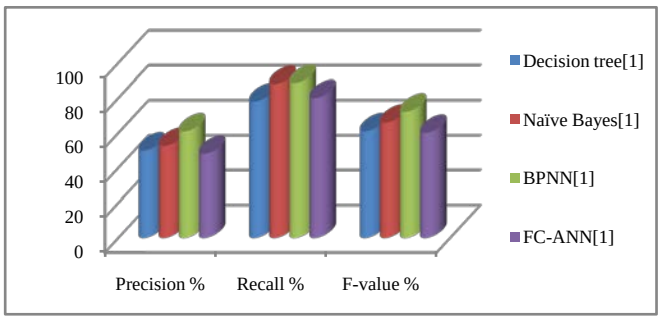



**Fig 9: Performance of previous methods in the Probe attack detection.**

From Fig 10 it is found that The BPNN got the best detection performance compared to the other methods. Refer to the BPNN results as datum.

**Table 15: Performance of the used methods in Probe attack detection.**

| Method / Metric | PFCM-FFNN | KMC-FFNN | PFCM-RBNN | KMC-RBNN | BPNN[1] |
|---|---|---|---|---|---|
| Precision % | 98.89 | 98.11 | 99.97 | 98.36 | 60.94 |
| Recall % | 99.66 | 99.58 | 99.98 | 99.15 | 88.75 |
| F-value % | 99.43 | 98.84 | 99.98 | 99.26 | 72.26 |

Table 15 shows that the results are significantly high compared to the BPNN results especially in the Precision percent in the Probe attack detection.

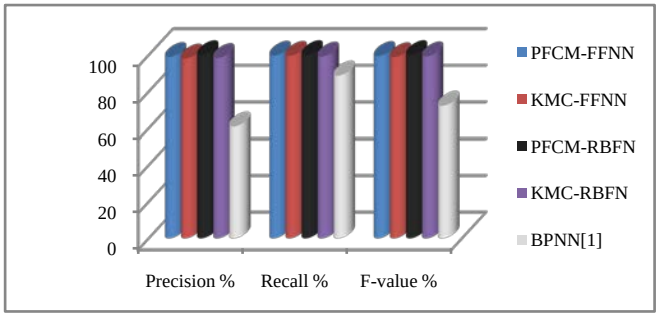



**Fig 10: Performance of the used methods in Probe attack detection.**

As shown in Fig 11, the four proposed methods got the highest results. There is no significant difference between the four methods. The PFCM-RBNN again got the highest Precision, Recall, and F-value percent.

**Table 16: The improvement percent in the used methods than the Decision tree method in Probe attack detection.**

| Method / Metric | PFCM-FFNN | KMC-FFNN | PFCM-RBNN | KMC-RBNN |
|---|---|---|---|---|
| Precision % | 62.27 | 60.99 | 64.05 | 61.40 |
| Recall % | 12.29 | 12.20 | 12.65 | 11.71 |
| F-value % | 37.60 | 36.78 | 38.36 | 37.37 |

Table 17 shows that the evaluation metrics are significantly increased using the proposed methods compared to the BPNN method.

The Precision percent is significantly increased with a min increase of 60.99% using KMC-FFNN and a max increase of 64.05% using PFCM-RBNN.

The Recall percent is increased with a min increase of 11.71% using KMC-RBNN and a max increase of 12.65% using PFCM-RBNN.

The F-value percent is considerably increased with a min increase of 36.78% using KMC-FFNN and a max increase of 38.36% using PFCM-RBNN.

Generally, it is noticed that The PFCM methods got higher results than the KMC methods as expected, as the soft clustering methods i.e. PFCM deals much better than the hard clustering method i.e. KMC in large datasets.

Also, it is noted that among the four proposed methods, the PFCM with RBNN got the highest results for all types of attacks, as the Radial Basis Networks deals better in the classification problems.

These results reveal that by introducing possibilistic fuzzy clustering with the ANN, detection precision can be enhanced. Especially to R2L and U2R attacks, the detection precision enhanced greatly.

## 4 CONCLUSION

In this paper, it has been found that clustering then classifying the data leads to superior results in IDS. In the proposed model, it has been found that using the soft clustering technique; PFCM with neural network got higher results even for low frequency attacks compared to the hard clustering technique KMC and the previously used methods. Comparing the performance of the FFNN and the RBNN networks, it is found that using the RBNN for classification leads to better results than FFNN.

## REFERENCES

[1] Wang, G., et al., A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. Expert Systems with Applications, 2010. 37(9): p. 6225-6232.

[2] McHugh, J., A. Christie, and J. Allen, Defending Yourself: The Role of Intrusion Detection Systems. Software, IEEE, 2000. 17(5): p. 42-51.

[3] Bezdek, J.C., R. Ehrlich, and W. Full, FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences, 1984. 10(2): p. 191-203.

[4] Singh, R.R., N. Gupta, and S. Kumar, To reduce the false alarm in intrusion detection system using self organizing map. International Journal of Soft Computing and Engineering (IJSCE), 2011(2231): p. 2307.

[5] Archive, T.U.K., KDD Cup 1999 Data. Information and Computer Science, 1999.

[6] Devaraju, S. and S. Ramakrishnan. Performance analysis of intrusion detection system using various neural network classifiers. in Recent Trends in Information Technology (ICRTIT), 2011 International Conference on. 2011: IEEE.

[7] Goswami, K., A Study on Some Variations of the Fuzzy C-Means Clustering Algorithm. 2010, JADAVPUR UNIVERSITY KOLKATA.

[8] Dutta, V., K.K. Sharma, and D. Gahalot, Performance comparison of hard and soft approaches for document clustering. International Journal of Computer Applications, 2012. 41(7).

[9] Pal, N.R., et al., A possibilistic fuzzy c-means clustering algorithm. Fuzzy Systems, IEEE Transactions on, 2005. 13(4): p. 517-530.

[10] Devaraju, S. and S. Ramakrishnan, Performance comparison of intrusion detection system using various techniques–A review. ICTACT Journal on Communication Technology, 2013. 4(3): p. 802-812.

[11] Munaiah, N., et al., Are Intrusion Detection Studies Evaluated Consistently? A Systematic Literature Review. 2016.

[12] Tang, L.-A., et al., A framework of mining trajectories from untrustworthy data in cyber-physical system. ACM Transactions on Knowledge Discovery from Data (TKDD), 2015. 9(3): p. 16.

[13] Gupta, K.K., B. Nath, and R. Kotagiri, Layered approach using conditional random fields for intrusion detection. IEEE Transactions on dependable and secure Computing, 2010. 7(1): p. 35.

[14] Haykin, S. and N. Network, A comprehensive foundation. Neural Networks, 2004. 2(2004).