Intrusion Detection using Artificial Neural Networks with Best Set of Features

Kaliappan Jayakumar[1], Thiagarajan Revathi[2], and Sundararajan Karpagam[1]
[1]Department of Computer Science and Engineering, Kamaraj College of Engineering and Technology, India
[2]Department of Information Technology, Mepco Schlenk Engineering College, India

Abstract: *An Intrusion Detection System (IDS) monitors the behavior of a given environment and identifies the activities are malicious (intrusive) or legitimate (normal) based on features obtained from the network traffic data. In the proposed method, instead of considering all features for intrusion detection and wasting up the time in analyzing it, only the relevant feature for the particular attack is selected and intrusion detection is done with help of supervised learning Neural Network (NN). The feature selection is done with the help of information gain algorithm and genetic algorithm. The Multi Layer Perceptron (MLP) supervised NN is used to train the relevant features alone in our proposed system. This system improves the Detection Rate (DTR) for all types of attacks when compared to Intrusion detection system which uses all features and selected features using genetic algorithm with MLP NN as the classifier. Our proposed system results, in detecting intrusions with higher accuracy, especially for Remote to Local (R2L), User to Root (U2R) and Denial of Service (DoS) attacks.*

Keywords: *IDS, genetic algorithm, feature selection, NN, information gain.*

Received March 13, 2013; accepted June 9, 2014; published online August 9, 2015

1. Introduction

An Introduction to Intrusion Detection System (IDS), feature selection and Neural Network (NN) is discussed.

1.1. Intrusion Detection System

IDS detect malicious or suspicious activities in the network. An ID was first introduced by Anderson [1] and it was later improved by Denning [6].

IDS can be in the form of an application or device, just like firewall. Basically, there are two Intrusion Detection techniques: Anomaly detection and misuse detection [7]. Anomaly detection [7] is based on assumption that attacker behavior is different from normal user's behavior. The strategy is to look for odd or abnormal activities in a system or network one of the advantages of this detection is that it has high Detection Rate (DTR) and able to detect new attack.

Misuse Detection [10] (also known as signature-based detection), uses pattern matching. In order to determine an attack, it compares the data with the signature in signature database and if the data matches with the pattern as in signature data base, it detects the attack. This type of detection has high DTR with low false alarm. In this paper, the first technique i.e., the anomaly detection is used. The anomaly detection can be implemented using different techniques such as statistical model, computer immunological approach and machine learning.

1.2. Feature Selection

Feature selection is the most crucial step in constructing intrusion detection system [7]. A set of attributes or features that identified to be the most effective, are extracted in order to construct suitable IDS. In feature selection, a key problem is to choose the optimal set of features as not all features are relevant to the learning algorithm. Also, in some cases redundant features can lead to noisy data that distract the learning algorithm and degrade the accuracy of the IDS which slow the training and testing process. Feature selection is proven to have an important impact on the performance of the classifiers. Experiments show that feature selection can reduce the building and testing time of a classifier.

1.3. Artificial Neural Networks

The NN is data processing units which imitate the neurons of human brain. The Figure 1 shows classification of Artificial Neural Networks (ANN). Multi Layer Perceptron (MLP) is generally used NN architecture in several pattern identification problems. An MLP network consists of an input layer with a set of nodes such as input nodes, one or more hidden layers of processing nodes and an output layer of computation nodes. Each interconnection is associated with a scalar weight which is adjusted during the training phase. In addition, the back propagation learning algorithm is usually used to train a MLP, which is also called as back propagation neural networks. First of all, random weights are given at the beginning of training. Then, the algorithm performs weights tuning to define whatever hidden unit representation is most effective at minimizing the error of misclassification.

