International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

A Survey of Artificial Neural Networks machine learning methods and Applications in Bio-Neuron System

Thangaraj[1] E, Subinson G[2], S.Rimlon Shibi[3]

[1, 2, 3]St.Joseph University in Tanzania, Department of Computer Science and Engineering, Dar es Salaam 11007, Tanzania

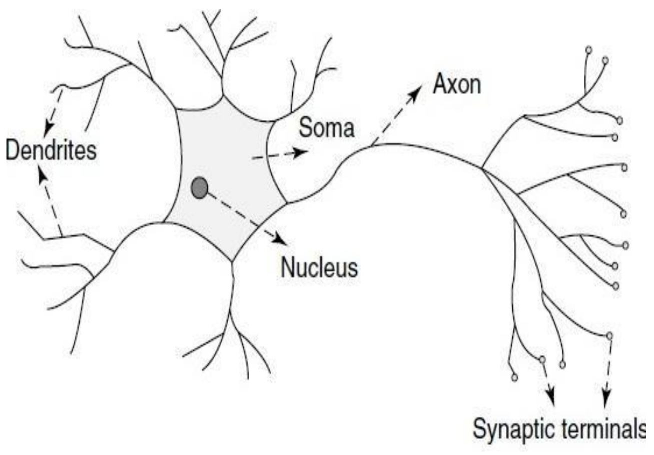
Abstract: ***In this paper we show that "what are neural networks?" and they are more important in today's Artificial intelligence, because more advances have been made in developing intelligent system, some inspired by biological neural networks. Artificial neural networks give a very exciting alternatives and other application which can play important role in today's computer science field and also provide for those readers with little or no knowledge of artificial neural networks to help them understand the other articles in this issue of Computer. Describe the basic biological neuron and the artificial computer model, network architectures and learning processes, and present some of the most commonly used artificial neural networks models. We conclude with artificial neural network applications.***

Keywords: Artificial neural networks, biological neuron, Hamming distance, Artificial Intelligence, Hebbian learning, data mining.

1. Introduction

The general idea of artificial neural networks is basically bring from the subject of biology where neural networks perform on important role in human body. In human body the operative is done with help of neural network. Neural is just a web of mutually joined neurons which are millions and millions in number with contribute to mutually joined neurons all the parallel processing is done in human body.

A neuron is a special biological cell that process information from one neuron to another neuron with the help of some electrical signal and chemical change. It is combined of a cell body or soma and two types of out reaching tree like branches: the axon and the dendrites. The cell body has a nucleus that contains data about hereditary traits and plasma that holds the molecular equipment's or producing material needed by the neurons. The whole process of receiving and passing signals is done in particular manner like a neuron receives signals from other neuron through dendrites. The Neuron pass the signals at spikes of electrical activity through a long thin stand known as an axon and an axon split this signals through synapse and send it to the other neurons.



Artificial Neural Networks

An artificial neuron is basically an engineering concept of biological neuron. It have device with more inputs and one output. Artificial neural network is contain of large number of simple processing elements that one mutually joined with each other and layered also.

Usage of Artificial Neural Networks

Artificial neural networks, with their remarkable ability to derive meaning from complicated or undetermined data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be produce of as an expert in the category of information it has been given to analyses. This expert can then be used to display projections given new situations of interest and answer what if questions.

2. Difference between Modern Computers and Biological Neural Systems:

2.1 Modern Computers

Contain one or few Processors which are high speed but complex. It has Localized Memory separate from processor. Computing is done with stored programs in a sequential and centralized manner. In terms of reliability it is much harmed. The Operating Environment is well defined and well constrained.

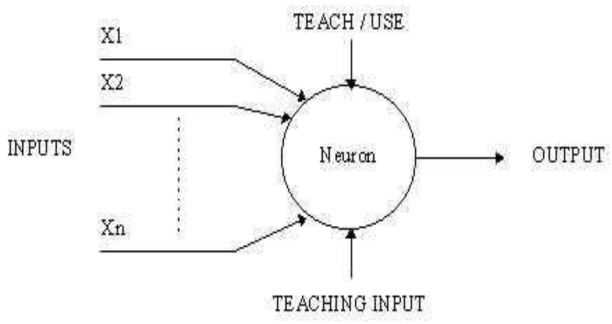
2.2 Biological Neural Systems

Contains a large number of processor which have low speed but simple in structure. Having Distributed Memory but integrated into processor. Computing is done with self-learning in a parallel and distributed manner. In terms of reliability it is robust. The operating environment is poorly defined and unconstrained.

