NETWORK LEARNING AND TRAINING OF A CASCADED LINK-BASED FEED FORWARD NEURAL NETWORK (CLBFFNN) IN AN INTELLIGENT TRIMODAL BIOMETRIC SYSTEM

Benson-Emenike Mercy E.[1] and Ifeanyi-Reuben Nkechi J.[2]

[1]Department of Computer Science, Abia State Polytechnic Nigeria.
[2]Department of Computer Science, Rhema University Nigeria.

ABSTRACT

Presently, considering the technological advancement of our modern world, we are in dire need for a system that can learn new concepts and give decisions on its own. Hence the Artificial Neural Network is all that is required in the contemporary situation. In this paper, CLBFFNN is presented as a special and intelligent form of artificial neural networks that has the capability to adapt to training and learning of new ideas and be able to give decisions in a trimodal biometric system involving fingerprints, face and iris biometric data. It gives an overview of neural networks.

KEYWORDS

CLBFFNN, Learning, Training, Artificial Neural Network, Trimodal, Biometric System

1. INTRODUCTION

With ever growing field of artificial intelligence, there are numerous intelligent systems providing algorithms to solve any particular problem which require human intelligence [1][2]. These artificial intelligent systems led to development of a more creative, knowledgeable and exceptional system that helps computers learn [3][4]. CLBFFNN is an intelligent Artificial Neural Network having interconnecting network of artificial neurons which uses some algorithms and the mathematical formulas for the processing of the data or information on the computational purposes. It uses modern technique of “Learning with examples” which the corresponding response seems to come from a kind of natural intelligence rather than using Artificial Intelligence.

2. OVERVIEW OF NEURAL NETWORKS

Neural Networks (NNs) are an abstraction of natural processes, which can be likened to the operations and functions of biological neurons [5]. NN has been incorporated into a broad and integrated computational model of adaptive systems, since early 1990s, in other to utilise their learning power and adaptive capabilities, they have the ability to learn and adjust to new incoming patterns, create adaptation to a dynamic environment much more effective and efficient [2]. Combined with evolutionary algorithms (EAs) they can be regarded as a general framework for adaptive systems i.e. systems that excel in changing their architectures and learning rules adaptively without human intervention.

2.1 Types of Neural Networks (NN)

Neural Networks are broadly divided into Feed-forward and Recurrent or feedback networks.

DOI : 10.5121/ijaia.2018.9603

Multilayer perceptron, single-layer perceptron and Radial Basis Function nets are grouped under Feed-forward networks while Hopfield network, Competitive networks, Kohonen's SOM and ART models are grouped under Recurrent or feedback networks[6].

2.1.1 Multilayer Perceptron [MLP]

MLP is one of the most largely used neural networks [7]. MLP, given two sets of data (i.e. input/output pairs) is capable of developing a precise nonlinear mapping, using a learning algorithm by adjusting the network weights. It has been proved that a two-layer MLP can effectively approximate any nonlinear mapping [8].MLP training uses majorly Back-propagation (BP) algorithm [7], where a steepest descent gradient approach and a chain-rule are adopted for back-propagated error correction from the output layer [1]. Significant efforts have been put into improving the speed of convergence, generalization performance, and the discriminative ability of MLP. To accelerate the BP algorithm, several heuristic rules have been proposed to adjust the learning rates or modify error functions [9]. Acceleration of training MLP can also be achieved by the use of other modifications to the standard BP algorithm, such as conjugate gradient BP, recursive least-square-based BP, and the Levenberg-Marquardt algorithm. To verify the generalization ability of MLP, the independent validation method can be used by dividing the available data set into a number of sub sets for training, validation and testing [10]. To improve the discriminative capability of MLP when applied to a classification task, a discriminative MLP learning rule was proposed which is more suitable for pattern classification tasks [11].

2.1.2Hopfield Neural Networks

Associative memory networks include linear associative memory and Hopfield associative memory. Linear associative memory is an effective single-layer network for the retrieval and reduction of information [12]. Given a key input pattern U = [u_1, u_2, ...,u_k] and the corresponding output V = [v_1, v_2, ..., v_k], associative memory learns the memory matrix P to map the key input u_1 to the memorized output $\hat{V}_1$. There are a number of ways to estimate the memory matrix. One estimate of the memory matrix W is the sum of the outer product matrices from pairs of key input and memorized patterns

$$P = \sum_{n=1}^{N} Vn \quad U_n^y \tag{1}$$

To further reduce the memorized error, an error correction approach has been introduced to minimize the error function

$$E(P) = \frac{1}{2} \| \; Vn \; - P \, Un \; \|^2 \tag{2}$$

Hopfield associative memory is a nonlinear content-addressable memory for storing information in a dynamically stable environment [13]. The Hopfield network is a single-layer recurrent network which contains feedback paths from the output nodes back into their inputs. Given an input x(0), the Hopfield network iteratively updates the output vector by

$$U(n + 1) = f\,(Pu \quad (n) - \theta), \tag{3}$$

until the output vector become constant, where f(·) is the activation function. Associative memory is able to deduce and retrieve the memorized information from possibly incomplete or corrupted data

.

2.1.3Modular Networks

There are strong biological and engineering evidences to support the fact that the information processing capability of NNs is determined by their architectures [4]. Much work has been devotedto finding the optimal or near optimal NN architecture using various algorithms, including EAs [14]. However, many real world problems are too large and too complex for any single ANN to solve in practice. There are ample examples from both natural and artificial systems that show that an integrated system consisting of several subsystems can reduce the total complexity of the entire system while solving a difficult problem satisfactorily. NN ensembles adopt the divide-and-conquer strategy. Instead of using a single large network to solve a complex problem, an NN ensemble combines a set of ANNs that learn to decompose the problem into sub-problems and then solve them efficiently. An ANN ensemble offers several advantages over a monolithic ANN [15]. First, it can perform more complex tasks than any of its components (i.e., individual ANNs in the ensemble). Second, it can make the overall system easier to understand and modify. Finally, it is more robust than a monolithic ANN, and can show graceful performance degradation in situations where only a subset of ANNs in the ensemble performs correctly. There have been many studies in statistics and ANNs that show that ensembles, if designed appropriately, generalize better than any single individuals in the ensemble. A theoretical account of why and when ensembles perform better than single individuals was presented in [16].

2.2Overview of Artificial Neural Networks

Structurally, ANN is a computational model that adopts human nervous system technique with back propagation algorithm for training-testing phase [17][18]. It is an adaptive system capable of transforming its arrangement according to input and output sequence that flows in the network [19]. Neural network provides rational success [20], removes the shortcomings of traditional methods and can be trained from examples. Its characteristics include robustness, less data requirement, self-organizing, fast computation, adaptive learning, parallelism, ease of software and hardware implementation, high rate of error tolerance and broad view [1][2]. However, ANN is prone to over-fitting, has an empirical nature of model development and a greater computational problem. The following are the various types of artificial neural networks: feed-forward neural networks, radial basis function (RBF) networks, Kohonen self-organizing networks, recurrent networks, stochastic neural networks, modular neural networks, dynamic neural networks, cascading neural networks, and neuro-fuzzy networks [21].

