European Journal of Molecular & Clinical Medicine

ISSN 2515-8260 Volume 7, Issue 08, 2020

INTRUSION PREDICTION AND DETECTION USING SUPPORT VECTOR MACHINE (SVM) AND ARTIFICIAL NEURAL NETWORK (ANN)

Mrs. Swetha M S[1] , Mr. Muneshwara M S[2], Dr. Chethan A. S[3], Mr. Shivakumara T[4] , Dr. Anil G N[5]

[1,2]Research Scholar BMS Institute of Technology and Management Yelahanka, Bengaluru-560064, Karnataka, India.

[3] Professor, Dept. of Mathematics BMS Institute of Technology and Management Yelahanka, Bengaluru-560064.

[4]. Asst. Prof, Dept. of MCA, BMS Institute of Technology and Management Yelahanka, Bengaluru-560064, Karnataka,

[5] Professor, Dept. of CSE, BMS Institute of Technology and Management Yelahanka, Bengaluru-560064, Karnataka,

E-mail: [1]swethams_ise2014@bmsit.in, [2]muneshwarams@bmsit.in, ,[3]aschethan@bmsit.in, [4]shivakumarat@bmsit.in, [5]anilgn@bmsit.in

Abstract---Intrusion detection and prevention systems are widely researched areas, rightly so being an integral part of network. As with all recent computing trends, Machine Learning and Deep Learning techniques have become extremely prevalent in intrusion detection and prediction systems security. The Intrusion detection system is used to detect and notify any malware activities and try to stop them. Soft computing techniques have the ability in learning data sets which is provided and it can also categories the packets or file coming through the network or any other source as normal and abnormal. Here, we will focus more on using Support Vector Machine (SVM) and Artificial Neural Network (ANN). In the proposed method, we are using SVM and ANN algorithms for the detection of malware; the data set is processed through SVM and ANN algorithms and compares their performances with respect to accuracy metrics. Since accuracy does not give a clear picture about how well classification algorithms perform, we have also measured and compared the performances of these two algorithms using AUC score. The AUC score is a value that ranges from 0 to 1 and closest to 1 will be considered as a better one. The results show that ANN can be implemented effectively for malware detection and is comparatively better than SVM.

Index Terms-- Support Vector Machine (SVM), **Artificial neural networks** ***(ANN), Area under the ROC Curve (AUC)***

I. INTRODUCTION

Malware detection can be broadly classified into Misuse/Signature Detection, Anomaly Intrusion Detection and Hybrid Detection. In Misuse/Signature Detection, each file is assigned with a signature or a hash which is added to a signature database. When a suspicious file is found, the program will look for patterns that will match with known family of malware. Due to constantly evolving malwares, this technique is not much used. Anomaly Intrusion Detection involves generating an alarm when there is a deviation from the normal behaviour that exceeds certain threshold. Certain machine learning and soft computing techniques are used for intrusion detection system to classify between normal and abnormal data. Hybrid Detection a blend of Signature Detection and Anomaly Intrusion Detection that can give better results. Machine learning and soft computing techniques are used here as well.

1.1 Support Vector Machine(SVM)

Support vector machines (SVM) which can be used for classification problems - support vector classification (SVC) and regression problems -support vector regression (SVR) is a supervised learning algorithm. It works well for smaller dataset as it takes too long to process for larger datasets. In this network dataset, we will be focusing on SVC.The main ideology behind SVM is to create a hyperplane and to classify the dataset given. To isolate the two classes of data points, there are numerous conceivable

Hyperplanes that could be picked. Our goal is to locate a plane that has the most extreme margin, i.e. the greatest separation between data points of the two classes. Expanding the margin enables the future data points to be classified with much more precession.