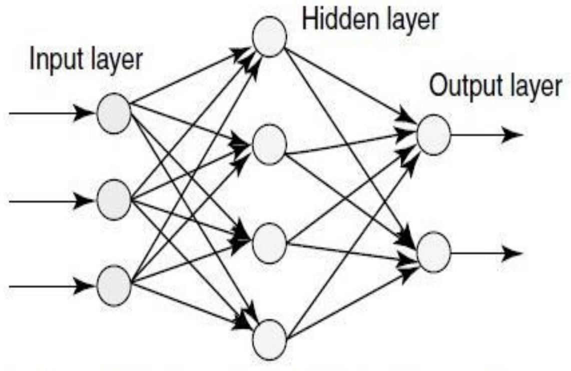
3. Artificial Neural Network Structure

A Simple Neuron

An artificial neuron is a device with more inputs and one output. The neuron has two kind of modes operation; the training mode and the applying mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the current mode, when a taught input pattern is detected at the input, it is associated output becomes the current output. If the input does not belong in the taught list of inputs, the firing rule is used to determine whether to fire or not.



A simple neuron

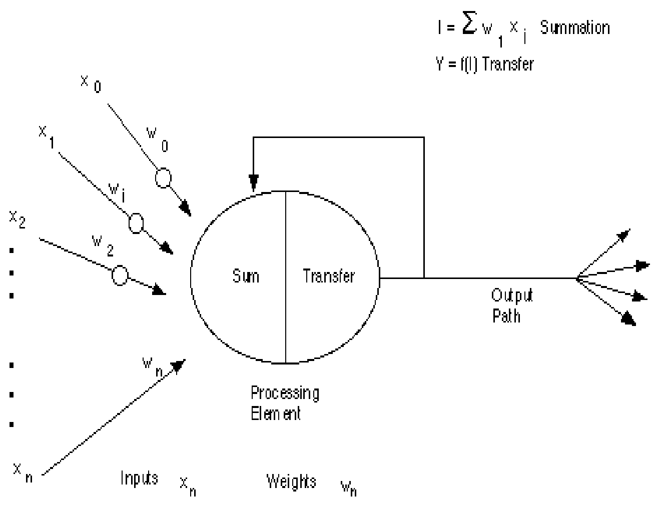


(b) Multilayered artificial neural network

3.1 Firing Rules

The firing rule is an important concept in neural networks and accounts for their high the ability to be easily modified. A firing rule determines how one calculates whether a neuron should fire for any input pattern. It is relates to all the input patterns, not only the ones on which the node was trained.

A simple firing rule can be implemented by applying Hamming distance method. The rule goes as follows: Take a collection of training patterns for a node, some of which cause it to fire (the 1-taught set of patterns) and others which prevent it from doing so (the 0-taught set). Then the patterns not in the collection cause the node to fire if, on comparison, they have more input data in common with the 'nearest' pattern in the 1-taught set than with the 'nearest' method in the 0-taught set. If there is a tie, then the method remains in the undefined state.



For example, a 3-input neuron is taught to output 1 when the input (X1, X2 and X3) is 111 or 101 and to output 0 when the input is 000 or 001. Then, before applying the firing rule, the truth table is

X1:	0	0	0	0	1	1	1	1
X2:	0	0	1	1	0	0	1	1
X3:	0	1	0	1	0	1	0	1
OUT:	0	0	0/1	0/1	0/1	1	0/1	1

As an example of the way the firing rule is applied, take the pattern 010. It differs from 000 in 1 element, from 001 in 2 elements, from 101 in 3 elements and from 111 in 2 elements. Therefore, the 'nearest' pattern is 000 which belongs in the 0-taught set. Thus the firing rule requires that the neuron should not fire when the input is 001. On the other hand, 011 is equally distant from two taught patterns that have different outputs and thus the output stays undefined (0/1).

