VOL. 15, NO. 3, FEBRUARY 2020 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

CONFIGURING APPROPRIATE ARTIFICIAL NEURAL NETWORK WITH MONARCH BUTTERFLY OPTIMIZATION FOR SPEAKER RECOGNITION

N. Dhana Lakshmi[1] and M. Satya Sai Ram[2]
[1]Chaitanya Bharathi Institute of Technology, Hyderabad, Telangana, India
[1]Acharya Nagarjuna University, Guntur, Andhra Pradesh, India
[2]RVR & JC College of Engineering, Guntur, Andhra Pradesh, India
E-Mail: dhananm@gmail.com

ABSTRACT

Identification of person voice from their characteristics is said to be speaker recognition; it is generally utilize in telecommunication, voice biometrics, criminal investigations and various industrial applications. To pursue, Mel-Frequency Cepstral Coefficient (MFCC) and Linear Prediction-filter Coefficients (LPC) are utilizes to extract features from voice signal as preliminary process. Subsequently, these features employ to Artificial Neural Network (ANN) to recognize speaker. This research includes configuring conventional ANN structure holding single hidden layer associate with ten neurons. To conserve time and process complexity optimization techniques incorporates to identify appropriate configuration for performance enhancement. An optimization technique includes Evolutionary Algorithm (EA), Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Monarch Butterfly Optimization (MBO). The comparative techniques are K Nearest Neighbour (KNN), Random Forest (RF), Radial Basis Neural Network (RBNN), Back-Propagation Neural Network (BPNN) and Generalized Regression Neural Network (GRNN). The result reveals the performance of MBO in configuring ANN accomplishes 99% accuracy in real time database and 99.8% in benchmark database which is superior results over contest techniques.

Keywords: speaker recognition, mel-frequency cepstral coefficient (MFCC), linear prediction-filter coefficients (LPC), artificial neural Network (ANN) and monarch Butterfly optimization (MBO).

1. INTRODUCTION

Speaker recognition is an explored region for the scientists of speech processing. Speaker recognition is undertaking of distinguishing individual from properties of speech samples [1]. Speaker recognition is a vital apparatus for innumerable applications namely access control and client security protection [2]. The speech signal carries essential data namely message content, language, speaker identity, speaker emotion, speaker personality, and so on [3]. Recent advances in voice examination have empowered the visualization of vocal fold dynamics related with particular voice disorders [4]. Recognition is generally more difficult when vocabularies are large or have many similar-sounding words. The objective of this work is to overcome this disadvantage and recognize the speaker. Implement a Speaker Recognition System (SRS) utilizing LPC and MFCC as feature extraction methods [5]. The model parameters are regularly learned on MFCC based front-end parameterization of speech signals. The MFCC is a popular feature in Automatic Speech Recognition (ASR) and is inferred following static (non-signal dependent) processing methods [6]. A LPC gives a decent model of the speech signal. This is particularly valid for the quasi steady state voiced regions of speech in which all-pole model of LPC give a good approximation to the vocal tract spectral envelope [7]. Different classifiers are likewise accessible for SR Snamely Support Vector Machine (SVM), Hidden Markov Model (HMM), Kernel Regression and K Nearest Neighbor (KNN), Maximum Likelihood Classifier (MLC) and ANN [8].

This work is concerned with ANNs that have turned out to be an intense pattern recognition tools effectively utilized for some real world applications in the course of the last few years [9].ANN has been characterized from multiple points of view by a few researchers. However, the essential truth they all concur is that, neural networks are prepared of several processing units named neurons. These processing units are trained utilizing input–output datasets introduced to the network. After the training procedure, the network makes suitable results when tested with comparable data sets, in other words, recognizes the presented patterns [10]. The major advantage of the ANN is the exchange work between the input vectors and the objective matrix does not need to be predicted in advance [11].So choosing the number of hidden layers and number of neurons in each hidden layer is a testing problem while considering any complex trouble [12]. Optimizing the number of hidden neurons to utilize without a pre-set focus for accuracy is one of the real difficulties for neural networks, generally referred to as the bias/variance dilemma [13]. The primary targets to minimize error, enhance accuracy and stability of network. This survey is to be valuable for scientists working in this field and chooses appropriate number of hidden neurons in neural networks. For optimizing the hidden layer and neuron, dissimilar optimization methods are used namely Evolutionary Algorithm (EA), Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Monarch Butterfly Optimization (MBO) [14]. Simulating the migratory behavior of monarch butterflies in nature with the seasons, MBO approaches as a new biologically

inspired computing method. The first MBO has demonstrated good performance in dealing with numerical optimization and combinatorial optimization problems [15] Abhay Kumar *et al.* [16] had planned to enhance the outcome of speech recognition in contrast with other algorithm which include formant as feature vector and KNN, linear discriminant analysis (LDA),Tree, and quadrature discriminant analysis(QDA) as machine learning algorithms. In that paper analysis and comparison was done between feature vector motivated by speech generation and hearing model to enhance the outcome of recognition for larger database of twenty words. Bright Kanisha, Ganesan Balarishnanan [17] had anticipated speech recognition applications are ending up progressively powerful. From the outcomes, the optimization algorithm Adaptive Particle Swarm Optimization (APSO) attains the 97.8% accuracy contrasted with the existing method SVM linear kernel function. Gai-Ge Wang *et al.* [18] had proposed by simplifying and idealizing the migration of monarch butterflies, another sort of nature-inspired metaheuristic algorithm, called MBO. The outcomes obviously display the capacity of the MBO technique toward finding the improved function values on large portion of the benchmark problems with as for the other five algorithms. Khaled Daqrouq, Tarek A. Tutunji [19] had projected a new technique for speaker feature extraction based on formants. Probabilistic neural network was additionally proposed for comparison. The outcomes are additionally contrasted with surely understood established algorithms for speaker recognition and are observed to be predominant. Seyed Reza Shahamiri, Siti Salwah Binti Salim *et al.* [20] had proposed that paper examines the utilization of ANNs as a fixed-length isolated-word Speaker Independent (SI) ASR for people who suffer from dysarthria. The outcomes demonstrate that the speech recognizers trained by the conventional 12 coefficients MFCC features without the utilization of delta and acceleration features gave the best accuracy, and the proposed SI ASR recognized the speech of the unexpected dysarthric assessment subjects with word recognition rate of 68.38%.

This paper organized as follows, section-2 illustrate proposed methodology subsequently section-3 illustrate the investigation on results and discussion eventually wind-up with conclusion in section 4.