Hyperplanes are those that help in classifying the data. Data points or vectors that fall on either side if the hyperplane can be credited to different classes. Additionally, the hyperplane being built depends or relies on the number of attributes or features i.e. in other words depend on the number of independent features that define the dependent feature of a data set. If we have two independent features then our hyperplane will be three dimensional. If we have one independent feature then we will have a simple one dimensional hyperplane

Below is the figure from [3] that shows how hyperplane is built and can be used to classify the data points. H represents the hyperplane. H1 and H2 are the lines drawn parallel to hyperplane such that the distance between these two i.e. the margin is maximum.

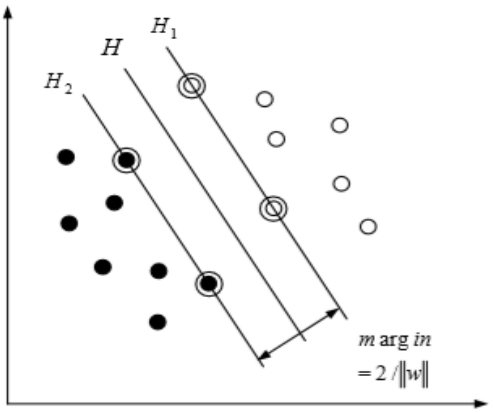

Fig 1: SVM Hyperplane

1.2 Artificial Neural Network (ANN)

Artificial Neural Networks (ANN) are multi-layer completely associated neural nets. They comprise of an input layer, numerous hidden layers, and an output layer. Each node in one layer is associated with each other node in the following layer. We make the system more profound by expanding the quantity of hidden layers.

A layer containing various nodes will receive weighted sum of the inputs fed to it and depending on the activation function, the layer's nodes get activated. Once the nodes get activated, these nodes act as inputs to the next layer. This happens from left to right i.e. from input to output layer. The final output from the output layer will be our predicted output. Training the network depends on number of times we propagate backwards and forwards to minimize the error.

We first need to train our model to really learn the weights, and the training method is as follows:

- We first assign random numbers/weights to all the nodes.
- Now once we assign the weights, we forward propagate for all the layers i.e. input, hidden and output layers where each of hidden layer's input will the output from the previous layer. Similarly the output layer's input will be output from the last hidden layer. The output that we get from the output layer is the predicted value.
- We then compare this predicted value with the actual value and calculate the error using loss function.
- Once we find out the error, we start backpropagation which acts exactly opposite to forward propagation and changes the weights accordingly in order to minimize the error.
- Below is a figure from [9] showing ANN with one input, two hidden and one output layer.

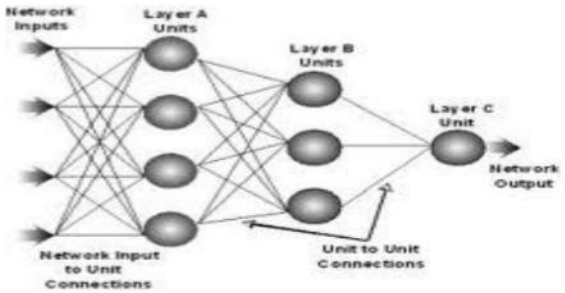

Fig 2: Artificial Neural Network

II. LITERATURE SURVEY

In [1], Intrusion detection System was implemented using SVM and Naïve Bayes and it was found that SVM outperformed Naïve Bayes and gave a better result. Intrusion detection and Intrusion prevention are required in current patterns. As ordinary occasions are principally subject to networks and information systems, intrusion detection and intrusion prevention are

fundamental. Numerous methodologies have been applied in intrusion detection systems. Among them AI assumes a crucial job. This investigation manages AI calculations like SVM and Naïve Bayes. It proposes while managing 19,000 examples SVM beats Naïve Bayes.

In [2], the author discusses that the network data is extremely enormous, heterogeneous, exceptionally shifting and imbalanced. The greater part of the accessible machine learning approaches created for consistently dispersed data and doesn't stress on these qualities of system data that this data isn't ordinarily distributed in equivalent classes. As volume of network traffic data is extremely gigantic and has enormous number of qualities it is for all intents and purposes difficult to run exemplary machine learning calculation on entire data. Or maybe sampling, feature extraction and selection should be performed. In any case, these activities may change in general character of data. Some great and exact methodologies for feature selection, extraction and sampling are required.