By applying the firing in every column the following truth table is obtained;

X1:	0	0	0	0	1	1	1	1
X2:	0	0	1	1	0	0	1	1
X3:	0	1	0	1	0	1	0	1
OUT:	0	0	0	0/1	0/1	1	1	1

The difference between the two truth tables is called the generalization of the neuron. Therefore the firing rule gives the neuron a sense of similarity and enables it to respond 'sensibly' to patterns not seen during training.

An artificial neural network operates by creating connection between may be different processing elements, each analogous to a single neuron may be physically constructed or simulated by a digital computer. Each neuron takes several input signals, then, based on internal weighting system, generate a single output signal that's typically sent as input to another neuron. The neurons are tightly mutually joined and organized into different layers. The input layer get the input, the output layers produces the final output. Usually one or more hidden layers are sandwiches in between them. This structure makes it impossible to predict or know the exact flow of data

A single artificial neuron is the basic element of the neural network. It consist of several inputs

(x1, x2,..., xm) and one output y that can be written as follows:

(1) y= f(xi, wi) where, wi are the function parameter weights of the function f.

Equation is called an activation function. It maps any real input into a usually limited

Range, often [0, 1] or [-1, 1]. This function may be linear or non-linear such as one of the following:
a) Hyperbolic tangent: f(x) = tanh(x) = 1 – 2/1+exp (2x)
b) Logistic f(x) = 1/1+exp (-x)
c) Threshold f(x) = 0 if x < 0, 1 otherwise,
d) Gaussian f(x) = exp(x2/2)

If equation transforms inputs into output linearly, then a single neuron is described by the following:

(2) y = WTx where,
y = [yj] - the vector consists of n outputs,
WT = [wij] T - transposed matrix [nxm] of weights,
x = [xi] - the vector comprises m inputs.
To solve a problem using neural networks, sample inputs and desired outputs must be given.
The network then learns by adjusting its weights.

4. Artificial Neural Network Learning

Artificial neural networks typically start out with randomized weight for all their neurons. That means that they don't know anything and must be trained to solve the particular problem for which they are intended. Broadly speaking there are several methods for training artificial neural networks, depending on the problem it must be solve.

The ability to learn is a fundamental train of intelligence. Although an accuracy definition of learning is difficult to formulate, a learning process in the Artificial neural network context can be viewed as the problem of updating network architecture and connection weights so that a network can efficiently perform a particular task. The network usually must learn the connection weights from available training patterns. Performance is increased over time by iteratively updating the weights in the network. Artificial neural network capability to automatically learn from examples makes them attractive and exciting. Instead of following a set of rules describe by human experts, artificial neural networks appear to learn underlying rules (like input-output relationships) from the given collection of representative examples. This is one of the major features of neural networks over traditional expert systems. To understand or make a learning process, you must first have a model of the environment in which a neural network operates, that is, you should know what information is available to the network.

4.1 A Self –Organizing Artificial Neural Networks

A self - organizing artificial network is exposed to large amounts of data and tends to discover patterns and relationships in that data. It is unsupervised learning. Self – organizing artificial neural networks are modeled on biological neural networks, when groups of neurons appear to self-organize into specific regions with common functionality. Different regions of the self-organizing artificial neural network are trained to be detectors for distinct features from the input set. Beginning network weights are either set randomly, or are based off the Eigen vectors of the input space. The Euclidean distance from each input sample to the weight vector of each neuron is computed, and the neuron whose weight vector is most similar to the input is declared the best match unit. The update formula is given as

Wj [n+1] =wj[n] +Φ [j,n]α[n](x[n]-wj[n])

Here, w is the weight vector at time n, α is a monotonically decreasing function that time that ensures that learning rate will decrease over time, x is the input vector, and Φ [j,n] is a measure of the distance between the best match unit and neuron j at iteration n. As can be seen from this algorithm, the amount by which the neuron weight vectors change is based on the distance from the best match unit, and the amount of time. Decreasing the potential for change over time helps to reduce volatility during training, and helps to ensure that the network convers.

4.2 A Back –propagation artificial neural networks:

Conversely, is trained by humans to perform particular tasks. During the training time, the teacher evaluates whether the artificial neural network's output is correct. If it's correct, the neural weightings that produced that output are reinforced; if the output is incorrect, those type is most often used cognitive research and for problem solving applications.