2. RESEARCH METHODOLOGY

This research emphasis recognition of speaker voice with the aid of refurbishes ANN in association with optimization techniques. Here utilize two types of database preliminary with 980 standard *TIMIT* voice signal database and 200 real time voice signal datasets; in standard database includes 98 speakers out of which 29 female and 69 male voice signal for ten words respectively. On other hand, real time database 20 speakers out of which 8 female and 12 male voice signal for ten words respectively. A mentioned this research intends to utilize Artificial Intelligence (AI) techniques namely K-Nearest Neighbour (KNN), Random Forest (RF), Radial Basis Neural Network (RBNN), Back-Propagation Neural Network (BPNN) and Generalized Regression Neural Network (GRNN) [11] and Artificial Neural Network (ANN). ANN utilize default (single hidden layer with ten neurons), this sort of network structure unveil inadequate performance in the context of predicting speaker recognition. This urges the research intention to configure conventional ANN structure by predicting appropriate number of hidden layers and neurons necessary in the context of predicting speaker recognition. To pursue this process through manually take a long time to compute this intend to incorporate optimization technique to predict appropriate number of hidden layers and neurons. The considered optimization techniques are Genetic Algorithm (GA), Evolutionary Algorithm (EA), Particle Swarm Optimization (PSO) and Monarch Butterfly Optimization (MBO). In general, ANN utilizes Levenberg–Marquardt (LM) as a training technique to execute in both conventional and refurbished structure. Here utilize 80% of datasets for training and remaining 20% for testing the configured structure.

2.1 Feature extraction

Feature extraction plays a vital role on recognition speaker voice, this process execute with the aid of couple of noteworthy techniques called MFCC and LPC. The following section details the process of both feature extraction techniques and their significance in speaker recognition.

2.1.1 Mel frequency cepstral coefficients (MFCC)

In 1980, Davis and Murmelstein introduce MFCC, which is extensively utilize in automatic speech and speaker recognition environment. The following block diagram (Figure-1) detail the working process of MFCC in the platform of speaker recognition.

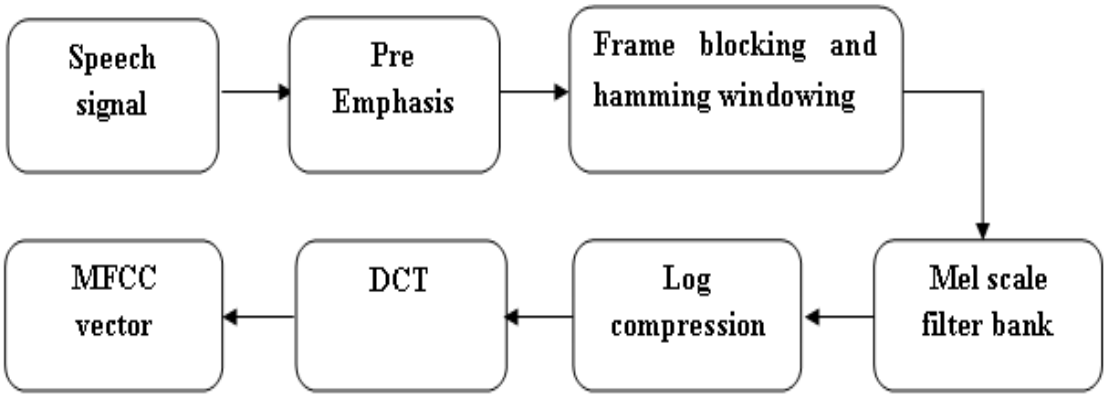

Figure-1. MFCC process.

Pre-Emphasis

Pre-Emphasis is the progression to amplify the magnitude of frequencies as for the magnitude of different frequencies. The fundamental thought process behind the procedure of pre-emphasis is contribute on scaling up the magnitude of certain higher frequencies in connection to those of different lower frequencies with an eye on increasing the general signal-to-noise ratio (SNR) value. The input speech signal furnish to a filter for guaranteeing higher frequencies. The related system well set to improve the energy of the input speech signal at higher frequency. The speech signal initially pre-emphasized by a first order Finite Impulse Response (FIR) filter with pre-emphasis coefficient β.Transfer function in z domain for first order FIR filter is represent as,

$$F(z) = 1 - \beta z^{-1} \tag{1}$$

The pre-emphasis coefficientβlies within the range0≤δ≤1.

$$e(v_i^r{}') = \rho(v_i^r{}') - \beta\rho(v_i^r{}'-1) \tag{2}$$

$$e(v_j^t{}') = \rho(v_j^t{}') - \beta\rho(v_j^t{}'-1) \tag{3}$$

For training, the word equation (2) is utilize and for testing the word, equation (3) is utilize. Where, pre-emphasis coefficient represented by β,ν in equation (2) represents training word and ν in equation (3) represents testing the word.

Frame blocking and hamming windowing

The arithmetic features of a speech signal are invariant only with in short time intervals. Presently, the adjacent frames being separate by f_A samples (frame shift) and the pre-emphasized signal is block into frames of f_S samples (frame size). If thekth frame of speech is,$x_k(v_i^{r'})$, $x_k(v_j^{t'})$and there are k frames inside the whole speech signal, at that point

$$x_k(v_i^r{}') - \rho(f_{A^r} + v_i^r{}'),\ \ 0 \le v_i^r{}' \le f_{A^r} - 1 \tag{4}$$

In windowing, every one of the above frames is multiply with a hamming window in order to keep continuity of the signal to reduce the signal discontinuities from beginning to end of the frames while each frame is windowed. To tape the signal at the edges of each frame windows picked. The paramount choice in the speech recognition is Hamming window, which coordinates all the connecting frequency lines. The hamming window equation represented as,

$$y(n) = x(n) \times w(n) \tag{5}$$

Where w(n) is a window function

$$w(n) = 0.54 - 0.46\cos\left(2\prod\frac{n}{N}\right) \qquad 0 \le n \le N \tag{6}$$

Mel scale filter bank

With an eye on changing every single time domain frame off $_S$ samples into frequency domain, the filter bank examination is viably completed. For modifying the convolution of the glottal pulse and the vocal tract, impulse response in the time domain into frequency domain the Fourier Transform is richly utilize. In the FFT spectrum, as the frequency range tends to be too wide, the voice signal fails to follow the linear scale. With the aim of assessing a weighted whole of filter spectral modules, a set of triangular filters are sent in order to rough the output of procedure to Mel scale. At the centre frequency, the magnitude frequency response of every single filter is triangular and equivalent to unity and tends directly to zero at centre frequency of two neighbouring filters. In this way, the total of its filtered spectral modules rises as the output of each filter. The filters taken in general prevalently known as a Mel scale filter bank and the perceptual processing arranged inside the ear is provoked by the frequency response of the filter bank. In this manner, the ensuing equation utilizes to assess the Mel for the pre-specified frequency f in HZ.

$$F(Mel) = 2595 \times \log_{10}\left(1 + \frac{f}{700}\right) \tag{7}$$

Logarithmic compression

Presently, the filter outputs got from filter bank investigation is compress by the logarithmic function. The f_a^{th} filter logarithmically compressed output is express as,

$$X_{f_{a^r}(\ln)} = \ln(X_{f_{a^r}}), \qquad 1 \le f_{a^r} \le f_{A^r} \tag{8}$$

Discrete cosine transformation (DCT)

This is the procedure to convert the log Mel spectrum into time domain utilizing DCT. The set of coefficient is acoustic vectors. At that point, DCT connected to the filter outputs and of a particular speech frame, the first few coefficients assembled together as a feature vector. Thekth MFCC coefficient represent as,

$$y(k) = w(k)\sum_{n=1} x(n)\cos\left(\frac{\Pi}{2N}(2n-1)(k-1)\right) \qquad k = 1,2,..N \tag{9}$$

Where

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}} & k = 1 \\ \sqrt{\frac{2}{N}} & 2 \le k \le N \end{cases} \tag{10}$$

2.1.2 Linear prediction-filter coefficients (LPC)

LPC decides the coefficients of a forward linear predictor by minimizing the prediction error in the least squares sense. It has functions in filter design and speech coding.[a,g] = LPC(y,p) finds the coefficients of a p^{th}-order linear predictor (FIR filter) that predicts the present value of the real-valued time series y based on past samples.

$$\hat{y}(n) = -a(2)\,y(n-1) - a(3)\,y(n-2) - .... - a(p+1)\,y(n-p) \tag{11}$$

p is the order of the prediction filter polynomial, a = [1 a(2) ... a(p+1)]. If p is unspecified, LPC utilizes as a default p = length(y)-1. On the off chance thatx is a matrix containing a different signal in every column, LPC returns a model estimate for each column in the rows of matrix a and a column vector of prediction error variances g. The length of p must be not exactly or equivalent to the length of y.