In [3], the author tells that one of the upcoming areas in network security is Intrusion Detection System. The author also tries to explain how using ANN will lead to overfitting and this issue is not a problem in Support Vector Machines. SVM has better classification rates than ANN. The support vector machine classification model is thus introduced and applied to intrusion detection system to classify the data into normal and the one being attacked. Support Vector Machine's tendency to self-learn makes it a better model to be put in use for Anomaly or Intrusion Detection System. The results show from this paper was very good using SVM classifier was a success and hence the author tells that this strategy can be used for network security purpose.

In [4], the authors in their research focus on genetic algorithm (GA) for creating the detection features. Support vector machine and artificial neural networks are used for detecting and classifying the network data. Along with these techniques the author makes use of genetic algorithm to blend and create a hybrid machine learning models. The result of this was that genetic algorithm with artificial neural network showed better detection and classification rates. In this experiment, the KDD cup 99 dataset is being used to classify the data into four types of network attacks. One of the most import technique that comes into the picture while applying such model is feature selection and it was seen that genetic algorithm with neural network require 18 features to show 100% detection rates while genetic algorithm with svm required 24 features to show 100% detection rates. The author also discusses on using different model along with genetic algorithm in future..

In [5], the authors begin with introducing to us about what an Intrusion Detection System is. They discuss about two types of intrusion detection system. The first one being misuse or signature detection system and the other being anomaly intrusion detection system. Later on, they focus more on anomaly based intrusion detection system. The model being used in this paper is neural networks. They gave a small diagram on how these neural networks work in general followed by how it can be used in intrusion detection systems.

In [6], targeting missing report rate and false alert rate which exist for the most part in the intrusion detection framework , this paper talks about an astute intrusion detection model. Based on the attributes of worldwide prevalence of genetic algorithm and region of nerve, the model optimizes the weights of the neural network utilizing genetic algorithm. Test results show that the wise way can improve the effectiveness of the intrusion detection.

In [7], the authors in this paper tell us how important the security is in network and how it is a big concern. Information or data is the most important resource of any company or organization and proving security for these against the hackers and attackers is a major concern. For the same purpose, Intrusion Detection Systems are being used to provide security and to classify and detect the data whether it is normal or malicious. In this paper, the author showed how neural networks along with backpropagation can be used for intrusion detection system. Although it did not perform well, the author plans on using elm technique in future for better detection rated.

In [8], the authors in this paper inform us that classifying the network data into normal and malicious has been their main ideology and main point for their examination. Various classification models are being utilized to effectively detect and classify the data into normal and malicious and attain high accuracy level by increasing true positive rates and decreasing false positive rates. In this paper, author has made used of different models for IDS like SOM, SVM, J48, back propagation using neural networks, and RBF. After thorough usage and data preprocessing, it was found out that J48 performs way more better than the rest of them in classifying and detecting the data into malicious and normal. In addition to it, principal component analysis was used as preprocessing step which improved the detection rates of SOM and back propagation.

III. PROPOSED SYSTEM

A smart Intrusion Detection System is the one that classifies the network data into "normal" and "attacked" data with AUC and accuracy values between 0.9 and 1. Using Soft Computing techniques like Support Vector Machine and Artificial Neural Network we will try building the best Intrusion Detection System with the best AUC and accuracy scores such that it classifies the network data into "normal" and "attacked" appropriately. We will try using some techniques like data pre-processing, feature selection and reduction using Principal Component Analysis (PCA), standardization and normalization in order to improve the AUC and accuracy scores of our model. We will also try comparing the results of these two methods based on their AUC and accuracy scores and try concluding which is better.