Artificial neural network is made up of a collection of neurons that are connected to each other mainly for processing of information [22][3]. Figure 1 shows an instance of a neuron. Neurons have many weighted inputs such as (x_1, x_2, x_3… x_n) with their corresponding weighting, (w_{1j}, w_{2j}, w_{3j}… w_{nj}), according to its importance. Every input is a scalar quantity and stands for the data. For instance, in a face recognition system having 12x12 pixel image, a 144 input representation ranging from X_1 to X_{144} and weightings of w_1 to w_{144}, equivalent to 144 pixels in the input image, will be achieved.

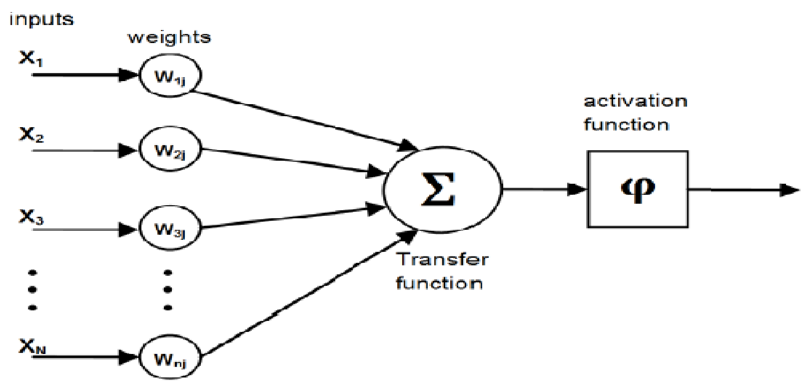

Figure 1: A Neuron[18]

Artificial Neural Network structure comprises of input layers, hidden nodes, output nodes and an activation function, as seen in Figure 2.

i) **Number of input layers**: The input layers offer broad view ability to the network.
ii) **Number of hidden nodes:** The best method for selecting the most appropriate number of hidden neurons is not in existence. One could use Kolmogorov theorem for computing number of hidden neurons. This is given as 2i+1 neurons, where i is the number of inputs.
iii) **Number of Output Nodes:** Single output neural network is preferable to networks that produce numerous outputs because they produce more superior results.
iv) **Activation or Transfer Function:** This is a mathematical formula that determines the output of a processing node. Every input node computes and adds activation to its net input. The essence of the transfer function is to ensure that the output does not assume a large value because this can obstruct neural network training and adversely affect its performance.

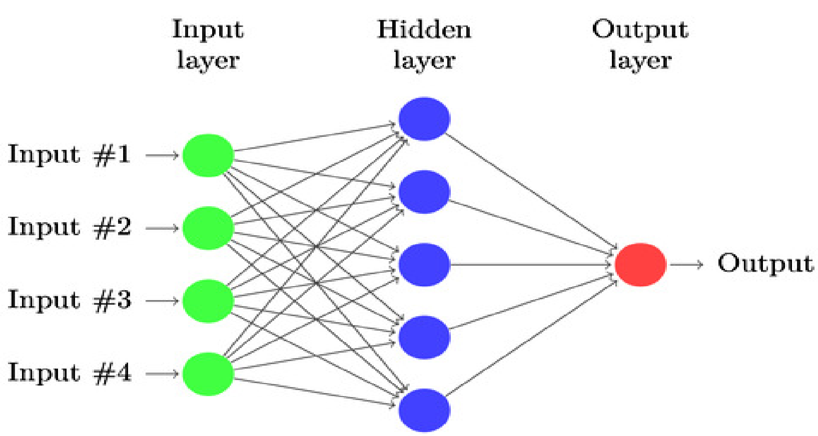

Figure 2: The Structure of an ANN[1]

The model of a neural network model is of three types:

i. **Feed-Forward Network:** Here, there is a sequential connection of neurons from one layer to another. There is a continuous connection of neurons and cannot connect backwardly [6].

ii. **Recurrent Network:** In a recurrent network, there must be at least one backward connecting loop or feedback loop. It is mainly used for memory relationship and computation of optimization [1].

iii. **Self-Organization Networks:** This network is based on unsupervised learning. In this network target output is not known to the network and is mainly used for cluster analysis. [23],[24].

2.3 Network Training Types

The two permissible actions that can take place in a neural network are training and no-training. A non-training scenario occurs when the network is only run with a specific amount of weights. In a training scenario, we first run the network with a given set of weights; modify some or all the weights to obtain an entirely new set of weights, then run the network again with the new set of weights. This procedure continues until some pre-set target is achieved. The network assumes the initial weight assigned to it and subsequently adjusts it based on the computed strategy. After this training, the network will be expected to exhibit 'intelligence' by performing what it has been trained to do, based on the input patterns. The obtained information is stored in secure memory or database, as in the human brain.

Every neural network must be trained by means of parameter adjustment. Neural networks can be trained using various methods such as supervised, unsupervised and reinforcement training [17].

a) Supervised Training: This kind of training involves presenting to the neural network a tester set of inputs and their corresponding desired outputs. After training, when any given set of input is given to the network, the neural network will in turn map the said input to its equivalent trained output.

Weights are determined to allow the network to produce answers as close as possible to the known correct answers. The back-propagation algorithm belongs into this category.

b) Unsupervised Training: This method is used when the output or outcome for a given set of inputs is unknown. This type of training does not require a correct answer associated with each input pattern in the training set. The neural network, in this case, is fed with enough data and is allowed to try, learn, organize, modify network's weights and biases and derive conclusions itself without outside intrusion[3]. Explores the underlying structure in the data, or correlations between patterns in the data, and organizes patterns into categories from these correlations. The Kohonen algorithm belongs to this category.

c) Reinforcement Training: It is a learning technique that stores insufficient information concerning the target output. It is similar to supervised learning; but gives less accurate result [19].

3. An Overview Of CLBFFNN

CLBFFNN is also a Feed-Forward Neural Network. Three processing layers are encompassed in a feed-forward neural network. They include the input layer which is the first layer, the output layer which is the last layer and the hidden layer which is the middle layer. Each processing element makes self-determining computations on data that it receives, calculates the weighted sum of its inputs and conveys the results to another layer [25]. These processing elements will jointly produce the output from the network. These elements are similar in operations to human brain neurons, hence, referred to as artificial neurons (see Figure 3).

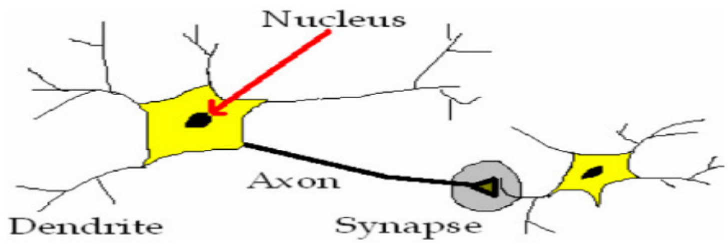

Figure 3: A Brain Neuron[2]

In the output layer, authentication of a neuron output is done using a threshold function. Synapses are connections between neurons. The artificial neurons and the synapses are respectively denoted by nodes and edges of a directed graph, as seen in figure 4. Each synapse has an assigned accompanying weight and the choice of these weights affects the operation and the classification of the neural network.