2.2 Artificial neural network (ANN)

ANN is a real learning framework used in psychological brain science and AI. The ANN is an adjusted computational model that hopes to copy the neural structure and working of the human cerebrum. It is included an interconnected structure of artificially created neurons that capacity as pathways for information exchange. ANN is versatile and adaptable, learning and changing with each distinctive interior or external stimulus. ANN utilize as a piece of succession and pattern recognition systems, data processing, robotics, and modelling.

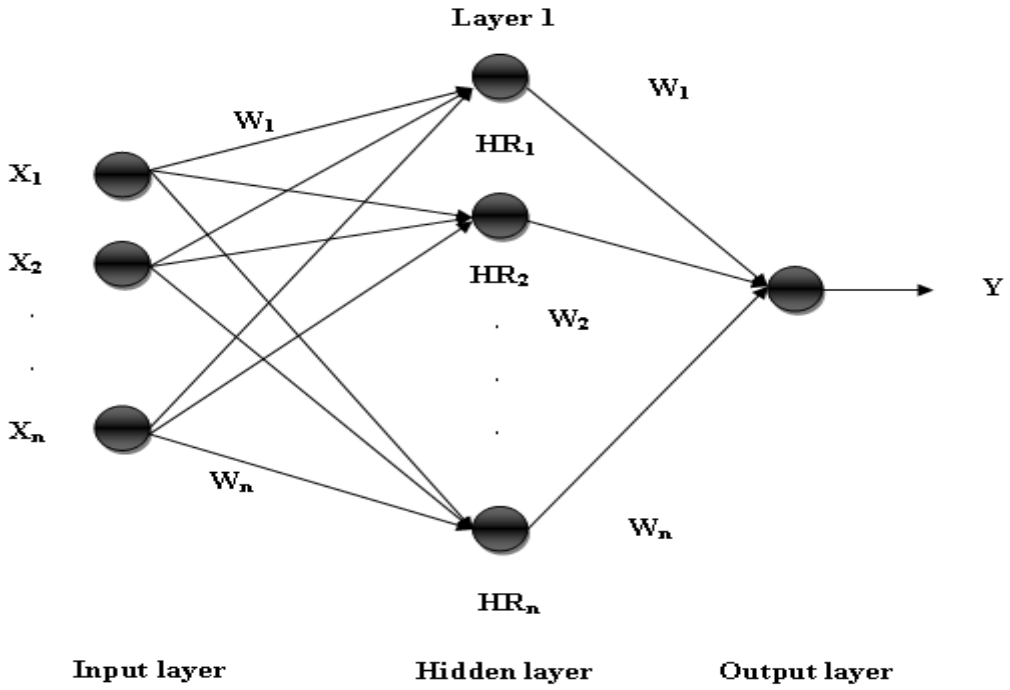

Figure-2. Neural network structure.

In general, Figure-2 the ANN comprises of three layers, for example the input layer, hidden layer and the output layer. Each layer of the ANN comprises of number of neurons. In this, every neuron in the input layer is associated with the hidden layer neuron and each hidden layer neuron is associated with the output layer with a random weight. The random weights appointed to each interconnected layer.

2.2.1 Input layer

Here, 61 features attributes consider to be feeding input in initial layer called input layer G_1, G_2, up to G_n applied to the input neuronsu$_1$, u_2 up to u_n each neurons in the input layer is connected with the hidden layer neurons with random weightw$_{11}$, w$_{12}$...w$_{ij.}$

2.2.2 Hidden layer

Here, default structure comprised of single hidden layer associate with ten neurons, whereas to enhance the performance of prediction refurbish is necessary. To refurbish ANN structure by configuring optimal number of hidden layer and associate neurons via manual take prolong time to compute, this urge incorporation of optimization technique to conserve the complex process. The quantity of neurons in the hidden layer is depicted as HR_1, HR_2,….HR_n, which are connected with the output layer neuron. Each neuron in the hidden layer are connected with the output layer neurons with an irregular weight w$_{11}$, w$_{12}$,…..w$_{ij.}$ Multilayer perceptron is a sigmoidal activation function in the form of a hyperbolic tangent.

The basic function of hidden neurons is assess based on the equation beneath,

$$P_f = \sum_{j=1}^{N} G_i \times w_{ij} \tag{12}$$

Where P_f is a basics function, w_{ij} is an input layer weight andi is a number of input with this basic function the active function combined.

The activation function utilize sigmoid function is practiced in view of the condition demonstrated as follows:

$$\tan sig\,(P_f) = \frac{2}{(1+\exp\,(-2*P_f))-1} \tag{13}$$

2.2.2.1 Levenberg–Marquardt algorithm

The LM algorithm is the most by and ordinarily used training optimization algorithms. It vanquishes facilitate tendency drop and other conjugate point approaches in a wide aggregation of issues. LM algorithm is a minute asks for those endeavouring to train neural networks, maybe in light of how it is more complicated to perform than EBP, dismiss an algorithm those assorted occasions. In any case, it verifiably compensates for this in unrivalled execution. LM takes after EBP in that it includes the figuring of the inclination vector, so far in like way, LM also chooses the Jacobian.

The gradient vector is represented as:

$$p = \begin{Bmatrix} \frac{\partial m}{\partial w_1} \\ \frac{\partial m}{\partial w_2} \\ . \\ . \\ . \\ \frac{\partial m}{\partial w_n} \end{Bmatrix} \tag{14}$$

Where m is there error of the network for that model and w suggests to the weights. The Jacobian is basically, every incline for each training model and network output. The Jacobian is revealed beneath.

$$JC = \begin{Bmatrix} \frac{\partial m_1}{\partial w_1} & \frac{\partial m_1}{\partial w_2} & \cdots & \frac{\partial m_1}{\partial w_n} \\ ..... & ..... & & ..... \\ \frac{\partial m_2}{\partial w_2} & \frac{\partial m_2}{\partial w_2} & ..... & \frac{\partial m_2}{\partial w_n} \\ ..... & ..... & & ..... \\ \frac{\partial m_n}{\partial w_1} & \frac{\partial m_n}{\partial w_2} & ..... & \frac{\partial m_n}{\partial w_n} \end{Bmatrix} \tag{15}$$

Where, N is the number of weights and I is the number of patterns.