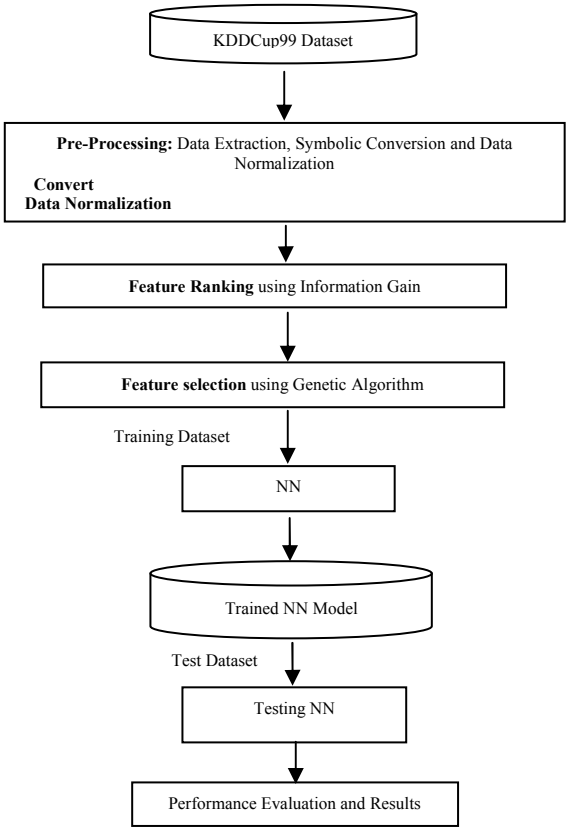

Figure 1. System flow diagram.

The paper is organized as follows: A Literature survey is given in the section 2. Details of KDD cup 99 dataset are given in section 3. Proposed systems are elaborated in section 4. Data preprocessing steps are given in section 4.1. Feature selection mechanisms are explained in section 4.2. The procedure of training and testing NNs are given in section 4.3. Performance evaluation, experiment details and the results are given in section 5. Section 6 concludes the paper with a sum up.

2. Related Work

Current IDSs use numerous techniques. Some of the techniques used for intrusion detection are statistic, hidden Markov, ANN, fuzzy logic [4], rule learning and outlier detection schema.

Chebrolu *et al.* [2] have identified significant input features in building IDS that is computationally efficient for detection systems. In the feature selection phase, Markov blanket model and decision tree analysis has been used. Bayesian Network (BN) Classifier and Regression Trees (CART) have been used to make an intrusion detection model.

Chena *et al.* [3] have used a Flexible Neural Tree (FNT) model for the IDS. The FNT model reduces the number of features. Using 41 features, the best accuracy for the Denial of Service (DoS) and User Gain Root (U2R) is given by the FNT model. The decision tree classifier supply the best accuracy for normal and probe classes, which are an improved less than the FNT classifiers.

Sung and Mukkamala [11] have removed one feature at a time to carry out an experiment on SVM and NN. KDD cup 1999 dataset has been used to verify this technique. In terms of the five-class classification, only 19 of the most significant feature are used, rather than all the 41features.

Cho [5] conduct a work where fuzzy logic and hidden Markov model have been deployed together to detect intrusions. In this work, the hidden Markov model is used for the feature reduction.

Li *et al.* [9] proposed a wrapper based feature selection algorithm to construct lightweight IDS. They applied modified RMHC for search strategy and modified linear SVM for valuation criterion. Their method speeds up the process of selecting features and generates high detection rates for an IDS.

3. KDD Cup 99 Dataset

The data set used in these experiments is ‘‘KDD Cup 99 data” [8], a well-known standard dataset for intrusion detection .The dataset includes a set of 41 features derived from each connection with label which specifies the status of connection records as either normal or specific attack type. Attacks are classified in to four main categories:

1. DoS.
2. Remote File Access (R2L).
3. U2R.
4. Probe.

The various attacks in each category are listed in Table 1.

Table 1. Different attacks falling into four major categories.

Attack Type	Attack Pattern
Probe	Ipsweep, nmap, portsweep, satan, mscan, saint
DoS	back, land, neptune, pod, smurf, teardrop, apache2, mailbomb, processtable, udpstorm
U2R	Buffer_overflow, loadmodule, perl, rootkit, ps, sqlattack, httptunnel, xterm
R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster, snmpgetattack, named, xlook, xsnoop, snmpguess, worm, sendmail

The sample distributions on the subset of 10% data of KDD Cup 99dataset is shown in Table 2.

Table 2. 10% data of KDD Cup 99 dataset.

Class	Number of Samples	Samples Percent (%)
Normal	97277	19.69
Probe	4107	0.83
DoS	391458	79.24
U2R	52	0.01
R2L	1126	0.23
	494021	100 %

Randomly selected data for training are listed in Table 3.

Table 3. The sample distributions on the train data with the corrected labels of KDD Cup 99 dataset.

Class	Number of samples	Samples percent (%)
Normal	13449	53.39
Probe	2289	9.09
DoS	9234	36.65
U2R	11	0.04
R2L	208	0.83
	25192	100

The sample distributions on the test data with the corrected labels of KDD Cup 99 dataset is shown in Table 4.