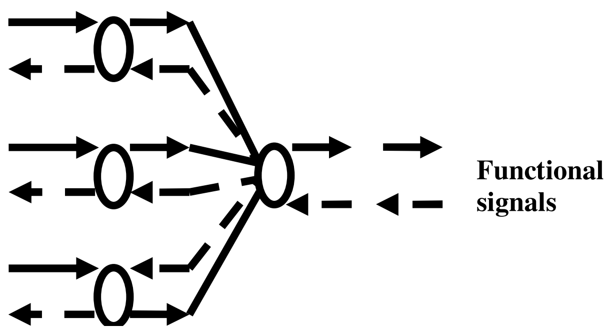

Figure 4: Functional and Error Signals in CLBFFNN

CLBFFNN is an intelligent multi-biometric system classifier. When a multi-biometric system is trained to have an artificial intelligence to be able to take decisions based on several inputs, then such system is said to be 'intelligent'. This is achieved by the application of an Artificial Neural Network (ANN), Fuzzy logic, Neuro-fuzzy method etc. System accuracy cannot be fully achieved when only one matching algorithm is applied [26].

After pre-processing, feature extraction and quality assessment, the next stage is verification or authentication which is done by a CLBFFNN classifier. CLBFFNN is an ordered cascade of two neural networks CLBFFNN(1) and CLBFFNN(2), as seen in figure 6. Their designed arrangement guarantees reduction in computation cost, increases system's detection accuracy and efficiency. The network components are interconnected with layers and flow of information via these network components is not interrupted in any way. Apart from learning and training of CLBFFNN with back propagation algorithm, it makes decisions on final association between its inputs. It is trained with a back propagation algorithm.

CLBFFNN(1) is a neural network that uses Back Propagation algorithm. It is composed of an input layer, a single hidden layer and an output layer with a sigmoid activation function. The feature vectors from pre-processed multi-biometric data form the input into CLBFFNN(1). The CLBFFNN(1) has three cascaded stages as seen in figure 5.

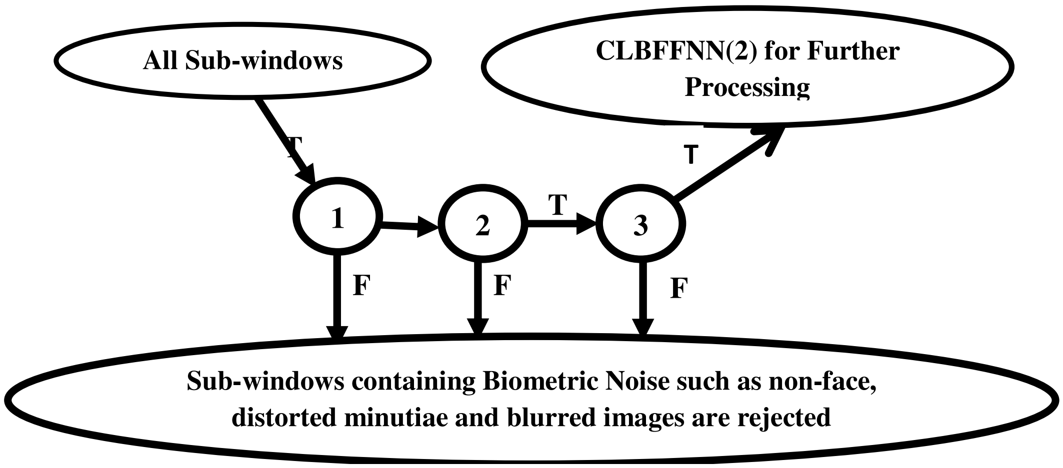

Figure 5: The Three Cascaded Stages of CLBFFNN(1)
[Source: [27]]

The first stage ensures that the feature patterns that are entering into the network are life-scanned features, i.e. facilitating further liveness detection. The second stage filters biometric noises such as the non-face from face feature, distorted or incomplete minutiae from fingerprints and blurred iris pattern from iris images. Lastly, the third stage determines the image width and height and calculates the time complexity of the biometric features. These sub windows are passed through each section of image and minimizes the false positive rate (i.e. non-faces for example) and detects the face. This method reduces false rejection and false acceptance errors to their barest minimum and increases system verification accuracy. The simulated result from CLBFFNN(1) is fed into CLBFFNN(2) for the function approximation.

Network learning and training, template matching, comparison and decision making take place in CLBFFNN(2). Output from CLBFFNN(1) is approximated since it is connected with CLBFFNN(2). CLBFFNN(2) is a universal approximator and has a very compact topology. CLBFFNN(2) is also composed of an input layer, a single hidden layer and an output layer with fast locally regulated neurons; and also has a feed-forward design.The output from CLBFFNN(2) determines the recognition result.