The Jacobian represent as, the consequent can recognize the LM algorithm:

$$W_{XY} = W_X - (JC_X{}^T\, JC_X + \mu IE)^{-1}\, JC_X{}^T\, m \tag{16}$$

Where, m is the total confuse for all portrayals IE is the character structure, and l is a learning parameter. The learning parameter l is then balance a little time in each iteration and the result in the best decay of blunder is picked. Right, when the l regard is tremendous, the LM algorithm affects the chance to be steepest great or BP, and when l is proportionate to zero it is the Newton Method. Entire structure is then rehashed until the moment when the slip-up is diminished to the required worth. This second request algorithm is basically, speedier than BP. For small networks with few training designs, this is not a fundamental issue, yet rather, for structures with various training designs, it is computationally raised. This inversion will fulfil that each training iteration for LM to take longer than iteration for BP. The time required for getting ready will at display is far not as much as that of BP in light of the fact that the LM will require such little iteration.

2.2.3 Monarch butterfly optimization (MBO)

MBO is normally metaheuristic algorithm, proposed by Wang in 2015. It is inspire by the behaviour of monarch butterfly through migration. In MBO, each individual monarch butterfly idealizes and situated in two terrains simply: northern USA and southern Canada (Land 1) and Mexico (Land 2). The places of the monarch butterflies are update in two different ways: the migration operator and the butterfly-adjusting operator.

Initialization

Here, the solution position two attributes hidden layers and neurons whereas the hidden layer range from one to five and neurons range from 1 to 30. Based on randomly allot value in the first position for hidden layer the solution length declare. For instance, if randomly allocate hidden layer value is 3, then the solution length will be 3+1. Likewise, randomly generate ten solutions further fed to fitness process to evaluate the solution strength.

Fitness function

Fitness process utilizes to evaluate the performance of generated solution, here accuracy considered as fitness function as follows. Correctly classified/ total number of validation data

$$\text{Accuracy} = \frac{\text{Correctly Recognize}}{\text{Total Number of Validation Data}} \quad (17)$$

Migration operator

The migration operator is expected to update the monarch butterfly migration between Land 1 and Land 2, on which monarch butterflies make up subpopulations 1 and 2 exclusively. At first, the number of monarch butterflies in Lands 1 and 2 can be considered as NP1 = ceil (p*NP) and NP2 = NP − NP1, independently. Where NP is the total number of monarch butterflies, p is the ratio of monarch butterflies in Land 1, and ceil(y) rounds y to the nearest entire number more prominent than or equivalent to y. In like way, migration operator can be arranged as

$$y^{t+1}{}_{i,k} = \begin{cases} y^{t}{}_{r1,k} & | r \le p \\ y^{t}{}_{r2,k} & | r > p \end{cases} \quad (18)$$

Wherever $y^{t+1}{}_{i,k}$ represents the k^{th} element of y_i at generation t+1. Basically, $y^{t}{}_{r1,k}$ demonstrates the k^{th} element of y_{r1} at generation t, and $y^{t}{}_{r2,k}$. represents the k^{th} element of y_{r2} at generation t. The current generation number is represented by t. Monarch butterflies (r1 and r2) are chosen randomly from subpopulation 1 and subpopulation 2. The condition variable (r) is discovered as

$$r = rand * peri \quad (19)$$

Where peri demonstrates the migration time frame and is set to 1.2 in the fundamental MBO procedure and rand is a random number got from a uniform distribution.

Butterfly adjusting operator

This operator is used to update the places of the monarch butterflies in subpopulation 2. It can be updated as takes after:

$$y^{t+1}{}_{j,k} = \begin{cases} y^{t}{}_{best,k} & | rand \le p \\ y^{t}{}_{r3,k} & | rand > p \end{cases} \quad (20)$$

Wherevery$^{t+1}{}_{j,k}$ represents the k^{th} element of y_j at generation t + 1; $y^{t}{}_{best,k}$ demonstrates the k^{th} element of y_{best} at generation t, which represents the best location of the monarch butterflies in Lands 1 and 2. By then $y^{t}{}_{r3,k}$ represents the k^{th} element of y_{r3} at generation t; the monarch butterfly r3 is chosen randomly from subpopulation 2. On the off chance that rand > p, there has another progression. The location of the butterfly is moreover updated using Levy flight, if rand > BAR:

$$y^{t+1}{}_{i,k} = y^{t+1}{}_{j,k} + \alpha \times (dy - 0.5) \quad (21)$$

Wherever, the variable BAR is the butterfly-adjusting rate. In the occasion that BAR is littler than a random value, the k^{th} element of y_j at generation t+1 is updated, where α is the weighting factor, as showed up in Equation (22).

$$\alpha = S_{max} / t^2 \quad (22)$$

Where, S_{max} is the maximum walk step. In Equation (21), dy is the walk step of butterfly j that can be considered by Levy flight.

$$dy = Levy(y^{t}{}_{j}) \quad (23)$$

Finally, the recently generated butterfly with best fitness is replacing with its parent and enthused to the next generation; also, it is discarding to stay the population size as it.

3. RESULTS AND DISCUSSIONS

The investigation on recognition of speaker voice with the aid of ANN associated with optimization techniques detailed. Various level of investigation involve in this research to evaluate the performance of proposed methodology. It is quite evident that incorporation of optimization technique in configuring ANN structure discloses superior result over default ANN. The following investigation establishes the superiority of refurbish ANN associate with MBO in different measures. The entire execution process in the working platform MATLAB R2015a (8.5.0.197613) having system configuration i5 processor, 8GB RAM.

3.1 Configuring ANN model with soft computing techniques

The following Tables 1 and 2 illustrate the execution of optimization techniques in predicting appropriate hidden layers and their associate neurons in identifying speaker voice reorganization along with default ANN in the platform of real time and standard database. The investigation determines that the incorporation of optimization (soft computing) techniques in configuring ANN structure have an effective impact over default ANN which is exhibit in the Table-3 and following Figures 5 to 13.A MATLAB representation for optimal configure ANN model from MBO shown in Figures 3 and 4.

Table-1. ANN structure optimal configuration for real time database.

Techniques	Input	Hidden Layers	Neurons	Neurons	Neurons	Output
Default ANN	**61**	**1**	**10**	**-**	**-**	**1**
GA	61	3	21	15	22	1
EA	61	2	27	13	-	1
PSO	61	2	17	20	-	1
MBO	61	3	20	22	23	1

Table-2. ANN structure optimal configuration for standard database.