Table 4. The new attacks sample distributions on the test data with the corrected labels of KDD Cup 99 dataset.

Class	Number of Novel Attack Samples	Total Number of Attack Samples	Percentage of Novel Attack (%)
Probe	1315	2421	54.36
DoS	1715	7456	23
U2R	32	67	47.76
R2L	538	2734	19.68
	3600	12678	28.39

4. Proposed System

In the proposed system data is pre processed first with extraction of data from the large data set, symbolic conversions and normalization are followed by feature selection in this stage, the selected features are applied to NN for training. For the trained NN the test data is applied and the performance is evaluated. The proposed system flow chart is given in Figure 1. The steps in the proposed system are discussed in detail below.

4.1. Data Preprocessing Steps

KDD cup 99 dataset is very large database. So, it is very difficult to consider all the data for the experiment, as it takes time. The preprocessing is based on following steps.

1. Extract of data is made from random training and test data set from full data set.
2. The sample distribution of the randomly chosen training data set and test data set are tabulated in Tables 3 and 4.
3. Symbolic features are converted in to numerical values.
4. Data normalization is the process of scaling the input to fall within a specific range. The min max normalization is used. The formula is given in Equation 1.

$$X_n = 2 * \frac{X - X_{min}}{X_{max} - X_{min}} - 1 \quad (1)$$

Where X_{min} minimum value of the input, X_{max} maximum value of the input, X_n normalized output.

4.2. Feature Ranking

Feature ranking is done before training. The process of feature ranking identifies which features are more discriminative than the others. The feature ranking algorithm is given in Figure 2. This helps in improving system performance by eliminating irrelevant and redundant features. The information gains for the feature are calculated and features are ordered according to the information gain value in descending order.

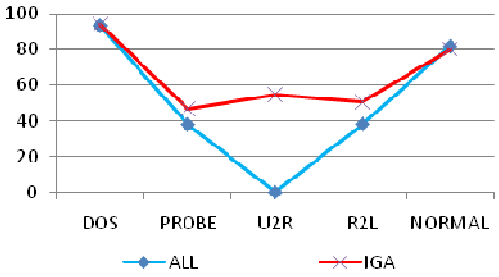

Figure 2. Detection rate.

4.2.1. Information Gain Ratio

Let S be a set of training set samples with their corresponding labels. Suppose there are m classes and the training set contains S_i samples of class I and s is the total number of samples in the training set, expected information gain ratio needed to classify a given sample. It is calculated by using the formula.

$$I(S_1, S_2, ..., S_m) = -\sum_{i-1}^{m} \left(\frac{S_i}{S}\right) log_2 \left(\frac{S_i}{S}\right) \quad (2)$$

A feature F with values $\{f_1, f_2, ..., fv\}$ can divide the training set into v subsets $\{S_1, S_2, ..., S_v\}$ where S_j is the subset which has the value f_j for feature F. Furthermore let S_j contain S_{ij} samples of class i. Entropy of the feature F is:

$$E(F) = \sum_{j-1}^{v} \frac{S_{1j} + ... + S_{mj}}{S} * I(S_{1j}, ..., S_{mj}) \quad (3)$$

Information gain for F can be calculated as:

$$IGR = Gain(F) = I(S_1, ..., S_m) - E(F) \quad (4)$$

4.2.2. Feature Selection

To reduce the dimensionality and to get better accuracy, the relevant features have to be selected using feature selection algorithm. Genetic algorithm can be used in this process. Genetic algorithm fitness function is designed in such a way that, the number of features selected has to be minimum and the sum of their information gain value should be maximum. The information gain for all the features is calculated and stored in Information Gain IG [ ] as per the Algorithm 1 Infogaincalculation. The genetic algorithm is designed to have a population size of 40.The binary chromosome is used. A chromosome of length 41 is constructed with each bit representing a feature. The fitness Algorithm 2 maxigminfcnt takes the chromosome as input. Scan the chromosome bits, if it is 1, take its respective information gain value and sum it up with the total information gain value. Feature count is calculated which is the total number of 1's is set in the chromosome. For example, consider the following chromosome.

11011100011110101100111001110110011010001

If the bit 5 is set (i.e., value=1) then, it indicates that the feature is selected. Its information gain value is

considered for finding the chromosome with minimum features.