The simple perceptron is just able to handle linearly separable or linearly independent problems. By taking the partial derivative of the error of the network with respect to each weight, we will learn a little about the direction the error of the network is moving. In fact, if we take the negative of this derivative (i.e. the rate change of the error as the value of the weight increases) and then proceed to add it to the weight, the error will reduce until it reaches local minima. This makes sense because if the derivative is positive, this tells us that the error is increasing when the weight is increasing. The understood thing to do then is to add a negative value to the weight and vice vers an if the derivative is negative. Because the taking of these partial derivatives and then applying them to each of the weights takes place, beginning from the output layer to hidden layer weights, then the hidden layer to input layer weights (as it turns out, this is necessary since changing these set of weights requires that we know the partial derivatives calculated in the layer downstream), this algorithm has been called the back propagation algorithm. A neural network can be trained in two different modes: online and batch modes. The number of weight updates of the two methods for the same number of data presentations is very different. The online method weight updates are computed for each input data sample, and the weights are modified after each sample. An alternative solution is to compute the weight update for each input sample, but store these values during one a through the training set which is called an epoch. At the end of the epoch, all the contributions are added, and only then the weights will be updated with the composite value. This method adapts the weights with a cumulative weight update, so it will follow the gradient more closely.

It is called the batch-training mode. Training basically involves feeding training samples as input vectors through a neural network, calculating the error of the output layer, and then adjusting the weights of the network to minimize the error. The average of all the squared errors (E) for the outputs is computed to make the derivative easier. Once the error is computed, the weights can be updated one by one. In the batched mode variant, the descent is based on the gradient

$$\Delta wij\ (n) = -n^* \partial E/\partial Wij + \alpha^*\Delta wij\ (n-1)$$

Where η and α are the learning rate and momentum respectively. The momentum term determines the effect of past weight changes on the current direction of movement in the weight space. A good choice of both η and α are required for the training success and the speed of the neural network learning. It has been proven that back propagation learning with sufficient hidden layers can approximate any nonlinear function to arbitrary accuracy. This makes back propagation learning neural network a good candidate for signal prediction and system modeling.

4.3 Hebbian learning

The learning paradigms discussed above result in an adjustment of the weights of the connections between units, according to some alternation rule. Perhaps the most influential work in connectionism’s history is the contribution of Hebb, where he presented a theory of behavior based, as much as possible, on the physiology of the nervous system. The important concept to emerge from Hebb’s work was his formal statement of how learning could occur. Learning was based on the alternation of synaptic connections between neurons. particularly, when an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased. The principles underlying this statement have become known as Hebbian Learning. Virtually, most of the neural network learning techniques can be considered as a variant of the Hebbian learning rule. The basic idea is that if two neurons are active at same time, their interconnection sould be strengthened. If we consider a single layer net, one of the mutually connected neurons will be an input unit and one an output unit. If the data are represented in bipolar form, it is express the desired weight update as wi(new)=wi(old)+xi0, where o is the desired output for i = 1 to n(inputs). Unfortunately, plain Hebbian learning continually strengthens its weights without bound (unless the input data is properly normalized).

5. Artificial Neural Network Applications

Given this description of neural networks and how they work, what real world apply are they suited for? Neural networks have been broad applicability to real world business problems. In fact, they have already been successfully applied in serveral industries.

Since neural networks are best at identifying patterns or trends in data, they are well suited for prediction or forecasting needs including
Sales forecasting
Industrial process control
Customer research
Data validation
Target marketing
But to give you some more important examples:

5.1 Artificial are also used in the following specific paradigms

Recognition of speakers in communications, Recovery of telecommunications from faulty software, Hand-written word recognition and Facial recognition.

5.2 Our neural network based data mining approach contain of three major phases:

1. Network construction and training.
This phase constructs and trains a three layer neural network based on the number of attributes and number of classes and chosen input coding method.

2. Network pruning
The pruning phase aims at removing redundant links and units without increasing the classification error rate of the network. A small number of units and links left in the network after pruning enable us to extract concise and comprehensible rules.

3. Rule extraction
This phase extracts the classification rules from the pruned network. The rules are in the form of "if (a, Bv,) and (x, Bv,) and ... and (x, Bv,) then Cy where a, s are the attributes of an input tuple, v, a~re constants, & are relational operators (=, <, 2, <>), and Ci is one of the class labels. Due to space limitation, in this paper we omit the discussion of the first two phases. Details of these phases can be found in our earlier work 191, [lo]. We shall elaborate in this paper the third phase. Section 2 describes our algorithms to extract classification rules from a neural network and uses an example to illustrate how the rules are generated using the proposed approach.