Techniques	Input	Hidden Layers	Neurons	Neurons	Neurons	Output
Default ANN	**61**	**1**	**10**	**-**	**-**	**1**
GA	61	2	23	20	-	1
EA	61	3	18	23	17	1
PSO	61	3	18	19	21	1
MBO	61	2	23	25	-	1

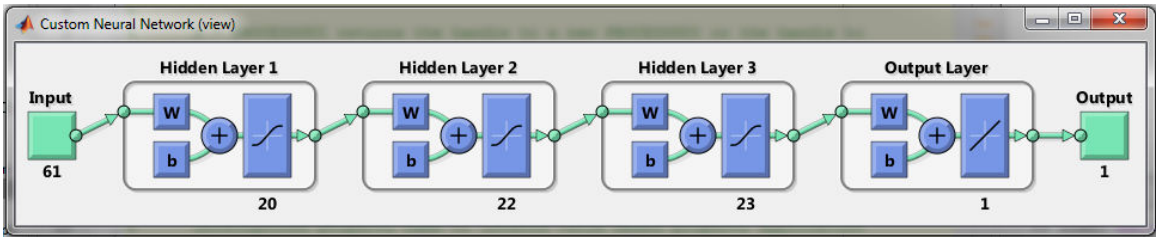

Figure-3. MBO configure ANN model for Real time database.

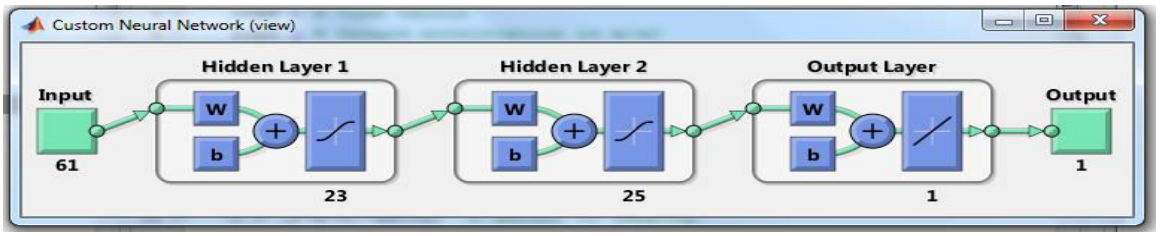

Figure-4. MBO configure ANN model for Standard database.

Table-3. Performance measures from MBO for two different databases.

Measures	MBO	
	Real time database	Standard database
Accuracy	0.99	0.99
FDR	0.1	0.086
FNR	0.1	0.086
FPR	0.005263	0.000894
MCC	0.894737	0.912371
NPV	0.994737	0.999106
PPV	0.9	0.913265
Sensitivity	0.9	0.913265
Specificity	0.994737	0.999106

3.2 Performance of AI techniques with standard measures in different database

The measures like accuracy, False Discover Rate (FDR), False Negative Rate (FNR), False Positive Rate (FPR), Matthews's Correlation Coefficient (MCC), Negative Predictive Value (NPV), Precision Predictive Value (PPV), sensitivity and specificity utilize to evaluate the performance of the employed AI techniques in the context of predicting speaker recognition. The performance measures evident the superiority of MBO utilizes to configure ANN over contest techniques in predicting speaker recognition. The investigated measures computed from true positive (Speaker id correctly identified), true negative (Speaker id correctly rejected), false positive (Speaker id incorrectly identified) and false negative (Speaker id incorrectly rejected). The Figures 5 to 13 illustrates the performance of employed techniques in the platform of real time and standard database for different standard measures. It is apparent in the following graphical representation performance measures that MBO associate in configuring ANN structure exhibits loftier predicting performance than other techniques implement in this research.

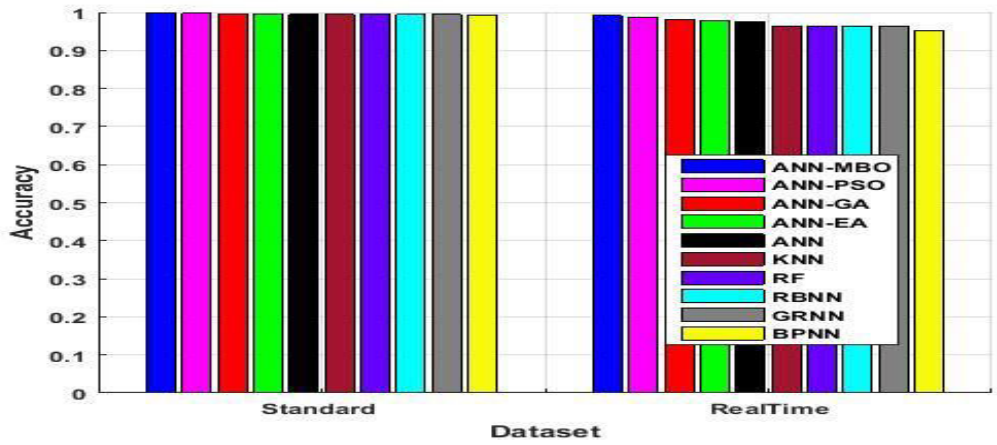

Figure-5. Performance evaluation for Accuracy.

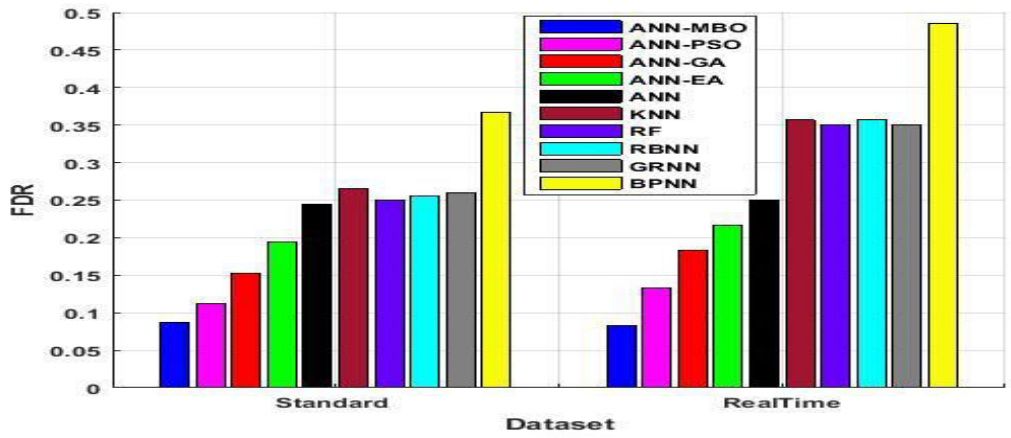

Figure-6. Performance evaluation for False Discover Rate.

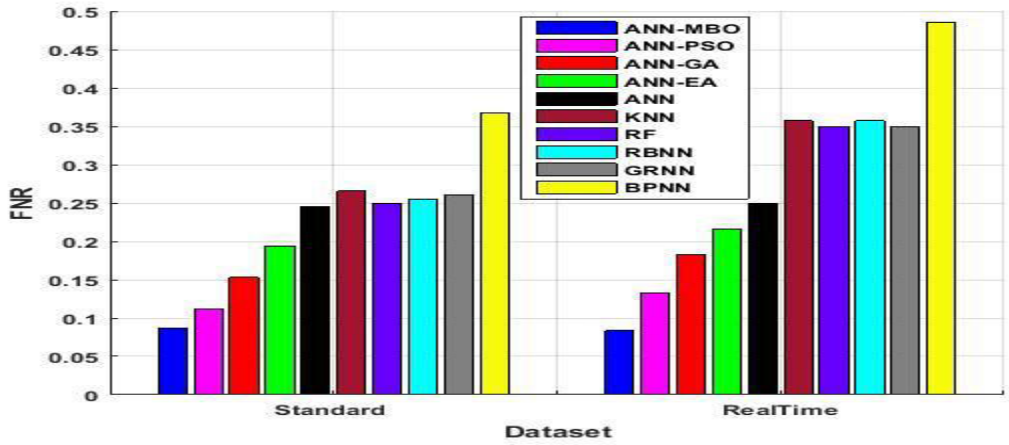

Figure-7. Performance evaluation for False Negative Rate.