- *Steps for Feature Selection*:

1. Generate the initial population.
2. Select individuals from the population to be parents.
3. With the help of crossover operator produce offspring.
4. Apply the mutation operator.
5. Calculate the information gain using Algorithm 1.
6. Apply the fitness function given in Algorithm 2.
7. Replace the population with the fittest of the whole population.

Algorithm 1: *Infogaincalculation.*

```
Feature set F [ ]
Information Gain IG[ ]
foreach (f in F)
        IG[i]=IGR(f);
end for
```

Algorithm 2: *Maxigminfcnt*.

```
Chromosome Cr [41]
for (i=0 to 41)
        if (Cr[ i ] == 1)
          then
           igsum = igsum+IG [ i ];
            fcnt = fcnt+1;
        endif
 endfor
```

The genetic algorithm parameter values are listed in Table 5.

Table 5. Genetic algorithm parameters.

Modeling Description	Setting
Population Size	40
Selection Technique	Roulette wheel
Crossover Type	Uniform crossover
Crossover Rate	0.5
Mutation Rate	0.1

- *Encoding and Initial Population*

In the GA-based feature selection, an entrant feature set can be represented by a binary string called a chromosome. Chromosomes comprising population are encoded in the form of binary vector in a manner to create of genes as the number of feature in each feature space. The i^{th} bit in the chromosome represents the occurrence of the i^{th} feature. Initialization of the population is generally prepared by seeding the population with random values. If the value of the gene, which is coded in binary system, is ''1'', it indicates that the corresponding feature is selected, in the converse, if the value of gene is ''0'', it indicates that the related feature is not selected.

In the proposed GA-based feature selection, each chromosome is randomly initialized, with each chromosome in the population coded to a binary. The size of chromosome is equal to the total number of features.

- *Fitness Function*

Fitness function is used to fix individuals that are fit to the most favorable solution. Every individual has its own fitness value. A higher value of fitness indicates that the individual is more suitable for problem solution; on the other hand, a lower value of fitness means that the individual is less appropriate for a solution. The encoded chromosomes are searched to optimize a fitness function after the initialization of population. In this method, the fitness value of each chromosome is evaluated according to the sum of the information gain of the features selected and the number of features selected. The information gain sum should be maximum and the number of features selected should be minimum for the selection.

- *Selection*

The aim of the selection process is to choose the individuals of the next generation according to the selected fitness function and selection method among the existing population. In the selection process, the transfer possibility of the fittest individual's chromosome to the next generation is higher than others. The decision of the individual's characteristic which will be transferred to the next generation is based on the values evaluated from the fitness function and shows the quality of the individual. The Roulette wheel selection method which is the most general and most easily applied one is chosen in this work.

- *Crossover*

In the pre-crossover phase, individuals are identified by using a mating process. Forming the new generation is called 'crossover'. The most commonly used method is forming two new individuals from the two chromosomes. In our work, uniform crossover is used in the crossover procedure.

- *Mutation*

To increase the range of the chromosomes which are applied on crossover, mutation process can be applied. Mutation introduces local variations to the individuals for searching diverse solution spaces and keeps the variety of the population. In our method, the number of chromosomes that will be mutated is found according to the mutation rate and their values are altered from '1' to '0' or '0' to '1' respectively.

- *Stopping Criteria*

The stopping criteria are set for this algorithm when maximum number of generations, reaches 25 the algorithm stops.

4.3. Training and Testing the NN

A supervised algorithm such as MLP Back Propagation is used for training. A three layer MLP

NN was simulated. The number of neurons in the input layer is the number of features considered for the experiment. Number of neurons in the output layer (sigmoid function) is 2. The number of hidden neurons can be determined using the following rules of thumb:

1. The number of hidden neurons should be two third of the size of the input layer plus the size of the output layer.
2. The amount of hidden neurons should be between the size of the square root of (input neuron*output neuron).
3. The amount of hidden neurons should be less than twice the size of the input layer.

5. Performance Evaluation and Results

The performance of the proposed IDS is evaluated with help of confusion matrix. The classification performance of IDS is measured by the False Alarm Rate (FAR), DTR and accuracy. They can be calculated using the confusion matrix in Table 6 and defined as follows:

Confusion matrix is a 2×2 matrix, where the rows represent actual classes; while the columns have the values correspond to the predicted classes.

Table 6. Confusion matrix.