5.3 Extracting rule from a trained neural network:

Network pruning results in a relatively simple network. However, even with a simple network, it is still difficult to find the clear relationship between the input tuples and the output tuples. A number of reasons contribute to the difficulty of extracting rules from a pruned network. First, even with a pruned network, the links may be still too several to express the relationship between an input tuple and its class label in the form of if . . . then ... rules. If a network still has n input links with binary values, there could be as many as 2, similar type input patterns. The rules could be quite lengthy or complex even for a small n. Second, the activation values of a hidden unit could be anywhere in the range 1-1, 11 depending on the input tuple. It is difficult to derive an explicit relationship between the continuous activation values of the hidden units and the output values of a unit in the output layer.

A Rule Extraction Algorithm

The rule extraction algorithm, RX, consists of the four steps given below:

1) Apply a clustering algorithm to find out clusters of hidden node activation values.
2) Mention the discretized activation values and compute the network outputs Generate rules that describe the network outputs in terms of the discretized hidden unit activation values.
3) For each hidden unit, enumerate the input values that lead to them and generate a set of rules to describe the hidden units' discretized values in terms of the inputs.
4) Combine the two sets of rules obtained in the previous two steps to obtain rules that relate the inputs and outputs.

The first step of RX clusters the activation values of hidden units into a manageable number of discrete values without sacrificing the classification accuracy of the network. After Clustering, we obtain a set of activation values at each hidden node.

The second step is to relate these discretized activation values with the output layer activahon values, i e, the class labels. And the third step is to relate them with the attribute values at the nodes connected to the hidden node a general purpose algorithm X2R was developed and implemented to automate the rule generation process. It takes as input a set of discrete patterns with the class labels and produces the rules describing the relationship between the patterns and their class labels the details of this rule generation algorithm can be found in our earlier work to cluster the activation values, we used a simple clustering algorithm.

5.4 Artificial Neural Networks in Marketing

There is a marketing application which has been integrated with neural networks system. The airline marketing tactician is a computer system made various intelligent technologies including expert systems. A feed forward neural network is desegregated with the AMT and was trained using back – propagation to assist the marketing control of airline seat allocations. The adaptive neural approach was governable to rule expression. Furthermore, the application's environment changed rapidly and constantly, which required an uninterrupted adaptive solution. The system is a monitor and recommend booking advice for each departure. Such information has a direct effect on the profitability of an airline and can provide a technological advantage for users of the system.

While it is significant that neural networks have been applied to this problem, it is also important to see that this intelligent technique can be integrated with expert systems and other approaches to make a functional system. Neural networks were apply to discover the influence of undefined interactions by the different variables. While these interactions were not defined, they were used by the neural system to develop useful conclusions. It is also noteworthy to see that neural networks can influence the bottom line.

5.5 Neural Networks in Medicine

Artificial Neural Networks (ANN) are currently a 'hot' research area in medicine and it is believed that they will receive extensive application to biomedical systems in the next few years. At the moment, the research is several on modeling parts of the human body and recognizing diseases from various scans.Neural networks are ideal in recognizing diseases using scans since there is no need to mention a specific algorithm on how to identify the disease. Neural networks learn by example to explain of how to recognize the disease are not needed. What is needed is a set of examples that are representative of all the variations of the disease. The quantity of examples is not as important as the 'quantity'. The examples need to be selected very carefully if the system is to perform reliably and efficiently.

5.5 Modeling and Diagnosing the Cardiovascular System

Neural Networks are used experimentally to model the human cardiovascular system. Diagnosis can be achieved by building a model of the cardiovascular system of an individual and comparing it with the real time physiological measurements taken from the patient. If this routine is carried out regularly, potential harmful medical conditions can be detected at an early stage and thus make the process of combating the disease much easier.

A model of an individual's cardiovascular system must mimic the relationship among physiological variables (i.e., heart rate, systolic and diastolic blood pressures, and breathing rate) at different physical activity levels. If a model is adapted to a separately, then it will becomes a model of the physical condition of that individual. The simulator will have to be able to adapt to the features of any individual without the supervision of an expert. This calls for a neural network.