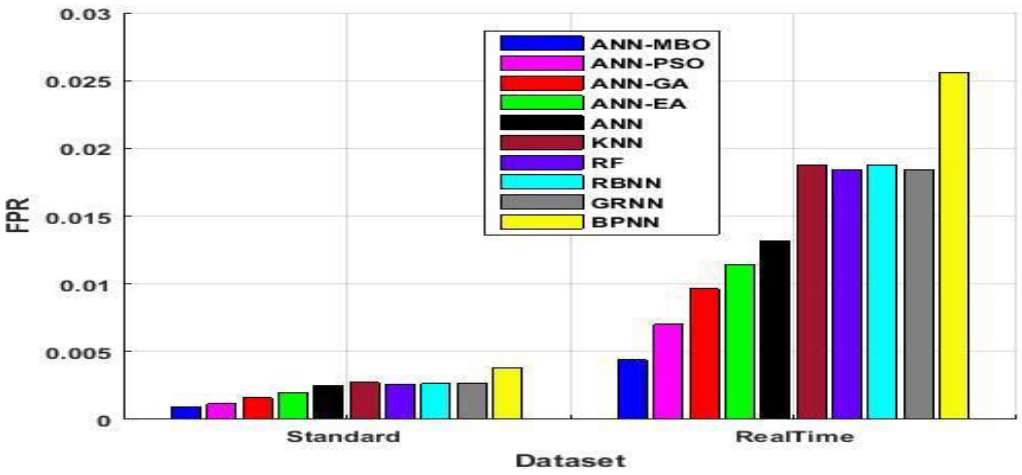

Figure-8. Performance evaluation for False Positive Rate.

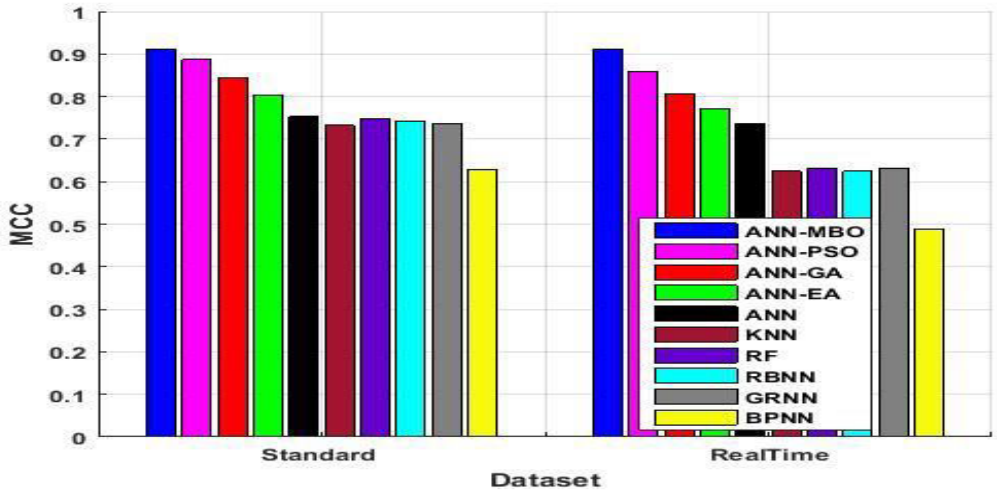

Figure-9. Performance evaluation for Matthews's Correlation Coefficient.

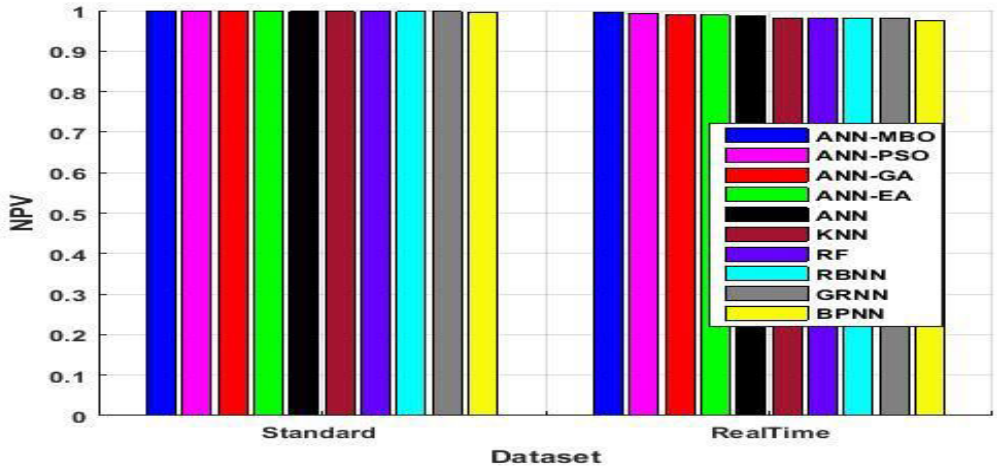

Figure-10. Performance evaluation for Negative Predictive Value.

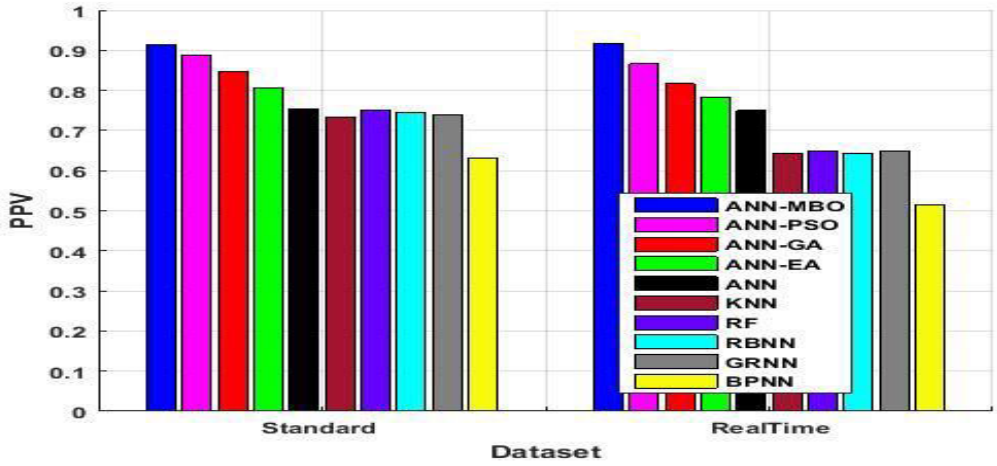

Figure-11. Performance evaluation for Precision Predictive Value.