	Predicted Attack	Predicted Normal
Actual Attack	True Positive (TP)	False Negative (FN)
Actual Normal	False Positive (FP)	True Negative (TN)

- *TP*: The number of attack detected when it is actually attack.
- *TN*: The number of normal detected when it is actually normal.
- *FP*: The number of attack detected when it is actually normal, namely false alarm.
- *FN*: The number of normal detected when it is actually attack.

$$FAR = \frac{FP}{TN + FP} * 100 \quad (5)$$

$$DTR = \frac{TP}{TP + FN} * 100 \quad (6)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100 \quad (7)$$

In this section, the performance of the proposed IDS is studied with the help of two experiments. In experiment I all the features are given as input to the Nn and trained. The test data is given and its accuracy, DTR and FAR are calculated. In experiment II only the relevant features are selected, using the genetic algorithm with information gain as the base. The selected features and training dataset are given as input to the experiment 2 and the performance measures accuracy, DTR, FAR are calculated. The results are tabulated and plotted as graphs.

5.1. Experiment 1

All experiments were performed on a Windows platform having configuration Intel core 2 Duo CPU 2.49 GHZ, 2GBRAM.Simulations and the analysis of experimental data are performed with the use of MATLAB NN Toolbox. The existing supervised and unsupervised algorithms are considered for the experiment.

In experiment 1 all features are considered for training the NN s and test data with 28.39 % of novel data is considered. DTR, FAR and Accuracy for the different types of attack using all the features of the test dataset of KDD Cup 99 dataset are tabulated in Table 7

Table 7. DTR, FPR,, accuracy for different classes on the test data set with corrected labels of KDD Cup 99 data set.

	Normal	Probe	DoS	U2R	R2L
DTRR	81.81	38.02	93.39	17	38.24
FAR	6.99	0.21	2.48	0.16	7.66
Accuracy	87.61	92.12	96.47	99.84	92.26

5.2. Experiment 2

Steps in experiment 2:

- *Step* 1: Data preprocessing is done as per the steps given in the section 4.1.
- *Step* 2: The information gain is calculated using the formula 4.3. Open source weka software [12] is used to calculate information gain for each feature in the training dataset.
- *Step* 3: The genetic algorithm is used to select the features for further processing. The strategy for selection of features is, the sum of information gain of the features must be maximum but, the number of features selected should be minimum.
- *Step* 4: The dataset of selected features is trained using the MLP NN.
- *Step* 5: For the trained NN the test data with novel data of 28.39% is applied and the performance is evaluated.

From the Table 8 we can infer that the DTR for U2R, R2L, Probe and DoS attack are highly improved compared to experiment1. Comparing the accuracy results in experiment 1 with experiment 2 there is a very good improvement in all the attacks except in Normal. The FAR has been decreased marginally compared to experiment 2.

Table 8. DTR, FPR, accuracy for different classes on the test data set with corrected labels of KDD Cup 99 data set

	Normal	Probe	DoS	U2R	R2L
DTRR	80.39	46.79	94.1	55	51.06
FAR	7.16	0.06	2.44	0.11	7.25
Accuracy	86.74	94.43	96.69	99.85	92.31

The Figure 2 shows the graph with DTR values plotted against attacks .It gives the comparison of the DTR obtained in experiment 1 and experiment 2.

The Figure 3 shows the graph with FAR values plotted against attacks .It gives the comparison of the FAR s obtained in experiment 1 and experiment 2.

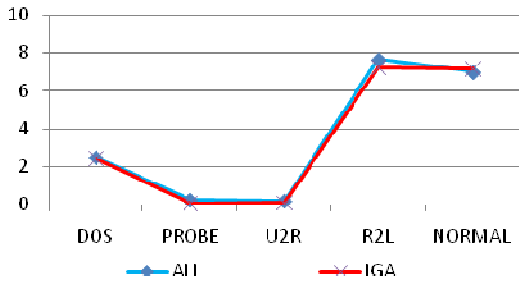

Figure 3. FAR.

The Figure 4 shows the graph with Accuracy values plotted against attacks. It gives the comparison of the accuracy obtained in experiment 1 and experiment 2.

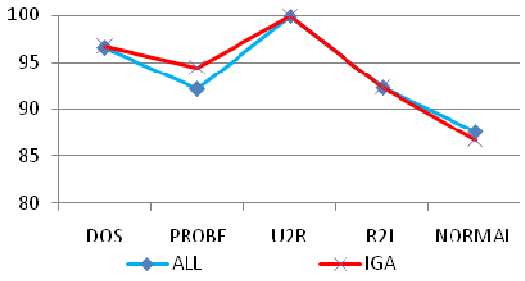

Figure 4. Accuracy.