Another reason that justifies the use of artificial neural network technology, is the ability of artificial neural networks to provide sensor fusion which is the combining of values from several different sensors. Sensor fusion activated the Artificial neural networks to learn complex relationships among the individual sensor values, which would otherwise be lost if the values were separately analyzed. In medical modeling and diagnosis, this implies that even though every sensor in a set may be sensitive only to a specific physiological variable, artificial neural networks are capable of detecting complex medical conditions by fusing the data from the individual biomedical sensors.

5.6 Instant Physician

An application developed in the mid-1980s called the "instant physician" trained an auto associative memory neural network to store a large number of medical records, each of which includes information on symptoms, diagnosis, and treatment for a particular case. After training, the net can be presented with input consisting of a set of symptoms; it will then find the full stored pattern that represents the "best" diagnosis and treatment.

6. Conclusion

By studying artificial Neural Network we had concluded that as per as technology is developing day by day the need of Artificial Intelligence is increasing because of only parallel processing. Parallel Processing is several needed in this present time because with the help of parallel processing only we can save more and more time and money in any work related to computers and robots. If we talk about the Future enhancement work we can only say that we have to develop much more algorithms and other problem solving techniques so that we can remove the limitations of the Artificial Neural Network. And if the Artificial Neural Network concept combined with the Computational Automata and Fuzzy Logic we will definitely solve some limitations of this excellent technology.

7. Acknowledgement

I would like to thank my mother and brother for their valuable encouragement and special thanks to Mr. Anto joseph and the staff members from the St. Joseph University for their Technical Support to us.

References

[1] Vidushi Shama, Sachin Rai & Anurag Dev "A Comprehensive Study of Artificial Neural Networks" Volume 2, Issue 10, October 2012.
[2] Yochanan Shachmurove, Dorota Witkowska "Utilizing Artificial Neural Network Model to Predict Stock Markets" September 2000.
[3] Ajith Abraham "Artificial Neural Networks" Oklahoma State University, Stillwater, OK, USA.
[4] Hongjun Lu, Member, IEEE Computer Society,Rudy Setiono, and Huan Liu, Member, IEEE Effective "Data Mining Using Neural Networks" DECEMBER 1996.
[5] Steve Lawrence,C. Lee Giles, Ah Chung Tsoi, Andrew D.Back Face Recognition "A Convolutional Neural Network Approach"
[6] Ani1 K. Jain Michigan State University Jianchang Mao K.M. Mohiuddin ZBMAZmaden Research Center "Artificial neural networks: A Tutorial" march 1996.
[7] Nal Kalchbrenner, Edward Grefenstette, Phil Blunsom "A Convolutional Neural Network for Modelling Sentences" Department of Computer Science University of Oxford.
[8] Derrick H. Nguyen, Bernard Widrow "Neural Network for self-learning Control systems"
[9] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade "Neural Network-Based Face Detection"
[10] Christos Stergiou and Dimitrios Siganos "Neural networks"
[11] Ilya Sutskever James Martens Geoffrey Hinton University of Toronto "Generating Text with Recurrent Neural Networks"
[12] Donald F. Specht "A General Regression Neural Network" IEEE transactions on neural networks. vol. 2. No. 6. November 1991.

Author Profile

Mr. Thangaraj E. received the M. Tech in Computer Science and Engineering at Dr. M.G.R Educational and Research Institute University in Chennai and B. E degree in Computer Science and Engineering at Madurai Kamaraj University in Madurai. Presently he is working as a lecturer in ST. Joseph College of Engineering and Technology in Tanzania for the past 6 years. His area of interest is networking and cloud computing.

Mr. Subinson G. received the M. E in Computer Science and Engineering at V.M.K.V Engineering College in Salem and M.SC in Information Technology at Sami Arul Arts and Science College at Tanjore. Presently he is working as a lecturer in ST. Joseph College of Engineering and Technology in Tanzania for the past 6 years. His area of interest is networking and cloud computing.

Mr. S.Rimlon Shibi. Received the M.E in Computer and Communication at National Engineering College in Kovilpatti and B.Tech in Information Technology at Francis Xavier Engineering College in Tirunveli. Presently he is working as a lecturer in ST. Joseph College of Engineering and Technology in Tanzania for the past 3 years. His area of interest is Wireless networking and cloud computing.