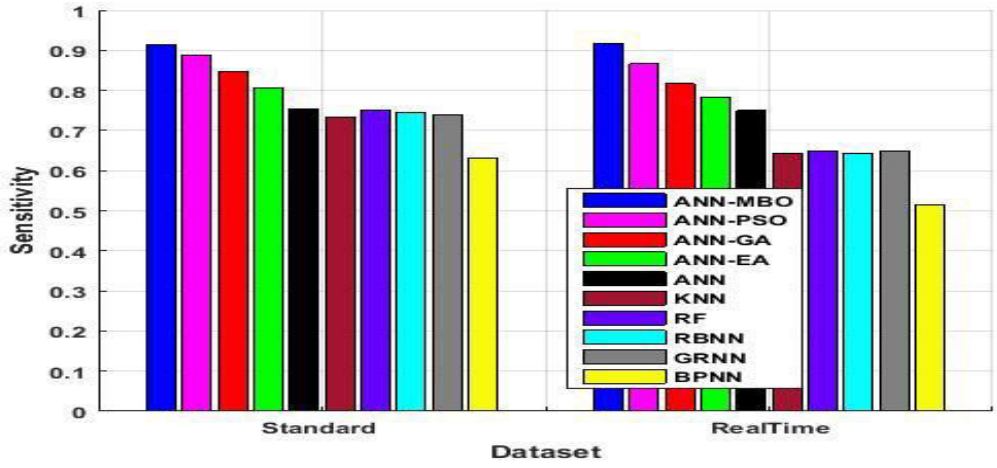

Figure-12. Performance evaluation for Sensitivity.

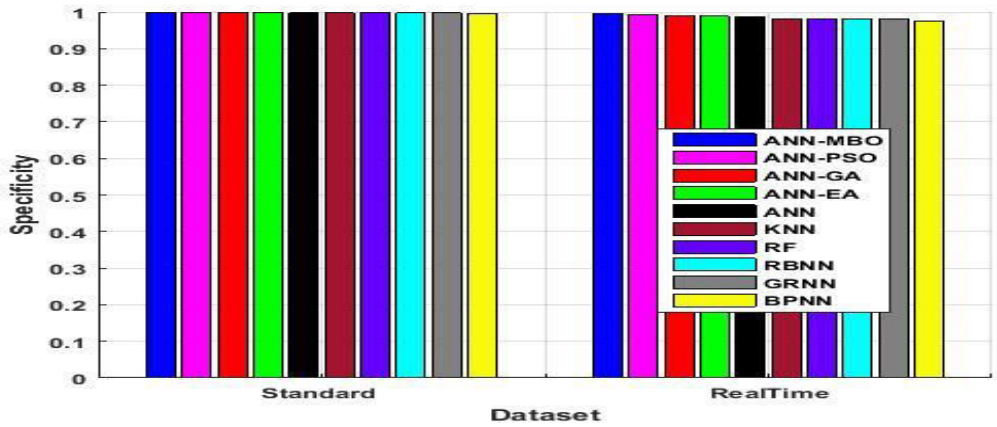

Figure-13. Performance evaluation for Specificity.

3.3 Performance of optimization techniques through convergence graph in the platform of ANN model configuration

The objective resolves in this investigation by incorporating optimization techniques in predicting appropriate number of hidden layers and their associated neurons essential to identify speaker recognition effectively. Over manual computation, incorporation of optimization techniques certainly reduce the computational time and process complexity in identify speaker recognition. The following convergence graph Figures 14 and 15 represent the performance of optimization associate AI techniques in two different database real time and standard respectively. Here, two sorts of optimization techniques employed in this execution namely evolutionary and swarm intelligence techniques; amid swarm intelligence techniques accomplished loftier performance over evolutionary techniques. In real time database, the performance of employed optimization associate AI techniques originates at the same position and slowly starts linking and fluctuates in their performance and eventually the employed optimization techniques gets saturated in 160^{th} iteration. In general, MBO accomplished superior performance over other techniques by predicting appropriate hidden layers and their associate neurons in the context of identifying speaker recognition. In standard database, like real time database the performance of employed optimization techniques originates at the same position and slowly starts linking and fluctuates in their performance. MBO, GA and EA get saturate at 150^{th} iteration and PSO get saturate at 180^{th} iteration. Here too, MBO utilize to configure ANN model accomplished loftier performance over contest optimization techniques.

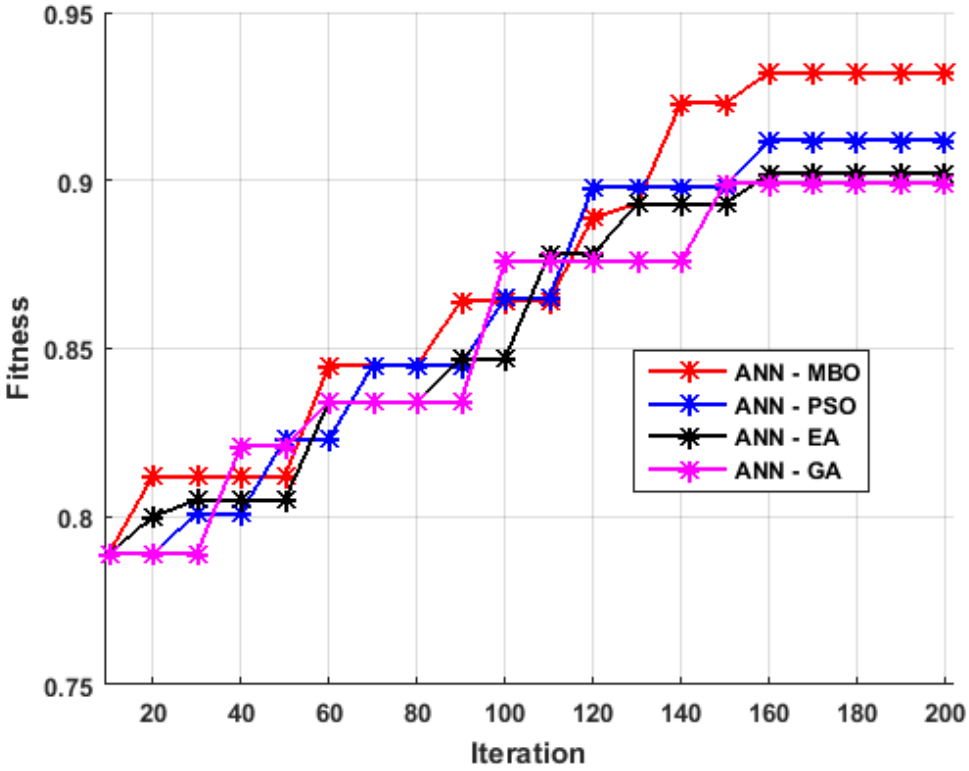

Figure-14. Convergence graph for real time database.