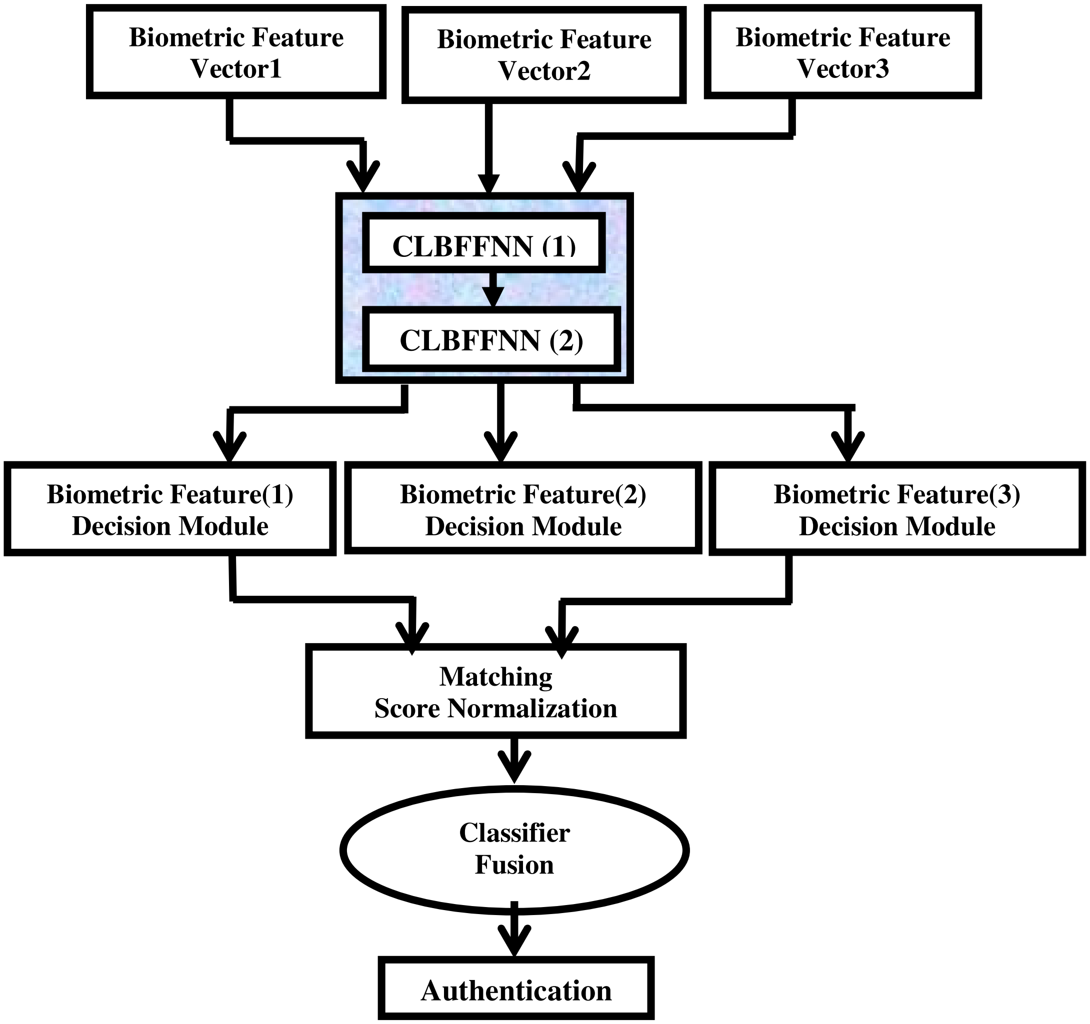

Figure 6: CLBFFNN in a Trimodal Biometric

3.1 CLBFFNN Learning Using Back Propagation Algorithm

Back propagation algorithm is a supervised or monitored learning process used for the design of a Feed Forward Neural Network (FFNN). Learning in CLBFFNN is accomplished by back propagation algorithm and it has to do with the modification of weights and other networks parameters to ensure proper training of the biometric features. Learning is done here by giving instances. When instances of what is to be achieved; is giving to the algorithm, it obtains the necessary output for the particular input through series of network's weight adjustments and training. Back propagation algorithm in

CLBFFNNhas two passes for each input-output vector pair:

i)**forward pass:** Here, the input propagates through the net and an output is produced.
ii)**backward pass:** Here, the difference (error) between the desired and obtained output is evaluated with an error function. This error function is minimised towards the weights of the net, starting backwards from the output layer to the input layer. Back propagation algorithm is easy to implement and is capable of solving all manners of problems.

Network learning process begins by submitting input patterns, whose output has already been determined and known by the network. The weighted sum of the neurons is calculated and transferred by each layer to the next layer following it through an activation function. The acquired outcome is used as an input to the subsequent layer. Finally output layer generates an output pattern which is then compared to the target pattern. The input and its corresponding target are called a Training Pair. A mean squared error is computed based on the dissimilarity between the result obtained and the expected output. The learning rate of the network and its efficiency is measured by the obtained dissimilarity or error [28]. The back propagation algorithm aims at minimizing this error to its lowest point. The following is a back propagation algorithm designed for this purpose:

BP Algorithm

1. Give initial values to weights, learning rate, momentum and error precision.
2. Enter the input vector and the desired output associated with it.
3. Compute the weighted sum and the output of the network.
4. Compute the mean square error. This is obtained by evaluating the contrast between the result obtained and the expected output.
5. Adjust the weights by back propagating them from the output layer [29].

3.2CLBFFNN Training and Learning

Network training and learning go hand in hand. At the commencement of training and learning, the acquired feature matrices for the biometric images are used as input data set to the CLBFFNN. This is done through a supervised training algorithm. After the training, the trained pairs are supplied to the neural network. The training of CLBFFNN is done using both positive and negative set of the biometric data because after the network training, it is expected to recognize negative set of data such as non-face, distorted and blurred image features. Therefore, the input to CLBFFNN could be aggregates of face and non-face training data; correct and distorted or incomplete minutiae for fingerprint; or correct and distorted or blurred image features for iris.

The network will be trained to provide an output 'positive' for good and complete image features of biometric features while with negative, distorted or blurred image features, the output will be 'negative'. We fix a target output for each feature based on the average feature value of each of the images. Then a program is developed to train the neural network for biometric feature recognition.

When the training begins, all the weights of the neurons are aggregated and stored in a weight file. After the first set of data is trained, an error is calculated, which is based on the difference between the computed output and the target output, as seen in figure7. This aims at reducing this error and ensuring that the final output obtained is equal to the target output; by series of network training and weight adjustments.

3.2.1 The Back Propagation Algorithm in CLBFFNN Training

Back Propagation algorithm is adopted for CLBFFNN training. It is a feed-forward supervised training network with the capability of automatically updating network weights using defined update procedures. This is done by taking an incomplete derivative of the error function to examine how each of the network weight contributes to the error. The learning and training is done from one area of the neuron to another.

There are different stages in training CLBFFNN with back propagation algorithm, as seen in Figure 7. The first stage is the initialization of all the network weights which involves assigning initial values to the network weights in the range of $-1 \geq x \leq 1$. The second stage involves the feeding of the input pattern into the network through the input neurons. The input neurons send the patterns to each of the hidden neurons; which will compute the activation function and transfer the result to the output neurons. The output neurons will calculate the activation, the output and the neuron error. The error is computed by the difference between the set target and the neuron output, as shown in equation (4).

$$\textbf{Neuron Error} = \textbf{Set Target} - \textbf{Neuron Output} \quad \textbf{(4)}$$