6. Conclusions

The test result proves that the performance measures of IDS get improved when the selected features alone are used instead of all features. In this approach the feature selection is done by genetic algorithm. Genetic algorithm fitness function aims at maximizing the information gain sum and minimizing the features count. Our proposed system provided much improved performance results in detecting intrusions with higher accuracy, especially for R2L, U2R and DoS attacks.

References

[1] Anderson P., "Computer Security Threat Monitoring and Surveillance," available at: http://csrc.nist.gov/publications/history/ande80.pdf, last visited 1980.

[2] Chebrolu S., Abraham A., and Thomas P., "Feature Deduction and Ensemble Design of Intrusion Detection Systems," *Computers and Security*, vol. 24, no. 4, pp. 295-307, 2005.

[3] Chena Y., Abrahama A., and Yanga B., "Feature Selection and Classification using Flexible Neural Tree," *Journal of Neuro Computing*, vol. 70, no. 1, pp. 305-313, 2006.

[4] Chimphlee W., Abdhullah A., Md M., Chimphlee S., and Srinoy S., "A Rough-Fuzzy Hybrid Algorithm for Computer Intrusion Detection," *the International Arab Journal of Information Technology*, vol. 4, no. 3, pp. 274-254, 2007.

[5] Cho S., "Incorporating Soft Computing Techniques into a Probabilistic Intrusion Detection System," *IEEE Transactions on Systems, MAN and Cybernetics*, vol. 32, no. 2, pp. 154-160, 2002.

[6] Denning E., "An Intrusion-Detection Model," *IEEE Transaction on Software Engineering*, vol. 13, no. 2, pp. 222-232, 1987.

[7] Kayacik G., Zincir-Heywood N., and Heywood I., "Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets," *in Proceedings of the 3rd Annual Conference on Privacy, Security and Trust*, Andrews, Canada, 2005.

[8] KDDCup99 Dataset., available at: http:// kdd.ics.uci.edu//databases/kddcup99/kddcup99.html, last visited 2013.

[9] Li Y., Wang J., Tiand Z., Luc T., and Young C., "Building Lightweight Intrusion Detection System using Wrapper-based Feature Selection Mechanisms," *Computers and Security*, vol. 28, no. 6, pp. 466-475, 2009.

[10] Michalski S., Carbonell G., and Mitchell M., *Machine Learning: An Artificial Intelligence Approach*, Tioga Publishing Company, 1983.

[11] Sung A. and Mukkamala S., "Identifying Important Features for Intrusion Detection using Support Vector Machines and Neural Networks," *in Proceedings of Inter-National Symposium on Applications and the Internet*, pp. 209-217, 2003.

[12] Weka., available at: http:// www.cs.waikato.ac.nz/ml/weka/, last visited 2013.

Kaliappan Jayakumar received the BE degree in computer science and engineering from M.K. University, India in 2002 and the ME degree in computer science and engineering from Anna University, India in 2005. Currently, he is pursuing his PhD degree in the field of intrusion detection at Anna University, India. He is currently working as an Assistant Professor at the Department of Computer Science and Engineering, Kamaraj college of Engineering and Technology, India. He has presented and published 6 papers in Conferences. His current research interests include: Intrusion detection, data mining and soft computing techniques. He is a Life Member of Indian Society for Technical Education.

Thiagarajan Revathi received the BE degree in electrical and electronics engineering from M.K. University, India in 1986 and the ME degree in computer science and engineering from Bharathiyar University, India in 1995 and PhD degree in computer networks from the M.S. University, India 2008. She is currently working as a Professor and Department Head with the Department of Information Technology, MEPCO Schlenk Engineering College, India. She has conducted Workshops and Conferences in the areas of computer networks and multimedia. She has presented and published more than 33 papers in conferences and journals. Her Research interests include: Computer networks, multimedia, data mining, algorithms and network security. She is a Life member of Computer Society of India and Indian Society for Technical Education.

Sundararajan Karpagam received the BE degree in computer science and engineering from Anna University, Chennai, India in 2005 and the ME degree in computer science and engineering from Anna University, India in 2010. Currently, she is working as an Assistant Professor at the Department of Computer Science and Engineering, Kamaraj college of Engineering and Technology, India. She has presented and published 3 papers in Conferences. Her current research interests include: Data hiding, data mining, neural networks and genetic algorithm. She is a Life Member of Indian Society for Technical Education.