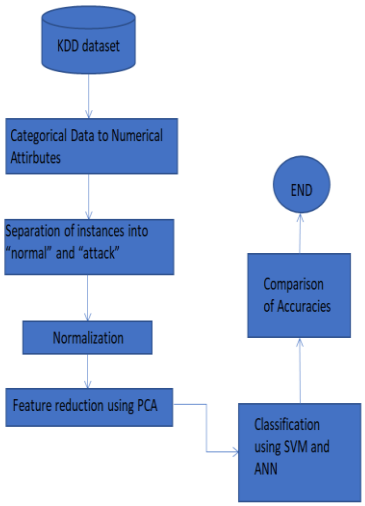

Fig 3: Flowchart of the proposed system

- KDD99 train and test data set is taken from KDD99 website for implementing Intrusion Detection System.
- All the categorical data is converted to numerical data using one hot encoding technique for better performance of SVM and ANN.
- The data is then classified into two features i.e. 'normal' and 'attack'. This in turn is converted into numerical data where "0" represents "normal" and "1" represents "attack" data.
- The whole data set in normalized using Standard Scaler which subtracts each attributes with mean and then divides it with standard deviation. The purpose of this is to give better results of our model.
- The next step is to reduce the features and dimensionality of the data set by feature reduction using PCA (Principal Component Analysis). The purpose of this is to give better results of our model.
- Use Support Vector Machine (SVM) and Neural Network (ANN) algorithms to classify the malicious data i.e. to classify 'normal' i.e. '0' and 'attack' i.e. '1' in the data set.
- Use these classifiers on the KDD99 test data set and calculate and compare the accuracies and AUC scores between SVM and ANN and conclude which model outperforms the other.

IV. METHODOLOGY

1. Datasets

The training and test data set contain around 42 columns where 41 of them are independent variables and the last column that is the 42nd column-"Class" is a dependent variable which tells us whether the data is attacked or normal.

	Duration	protocol_type	Service	Flag	src_bytes	dst_bytes	Land	wrong_fragment	Urgent	Hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_host
0	0	udp	private	SF	105	146	0	0	0	0	...	254	1.0	
1	0	udp	private	SF	105	146	0	0	0	0	...	254	1.0	
2	0	udp	private	SF	105	146	0	0	0	0	...	254	1.0	
3	0	udp	private	SF	105	146	0	0	0	0	...	254	1.0	
4	0	udp	private	SF	105	146	0	0	0	0	...	254	1.0	

st_rate	dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	dst_host_srv_rerror_rate	Class
0.0	0.0	0.0	0.0	0.0	normal
0.0	0.0	0.0	0.0	0.0	normal
0.0	0.0	0.0	0.0	0.0	normal
0.0	0.0	0.0	0.0	0.0	attack
0.0	0.0	0.0	0.0	0.0	attack

Figure 4: Data sets

2. Datasets After data preprocessing

The training and test datasets are converted as follows:

- First we apply one hot encoding to all categorical data and make them numeric.

- Then we convert the last column to numeric such that 'normal' will be represented by '0' and 'attack' will be represented by '1'. At this point we will have 118 columns in our datasets.
- We then perform standardization such that each feature is subtracted by its mean value and divided by standard deviation. The purpose of doing this is for better performance while applying a model for prediction.
- Finally we perform feature reduction using Principal Component Analysis (PCA) to reduce the dimensionality of the data set. From second step we get around 118 columns and these many columns will slow down the prediction made by models and hence PCA is used to scale it down to 29 columns in our data set.