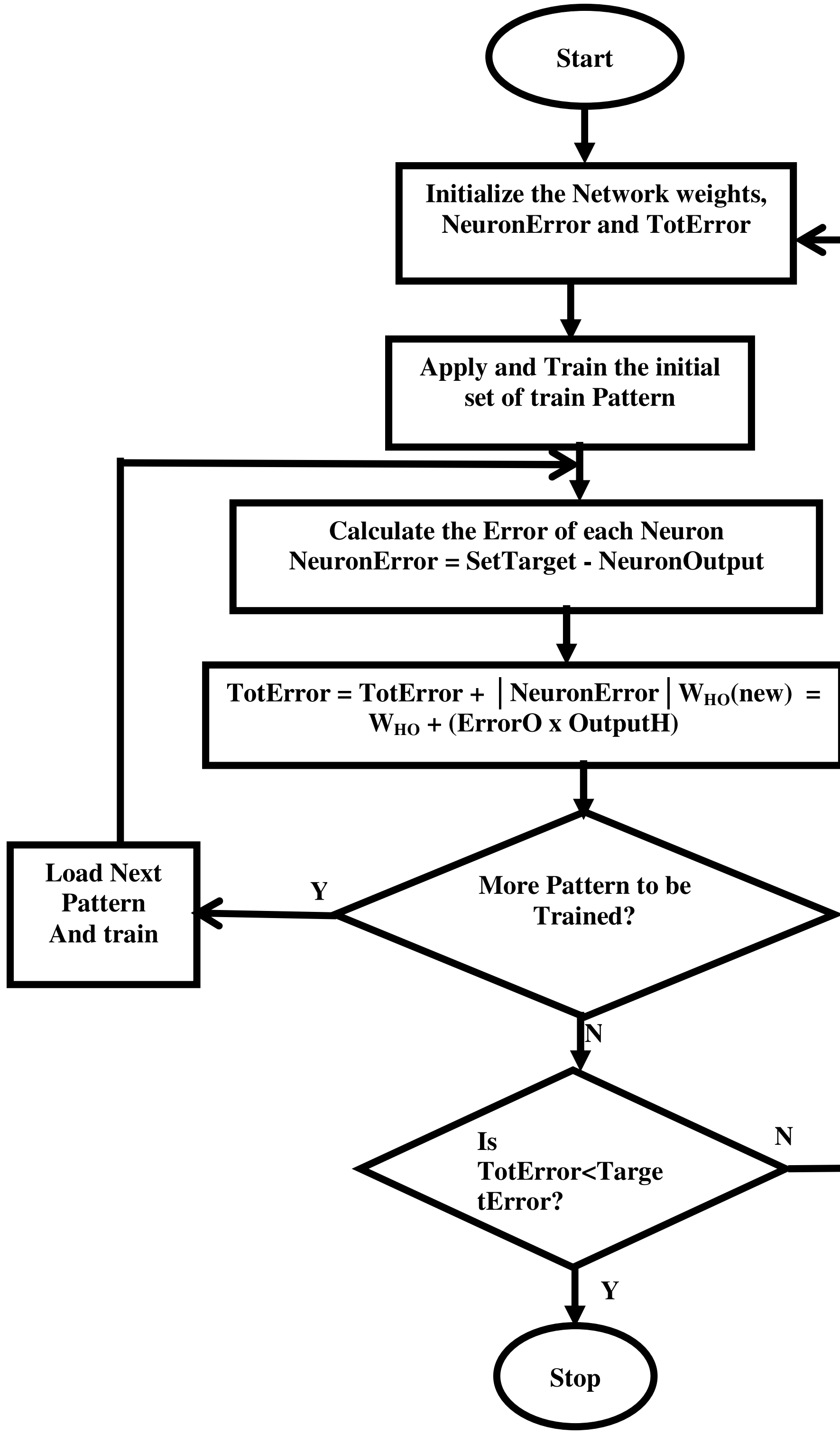

Figure 7: CLBFFNN Training

In the third stage, the NeuronError gotten is propagated back to the hidden layer neurons (O) to obtain the output neuron error, as given by equation (5).

ErrorO= OutputO (1-OutputO)(TargetO – OutputO) **(5)**

The next stage has to do with weight adjustment, in which case, the neuron weights assume new values, as given by equation (6).

W_{HO}(new) = W_{HO} + (ErrorO x OutputH) **(6)**

Where
H = The hidden layer neuron,
O = The output neuron,
W_{HO} = The weight of the synapse between the hidden layer and the output

layer, and which stands for the initial neuron weight,
W_{HO}(new) = The new or trained neuron weight.

After the weight adjustment, the hidden neuron errors are calculated. Finally, the total error is compared with the target error; if it is less than the target error the training stops, otherwise the whole process repeats. The network will be well trained so as to recognize input patterns and give out output accordingly.

3.2.2 Face Learning and Training in CLBFFNN

Figure 8 is the flowchart for learning and training procedure for face feature in CLBFFNN.
Where
V_i = input layer neuron vector
V_h = hidden layer neuron vector
P = output layer neuron
w_{ih} = the weight matrix between the input and the hidden layer
b_{h0} = the bias of the hidden layer neurons
w_{ho} = weight matrix connecting the hidden and the output layers
b_0 = the bias of the output layer neurons
E_o= the error vector for output neurons
E_h = the error vector of each hidden layer neuron
V_d = the desired output vector
LR = learning rate
MF = momentum factor

The sigmoid activation function is defined by

$$\mathbf{F}(\boldsymbol{V_i}) = \frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{exp}\,(-\boldsymbol{V_i})} \qquad \mathbf{(7)}$$

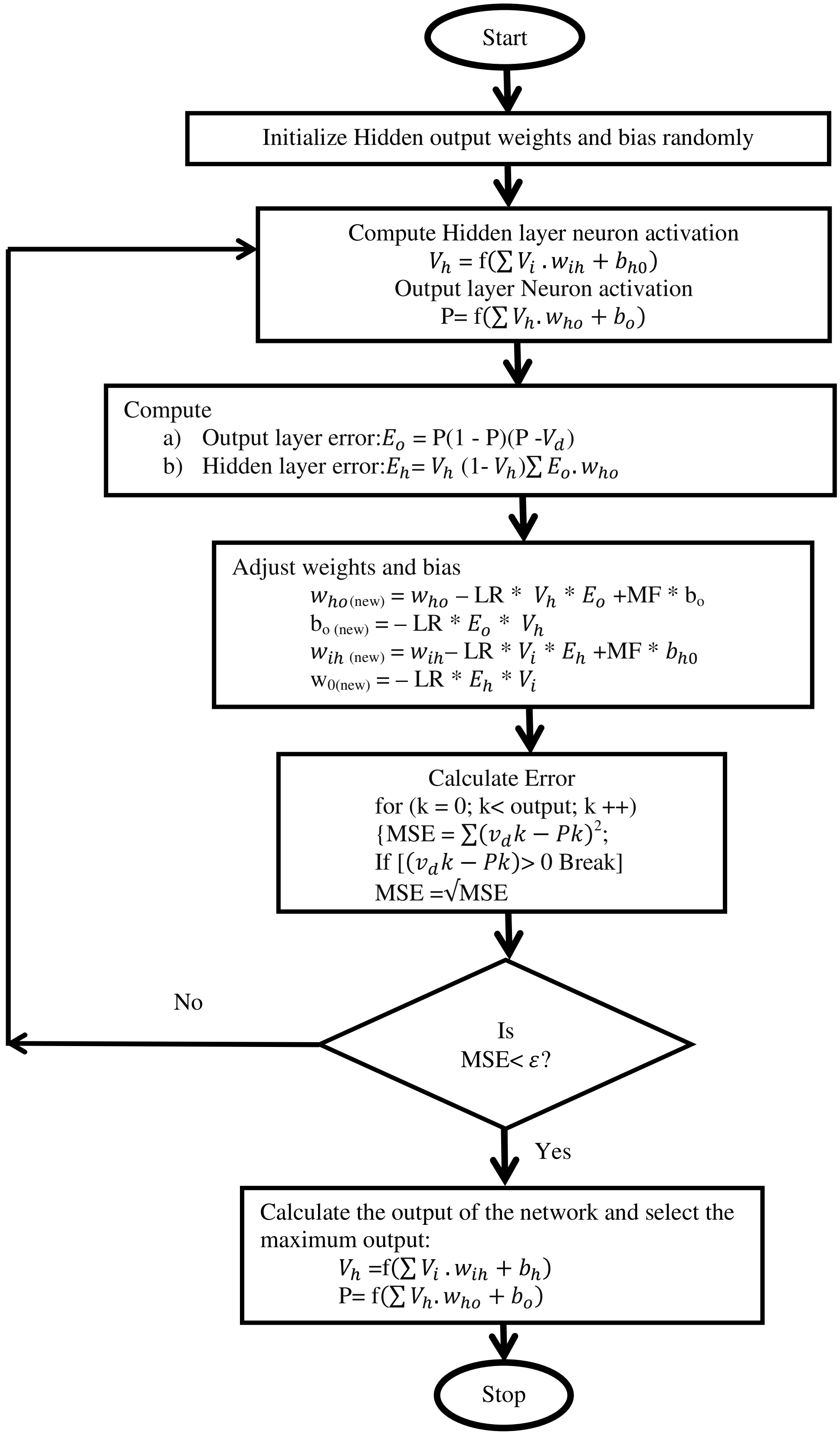

Figure 8: Flowchart for Face Learning and Training in CLBFFNN

3.2.3 Fingerprint Learning and Training in CLBFFNN

The flowchart in Figure 9 gives the algorithm for the fingerprint learning and training in CLBFFNN. The following learning rule is adopted in the training of the network:

Let V(x)(a) be the input vector,

X = Mr. V's fingerprint

A = pattern matrices, 16x16 array (a = 0, 1,2,3…., 256)

G(x)(a) = The target output

W_{ab} and W_{bc}= The weight vectors

Ixa= unit a input for pattern x.