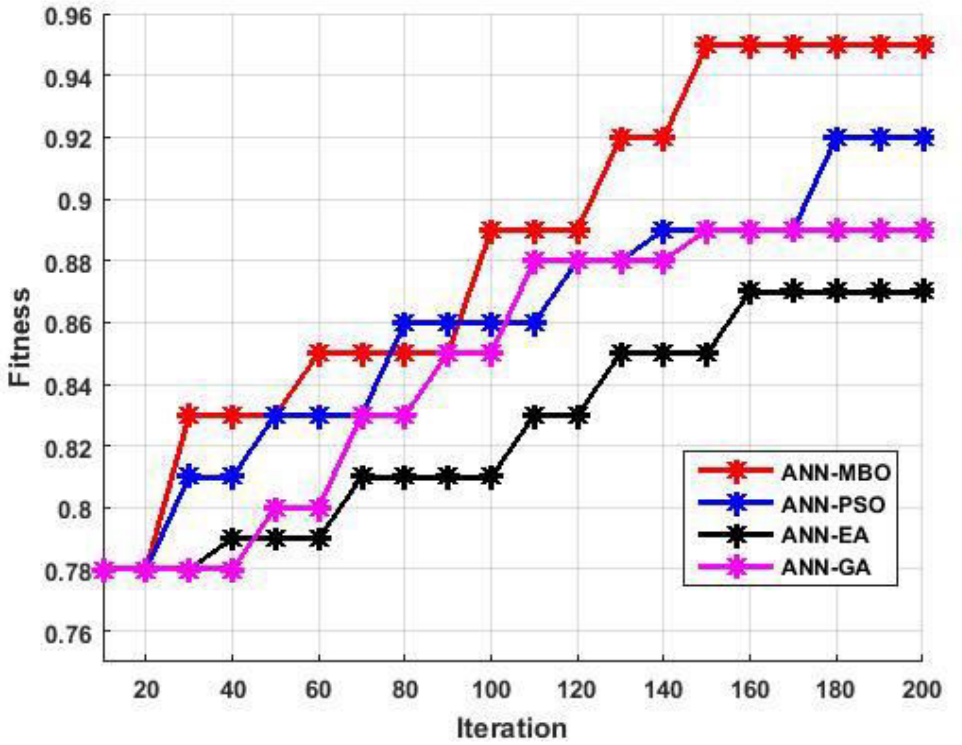

Figure-15. Convergence graph for standard database.

4. CONCLUSIONS

The purpose of incorporating MBO in configuring ANN structure in the context of recognizing speaker voice accomplished successfully. It is apparent that the strategy of incorporating optimization techniques is appropriate in improving the performance of

conventional ANN structure which certainly conserve time and process complexity over other comparative techniques like KNN, RF, RBNN, GRNN and BPNN. The performance of the employed techniques evaluate with nine different measures; amid MBO incorporate to configure ANN structure reveals 99% accuracy for real time database and 99.8% accuracy for standard database, which is superior over contest techniques. In contest with evolutionary and swarm intelligent techniques, it is quite evident that swarm intelligent techniques both PSO and MBO unveils superior results over GA and EA. This urge the future researcher to modify the objective function of MBO or to develop novel swarm based optimization technique to enhance the performance in recognition of speaker voice.

REFERENCES

[1] Ankur Maurya, Divya Kumar, R. K. Agarwal. 2018. Speaker Recognition for Hindi Speech Signal using MFCC-GMM Approach. Procedia Computer Science. 125, pp. 880-887.

[2] Sumithra Manimegalai Govindan, Prakash Duraisamy, Xiaohui Yuan. 2014. Adaptive wavelet shrinkage for noise robust speaker recognition. Digital Signal Processing. 33, pp. 180-190.

[3] Mansour Alsulaiman, Awais Mahmood, Ghulam Muhammad. 2017. Speaker recognition based on Arabic phonemes. Speech Communication. 86, pp. 42-51.

[4] Guy Hotson, Chi Zhu, Jonathan T. Ang, Anand Bhandari, Yuling Yan. 2012. Nyquist plot and quantitative characterizations of acoustic voice signals for speaker recognition. Proceedings of IEEE Conference on Industrial Electronics and Applications. pp. 1778-1783.

[5] Shahzadi Farah, Azra Shamim. 2013. Speaker Recognition System Using Mel-Frequency Cepstrum Coefficients, Linear Prediction Coding and Vector Quantization. Proceedings of International Conference on Computer, Control & Communication. pp. 1-5.

[6] Rohit Sinha, S. Shahnawazuddin. 2018. Assessment of pitch-adaptive front-end signal processing for children's speech recognition. Computer Speech and Language. 48, pp. 103-121.

[7] Oday Kamil Hamid. 2017. Speech Sound Coding Using Linear Predictive Coding (LPC). Journal of Information, Communication, and Intelligence Systems. 3, 1, pp. 13-17

[8] Raviraj Vishwambhar Darekar, Ashwinikumar Panjabrao Dhande. 2018. Emotion recognition from Marathi speech database using adaptive artificial neural network. Biologically Inspired Cognitive Architectures. pp. 1-8.

[9] Sabato Marco Siniscalchi, Torbjørn Svendsen, Chin-Hui Lee. 2014. An artificial neural network approach to automatic speech processing. Neurocomputing. 140, pp. 326-338

[10] Gülin Dede, Murat Hüsnu Sazlı. 2010. Speech recognition with artificial neural networks. Digital Signal Processing. 20, pp. 763-768.

[11] Jian-Da Wu, Yi-Jang Tsai. 2011. Speaker identification system using empirical mode decomposition and an artificial neural network. Expert Systems with Applications. 38, pp. 6112-6117.

[12] Saurabh Karsoliya. 2012. Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture. International Journal of Engineering Trends and Technology. 31, 6, pp. 714-717.

[13] Yinyin Liu, Janusz A. Starzyk, Zhen Zhu. 2007. Optimizing Number of Hidden Neurons in Neural Networks. Proceedings of the IASTED International Multi-Conference: artificial intelligence and applications. pp. 121-126.

[14] K. Gnana Sheela, S. N. Deepa. 2013. Review on Methods to Fix Number of Hidden Neurons in Neural Networks. Mathematical Problems in Engineering. pp. 1-12.

[15] Yanhong Feng, Gai-Ge Wang, Junyu Dong, Ling Wang. 2017. Opposition-based learning monarch butterfly optimization with Gaussian perturbation for large-scale 0-1 knapsack problem. Computers and Electrical Engineering. 000, pp. 1-15.

[16] Abhay Kumar, Sidhartha Sankar Rout, Varun Goel. 2017. Speech Mel Frequency Cepstral Coefficient feature classification using multi level support vector machine. Proceedings of International Conference on Electrical, Computer and Electronics (UPCON). pp. 134-138.

[17] Bright Kanisha, Ganesan Balarishnanan. 2016. Speech Recognition with Advanced Feature Extraction Methods Using Adaptive Particle Swarm

Optimization. International Journal of Intelligent Engineering and Systems. 9, 4, pp. 21-30.

[18] Gai-Ge Wang, Suash Deb, Zhihua Cui. 2015. Monarch butterfly optimization', Neural Computing and Applications. pp. 1-20.

[19] Khaled Daqrouq, Tarek A. Tutunji. 2015. Speaker identification using vowels features through a combined method of formants, wavelets, and neural network classifiers. Applied Soft Computing. 27, pp. 231-239.

[20] Seyed Reza Shahamiri, Siti Salwah Binti Salim. 2014. Artificial neural networks as speech recognisers for dysarthric speech: Identifying the best-performing set of MFCC parameters and studying a speaker-independent approach', Advanced Engineering Informatics. 28, 1, pp. 102-110.