	0	1	2	3	4	5	6	7	8	9	10
0	1.009547	2.721875	4.319461	-0.359954	0.624127	-0.243856	-0.945141	0.634793	-0.414786	0.561018	-0.682229
1	1.011112	2.700872	4.300598	-0.367219	0.626373	-0.238986	-0.988315	0.592740	-0.414861	0.547776	-0.686330
2	0.992844	2.675120	4.273205	-0.365069	0.626115	-0.235235	-1.026092	0.581218	-0.405217	0.550211	-0.696940
3	0.968220	2.648060	4.242250	-0.363249	0.622382	-0.232364	-1.059025	0.568749	-0.399322	0.551307	-0.706360
4	0.944913	2.620482	4.211396	-0.360868	0.620974	-0.228747	-1.094363	0.558314	-0.391142	0.553512	-0.716480

	11	12	13	14	15	16	17	18	19	20	21
	0.298186	-0.157484	-0.173936	-0.009943	-0.078714	0.029837	0.035032	0.016762	0.031956	-0.057640	0.011043
	0.302964	-0.173132	-0.191300	0.016423	-0.109007	0.019629	0.082441	0.048013	0.030239	-0.059209	0.009552
	0.302606	-0.175488	-0.187616	0.016669	-0.115658	0.021965	0.080942	0.041007	0.029036	-0.055469	0.008384
	0.302111	-0.179553	-0.187547	0.021557	-0.124006	0.022137	0.085613	0.039500	0.027880	-0.052532	0.007433
	0.301459	-0.182150	-0.184540	0.022971	-0.130544	0.024005	0.084955	0.033468	0.026698	-0.048943	0.006385

	22	23	24	25	26	27	28
	0.056779	0.022042	0.019839	-0.034443	-0.001649	-0.028527	0.000915
	0.053228	0.030041	0.016264	-0.027058	-0.009677	-0.019701	-0.002900
	0.051865	0.029643	0.015652	-0.025926	-0.009943	-0.021004	-0.003122
	0.050450	0.030254	0.014723	-0.024073	-0.011231	-0.020994	-0.004027
	0.049096	0.029981	0.014040	-0.022836	-0.011629	-0.022047	-0.004390

These are the first 5 rows of first 28 columns.

	0	1
	1	1
	2	1
	3	1
	4	1

This is the 29th column representing whether data attacked-'1' or normal-'0'.

Fig 5: Datasets after Data Preprocessing.

3. Support Vector Machine:

We make use of svc i.e. Support Vector Classifier model from sklearn package present in python library to implement support vector machine for our network dataset. Below is the small snippet of how svc is used.

- We get accuracy score around 90.833(out of 100).
- We get AUC score around 0.84(out of 1).

SVM

```
In [22]: lin = svm.SVC()
         lin.fit(train_data_pca_df_1, train_target_decoded[0])
         lin_predict = lin.predict(test_data_pca_df_1)
         print(lin.score(test_data_pca_df_1, test_target_decoded)*100)
         print("Number of support vectors for each class", lin.n_support_)

         90.82047011693444
         Number of support vectors for each class [1200 1047]
```

```
In [23]: from sklearn.metrics import roc_curve, auc
         fpr, tpr, thresholds = roc_curve(test_target_decoded[0], lin_predict)
         roc_auc = auc(fpr, tpr)
         print(roc_auc)

         0.8371146233486747
```

```
In [26]: label = 'Support Vector Classifier AUC:' + ' {0:.2f}'.format(roc_auc)
         plt.plot(fpr, tpr, c = 'g', label = label, linewidth = 4)
         plt.xlabel('False Positive Rate', fontsize = 12)
         plt.ylabel('True Positive Rate', fontsize = 12)
         plt.title('Receiver Operating Characteristic', fontsize = 12)
         plt.legend(loc = 'lower right', fontsize = 12)

Out[26]: <matplotlib.legend.Legend at 0x278003d0908>
```

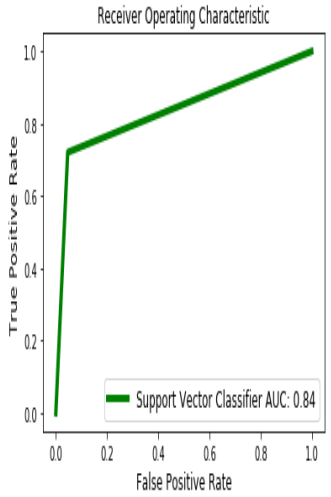

Fig 5: SVM process

4. Artificial Neural Network

We make use of keras model present in python library to implement Artificial Neural Network. The activation functions used are relu and softmax. Our ANN consists of one input layer with 29 input nodes, 2 hidden layers one with 16 nodes and the other with 12 nodes and finally 1 output layer containing two nodes. Below is the snippet of implementation of ANN.