Ix_b=The output of the respective PE which is calculated to be either 1 or 0 using a sigmoid function

Nx_b= The weighted Sum

atv_b = The activation function

yf_b= the hidden threshold weight for bth PEs

C1 = called the spread factors

yI_c= output threshold weight for cth output

NTW = new threshold weight

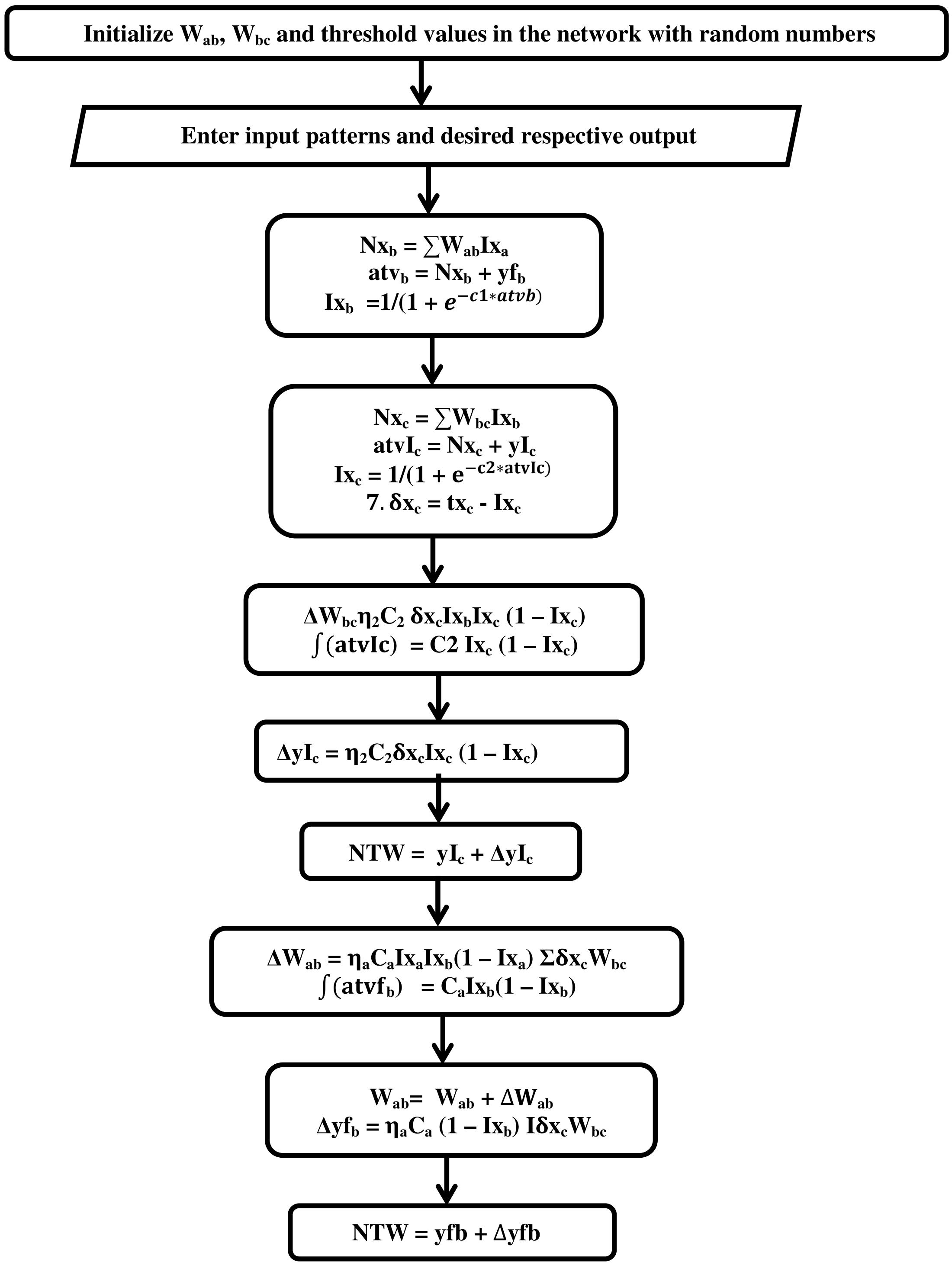

Figure 9: Flow Diagram for Fingerprint Learning and Training Using CLBFFNN

3.2.4 Iris Learning and Training in CLBFFNN

After iris image normalization, the feature vectors are fed into CLBFFNN for network learning and training. After series of training and weight adjustments, the network training stops when the computed error is at its minimum. The system is evaluated by feeding the trained network with a test data; to prove its efficiency in handling noisy set of data. Figure 10shows the NN learning and training algorithm for iris feature.

The variables used include:

thd = threshold
S = security band
N = the numbers of characteristics
L1 and *L2*= learning rates
Stop = stopping criterion
C(i,j) = a comparison code
S(C) = a similarity score
ICC = the set of imposter comparison codes
GCC = the set of genuine comparison codes

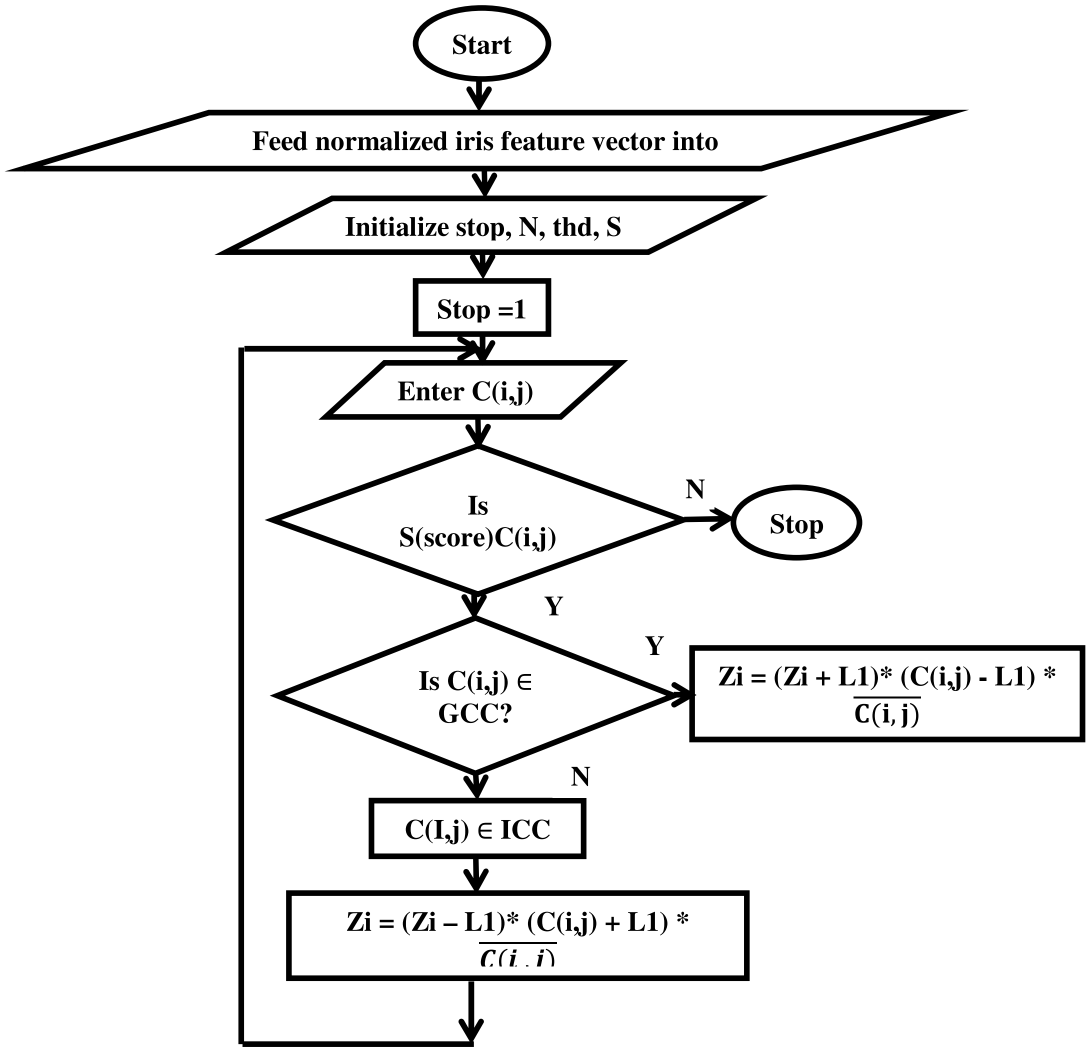

Figure 4.5: Iris Learning and Training Using CLBFFNN

3.3 Overtraining Check Facility

Adopting the use of the Overtraining Check Facility helps to determine when to stop network training. This prevents network overtraining (i.e. network being extremely correct). This lowers the efficiency of the biometric system because it will lack the capability of handling noisy versions of data. This facility is ensured by having another set of a training data containing negative attributes (such as distorted or blurred images). This set is called a validation set. It is not involved in the network training rather acts as a check against network overtraining. After each training session, the validation set is used to calculate an error. Network training stops when validation error attains its minimum value, but when the error starts increasing, it then means that the network is being over-trained.

3.4 Comparison of CLBFFNN with Other Methods

The comparison of CLBFFNN with other methods is done in MATLAB 7.5

environment. CLBFFNN is seen to have a better recognition performance when compared with Adaptive Principal Component Analysis (APCA) and Multilayer Perceptron (MLP) methods [30] as seen in Figure 11. CLBFFNN approach also gives better performance than that of Long & Thai [31]who used RVM approach, as seen in figure 12.

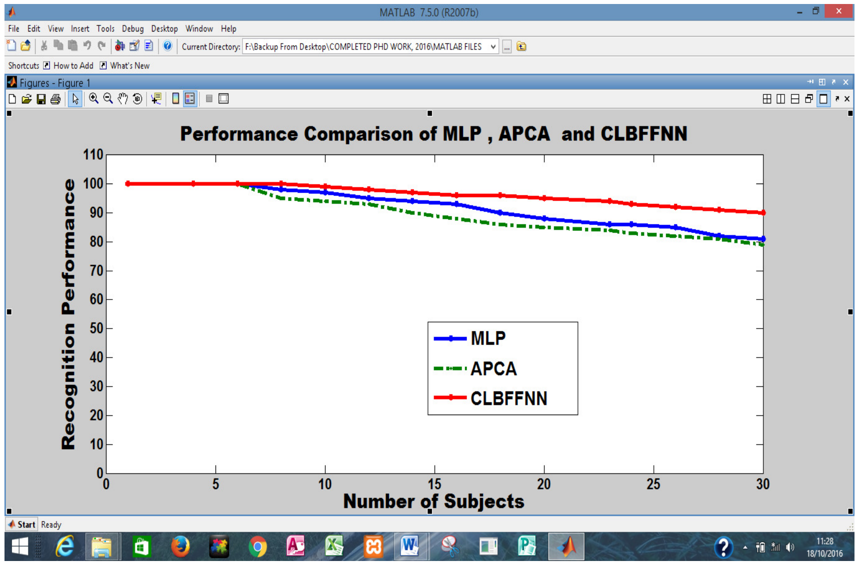

Figure 11: Performance Comparison MLP, APCA and CLBFFNN

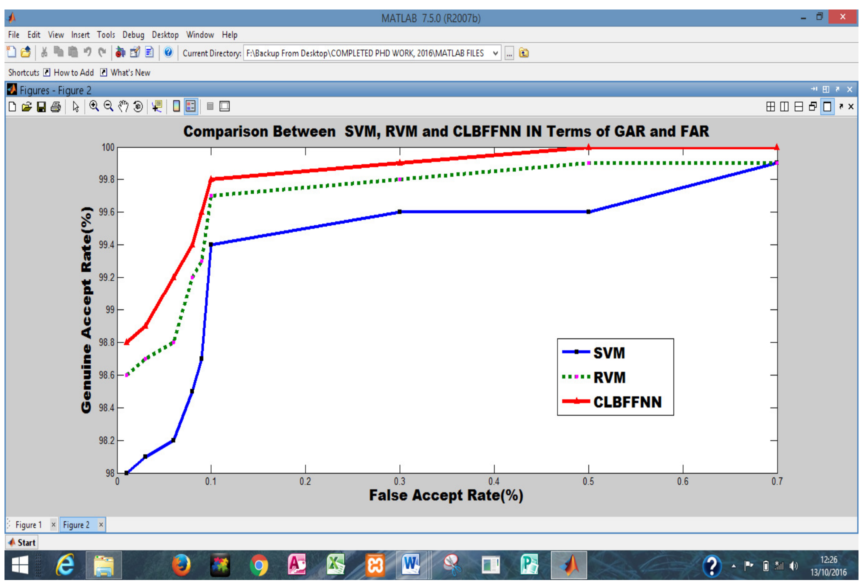

Figure 12: Comparing SVM and RVM Classifiers with CLBFFNN

4. CONCLUSIONS

CLBFFNN is an intelligent multi-biometric system classifier and has input layer, hidden layer and output layer, like every other artificial neural network. It is an ordered cascade of two neural networks: CLBFFNN(1) and CLBFFNN(2). The use of a Cascaded Link-Based Feed Forward Neural Network (CLBFFNN) with Back Propagation algorithms offers effective training of the network with reduced FAR and FRR. Security enhancement of the algorithms is enforced by the use of liveness checks, biometric data quality checks and overtraining check facilities.

5. ACKNOWLEDGEMENTS

The authors would like to appreciate the reviewers of this paper, for their useful comments and contributions which added to the quality of this work.

REFERENCES

[1] El-Zoghabi A. A., Yassin A. H., Hussien H. H. (2013). Survey Report on Cryptography Based on Neural Network.International Journal of Emerging Technology and Advanced Engineering. Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 12, December 2013), 456 -462.