- We get accuracy score around 93.159
- We get AUC score around 0.951

ANN

```
In [29]: import keras
         from keras.models import Sequential
         from keras.layers import Dense
         # Neural network
         model = Sequential()
         model.add(Dense(16, input_dim=29, activation='relu'))
         model.add(Dense(12, activation='relu'))
         model.add(Dense(2, activation='softmax'))

In [30]: model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

In [31]: history = model.fit(X_train, y_train, epochs=100, batch_size=64)

In [32]: y_pred = model.predict(X_test)
         #Converting predictions to label
         pred = list()
         for i in range(len(y_pred)):
             pred.append(np.argmax(y_pred[i]))
         #Converting one hot encoded test label to label
         test = list()
         for i in range(len(y_test)):
             test.append(np.argmax(y_test[i]))

In [33]: from sklearn.metrics import accuracy_score
         a = accuracy_score(pred,test)
         print('Accuracy is:', a*100)

         Accuracy is: 93.15948030569497

In [34]: from sklearn.metrics import roc_curve, auc
         fpr, tpr, thresholds = roc_curve(test, pred)
         roc_auc = auc(fpr, tpr)
         print(roc_auc)

         0.9512418002634057

In [35]: label = 'ANN AUC:' + ' {0:.2f}'.format(roc_auc)
         plt.plot(fpr, tpr, c = 'g', label = label, linewidth = 4)
         plt.xlabel('False Positive Rate', fontsize = 12)
         plt.ylabel('True Positive Rate', fontsize = 12)
         plt.title('Receiver Operating Characteristic', fontsize = 12)
         plt.legend(loc = 'lower right', fontsize = 12)

Out[35]: <matplotlib.legend.Legend at 0x27809a58288>
```

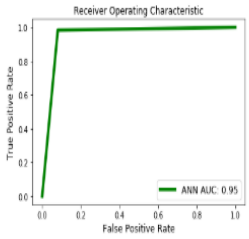

Fig 6: ANN process

V. RESULT AND ANALYSIS

Experimental results based on two metrics: Accuracy score and ROC_AUC curve along with AUC score.

1. Accuracy scores:

It is clear that the accuracy scores obtained from SVM is around 90.833 and the one obtained from ANN which is around 93.159. Hence, ANN outperforms SVM in case of accuracy scores.

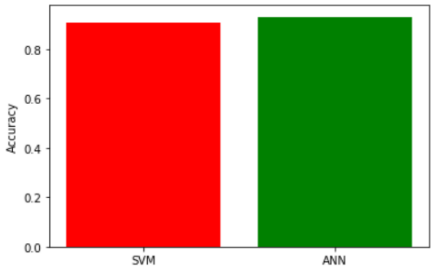

Fig 6: Accuracy scores of SVM and ANN

2. **ROC_AUC curve and AUC scores.**

It is seen that ANN outperforms SVM in AUC score as well. ANN has AUC score around 0.95 while SVM has AUC score around 0.84