[2] Jogdand R. M. and BisalapurS. S. (2011). Design of An Efficient Neural Key Generation. International Journal of Artificial Intelligence & Applications (IJAIA), Vol.2, No.1, 60-69, January 2011. DOI : 10.5121/ijaia.2011.2105 60 (PDF). Available from: https://www.researchgate.net/publication/49612116_Design_of_An_Efficient_Neural_Key_Generation

[3] Dilek S., Çakır H. and AydınM. (2015). Applications of Artificial Intelligence techniques To Combating Cyber Crimes: A Review. International Journal of Artificial Intelligence & Applications (IJAIA), Vol. 6, No. 1, January 2015. 21-39

[4] Bagrow, J. P., Lehmann, S., &Ahn, Y-Y. (2015). Robustness and modular structure in networks. Network Science , 3(04), 509-525. DOI: 10.1017/nws.2015.21

[5] Debashish D. and Mohammad S. U. (2013). Data Mining And Neural Network Techniques In Stock Market Prediction:A Methodological Review. International Journal of Artificial Intelligence & Applications (IJAIA), Vol.4, No.1, January 2013 DOI : 10.5121/ijaia.2013.4109, 117 – 127.

[6] Chiang Y., Chang L., Chang F. (2004). Comparison of static-feed-forward and dynamic-feed-backneural networks for rainfall – runoff modelling. ELSEVIER, Journal of Hydrology 290 (2004) 297–311.

[7] Islam M. J., AhmadiM. and Sid-Ahmed M. A. (2010). An Efficient Automatic Mass Classification Method In Digitized Mammograms Using Artificial Neural Network. International Journal of Artificial Intelligence & Applications (IJAIA), Vol.1, No.3, July 2010, 1-13. DOI : 10.5121/ijaia.2010.1301

[8] Khaze S. R., MasdariM. and Hojjatkhah S. (2013).Application of Artificial Neural Networks in Estimating Participation In Elections. International Journal of Information Technology, Modelling and Computing (IJITMC) Vol.1, No.3, August 2013.DOI : 10.5121/ijitmc.2013.1303, 23-31

[9] Li Y., Wang K., and Zhang D. (2002). Step acceleration based training algorithm for feed-forward neural networks. In IntConf Pattern Recognition (ICPR), pages 84–87.

[10] Levin E., Tishby N., and Solla S.A. (1990). A statistical approach to learning and generalization in layered neural networks. IEEE, 78(10):1568–1574.

[11] JuangB. and Katagiri S. (1992).Discriminative learning for minimum error classification.IEEE Trans Signal Processing, 40(12):3043–3054.

[12] Lee K. Y. and Sode-YomeA. and Park J. H. (1998).Adaptive Hopfield Neural Networksfor Economic Load Dispatch. IEEE Transactions on Power Systems, Vol. 13, No. 2, May1998 519-526

[13] Hopfield J. (1982). Neural networks and physical systems with emergent collective computational abilities. National Academy of Science USA, 79(8):2554 –2558.

[14] Yao X. (1999). Evolving artificial neural networks.Proceedings of the IEEE, 87:1423 – 1447.

[15] Ludmila I. Kuncheva (2004).Combining Pattern Classifiers - Methods and Algorithms.Wiley.

[16] Tang E. K., Suganthan P. N., and Yao X.(2006). An analysis of diversity measures.Machine Learning, 65:247–271.
[17] Krenker A., Bester J. and Kos A. (2011). Introduction to the Artificial Neural Networks, Artificial Neural Networks - Methodological Advances and Biomedical Applications, Prof. Kenji Suzuki (Ed.), ISBN:978-953-307-243-2, InTech, Available from: http://www.intechopen.com/books/artificial-neural-networks-methodological-advances-and-biomedical-applications/introduction-to-the-artificial-neural-networks.
[18] Shubhangi S. A. and Madhuri A. C. (2012). Artificial Neural Network Controller for Performance Optimization of Single Phase Inverter.International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.1, January 2012. DOI : 10.5121/ijaia.2012.3105. 53-64.
[19] Vijaya, S. (2012). A study on the neural network model for finger print recognition. International Journal of Computational Engineering Research (ijceronline.com)2 (5).
[20] Le Cun, V., Bottou, L., Bengio, Y., and Haffner, P., (2012). Handwritten digit recognition with a back propagation network. Neural Information Processing Systems, 2: 396-404.
[21] Fausett, L. (1994). Fundamentals of neural networks. New York: Prentice Hall. ACM, New York, 89: 151– 158.
[22] Schmidhuber J. (2015) Neural Networks. ELSEVIER 61(2015) 85-117
[23] Ritu, M. G. (2014).A review on fingerprint-based identification system.International Journal of Advanced Research in Computer and Communication Engineering 3(3).Copyright to IJARCCE www.ijarcce.com 5849.
[24] Hazem, M. E. (2002). Face detection using neural networks and image decomposition. Lecture Notes in Computer Science 22: 205-215.
[25] Askarunisa, A., Sankaranarayanan, K., Sundaram, R., and Sathick, M. B. (2009).Fingerprint authentication using neural networks, MASAUM Journal of Computing, 15(2): 234.
[26] Ross, A. and Jain, A. K. (2003). Information fusion in biometrics.PatternRecognition Letters, 24(13): 2115-2125.
[27] Benson-Emenike, M. E. &Sam-Ekeke Doris C. Trimodal Biometric Authentication System using Cascaded Link-based Feed forward Neural Network [CLBFFNN]. International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA 12 (5): 7-19 August 2017 – www.ijais.org
[28] Salim L. (2011). A comparative study of back propagation algorithms in financial prediction. International Journal of Computer Science, Engineering and Applications (IJCSEA),1(4).
[29] Devika, C., Amita, S. and Manish, G. (2013). Recapitulation on Transformations in Neural Network Back Propagation Algorithm.International Journal of Information and Computation Technology 3(4): 323-328
[30] Nayak, P.K. and Narayan, D., (2013). Multimodal biometric face and fingerprint recognition using adaptive principal component analysis and multilayer perception.International Journal of Research in Computer and Communication Technology, 2(6).
[31] Long, B.T., and Thai, H. L., (2015). Person authentication using relevance vector machine (RVM) for face and fingerprint.I. J. Modern education and Computer Science, 5, 8-15.

Authors

Benson-Emenike Mercy E. has a doctorate degree and Master's degree in Computer Science from University of Port Harcourt, Nigeria. She obtained her Bachelor of Technology degree [B.Tech] from Federal University of Technology, Minna, Niger state. She is a lecturer in the Department of computer Science, Abia State Polytechnic and an adjunct lecturer in Computer science Department, Rhema University Nigeria and National Open University of Nigeria [NOUN]. She is a member of Computer Professionals (Registration Council) of Nigeria (CPN) and Nigeria Computer Society (NCS). Her publications have appeared in an international journal and conference proceedings. Her research interests include Artificial Intelligence, Biometrics, Operating System, and Information Technology.

Ifeanyi-Reuben Nkechi J. has a doctorate degree in Computer Science from the University of Port-Harcourt Nigeria. She obtained her M.Sc. and B.Sc. in Computer Science from the University of Ibadan, Nigeria and University of Calabar, Nigeria respectively. She is a lecturer at the Department of Computer Science, Rhema University Nigeria. She is a member of Computer Professionals (R egistration Council) of Nigeria (CPN), Nigeria Computer Society (NCS) and Nigeria Women in Information Technology (NIWIIT). Her research interests include Database, Data mining, Text mining, Information Retrieval and Natural Language Processing.