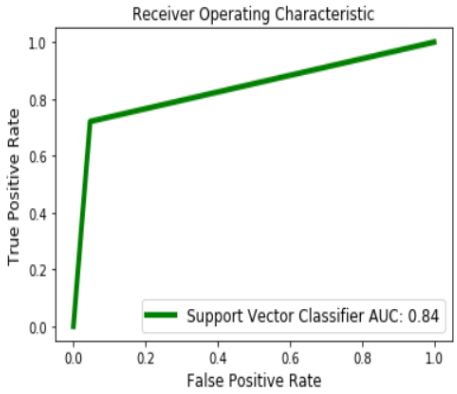

Fig 7: ROC_AUC Curve and AUC scores of SVM & ANN

VI. CONCLUSION

Malware detection is very much needed in current trends. With advancement in network communication, Intrusion Detection System plays a very vital role. There are many approaches and techniques such as Machine learning, Soft Computing and Artificial Intelligence that can be used to detect malicious content and activity in files and network. Among them Machine Learning techniques such as SVM and ANN were used to implement and to compare the accuracies and AUC scores.
It was seen that ANN outperformed SVM in both accuracy and AUC scores and hence ANN could detect and classify malicious data better than SVM. Future work deals with using an hybrid approach i.e. to blend other algorithms with SVM and ANN to give better classification accuracies

REFERENCES

[1]. Anish Halimaa A and Dr. K.Sundarakantham, "Machine learning based intrusion detection system ", Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) , IEEE 2019.
[2]. Raman Singh, Harish Kumar, R.K Singla. "Review of Soft Computing in Malware Detection", IP Multimedia Communications A Special Issue from IJCA - www.ijcaonline.org
[3]. Meijuan Gao, Jingwen Tian and Mingping Xia , "Intrusion Detection Method Based on Classify Support Vector Machine", 2009 Second International Conference on Intelligent Computation Technology and Automation, IEEE 2009.
[4]. Amin Dastanpour, Suhaimi Ibrahim, et al "Comparison of Genetic Algorithm Optimization on Artificial Neural Network and Support Vector Machine in Intrusion Detection System", 2014 IEEE Conference on Open Systems (ICOS), Subang, Malaysia.
[5]. Yusuf Sani, Ahmed Mohamedou, Khalid Ali, et al "An Overview of Neural Networks Use in Anomaly Intrusion Detection Systems", Proceedings of 2009 IEEE Student Conference on Research and Development (SCOReD 2009), UPM Serdang, Malaysia.
[6]. V. Jaiganesh, Dr. P. Sumathi, S. Mangayarkarasi, "An Analysis of Intrusion Detection System using Back Propagation Neural Network".
[7]. Chen Yan, "Intelligent Intrusion Detection based on Soft Computing", 2015 Seventh International Conference on Measuring Technology and Mechatronics Automation.
[8]. Manas Ranjan Patra, Ashalata Panigrahi, "Enhancing Performance of Intrusion Detection through Soft Computing Techniques", 2013 International Symposium on Computational and Business Intelligence.
[9]. Nouman Nazir, "Introduction to Artificial Neural Networks & Hidden Layer".
[10]. Adel Nadjaran Toosi, Mohsen Kahani, "A Novel Soft Computing Model Using Adaptive Neuro-Fuzzy Inference System for Intrusion Detection", Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control, London, UK, 15-17 April 2007.
[11]. Vikas Kumar, MS Swetha, MS Muneshwara, S Prakash, "Cloud computing: towards case study of data security mechanism," vol-2 issue-4 page no-1-8 2011
[12]. MS Muneshwara, MS Swetha, M Thungamani, GN Anil, "Digital genomics to build a smart franchise in real time applications," IEEE International Conference on Circuit, Power and Computing Technologies (ICCPCT),IEEE page no 1-4 2017 .
[13]. MS Muneshwara, A Lokesh, MS Swetha, M Thungamani, "Ultrasonic and image mapped path finder for the blind people in the real time system," IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI) IEEE, page no 964-969 2017
[14]. MS Swetha, M Thungamani A Novel Approach to Secure Mysterious Location Based Routing For Manet," International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-7 May, 2019
[15]. Sarraf G., Swetha M.S. (2020) "Intrusion Prediction and Detection with Deep Sequence Modeling" Security in Computing and Communications. SSCC 2019. Communications in Computer and Information Science, vol 1208. Springer, Singapore 978-981-15-4825-3