INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 08 ISSUE: 04 | APRIL - 2024 SJIF RATING: 8.448 ISSN: 2582-3930

Speaker Recognition Using MFCC-BPNN-HHO

Dasari Harish (Y20EC036)
dept of Electronics and communication engineering
RVR & JC College Of Engineering

Gopu Karthika Chandrika (Y20EC061)
dept of Electronics and communication engineering
RVR & JC College Of Engineering

Guddanti Vasudha (Y20EC064)
dept of Electronics and communication engineering
RVR & JC College Of Engineering

Barigala Tarun (Y20EC013)
dept of Electronics and communication engineering
RVR & JC College Of Engineering

Abstract-Speaker recognition plays a pivotal role in speech processing. This paper proposes an enhancement to the Backpropagation Neural Network (BPNN) by incorporating Harris Hawks Optimization (HHO) for weight optimization, and evaluates its performance compared to the standalone BPNN. Both methods employ Mel Frequency Cepstral Coefficients (MFCC) for feature extraction from input data. The study assesses the proposed system on a dataset comprising 10 speakers, with each providing 10 utterances. Results demonstrate that the integrated MFCC-BPNN-HHO approach outperforms the standalone BPNN, achieving enhanced accuracy in speaker recognition tasks. Specifically, the accuracy of the BPNN-HHO was found to be significantly higher than that of the BPNN alone, indicating the effectiveness of the HHO optimization technique in improving speaker recognition accuracy. This study underscores the potential of integrating optimization algorithms like HHO with BPNN to further refine speaker recognition systems and contribute to advancements in speech processing technology.

This approach has promising applications in access control, identity verification and other security-related domains where biometric authentication is essential.

Keywords—MFCC, BPNN, HHO

I. INTRODUCTION

Speaker recognition is the process of identifying a speaker based on their voice characteristics. It has many applications, such as in security systems, forensic investigations, and personalized services.

In this project we will design Speaker Identification system using Mel-Frequency Cepstral Coefficients (MFCCs), Back Propagation Neural Network (BPNN) integrating with Harris Hawk Optimization (HHO) to improve the accuracy and to reduce the error rate of the Recognition system .The MFCCs of each speech sample are derived by pre-processing the speech signal. Then the features are classified towards the target speaker using Back Propagation Neural Network (BPNN) and Harris Hawk Optimization (HHO). The performance of proposed Speaker Recognition system MFCC-BPNN-HHO is evaluated using standard metrices like Accuracy, Precision, Recall, Sensitivity, Specificity. The integration of MFCC, BPNN and HHO yields a powerful Speaker recognition system that excels in accuracy.

II. DATASET

The dataset used in our experiments consists of 10 speakers, with each speaker providing 10 utterances, each of which is approximately 5-10 seconds long. total no of utterances are 100.Out of these 100 utterances,70% of the utterances are used for training the classifiers i.e,7 utterances from each speaker. And 30% of the utterances are used for testing i.e,3 utterances from each speakers.

III. METHODOLOGY

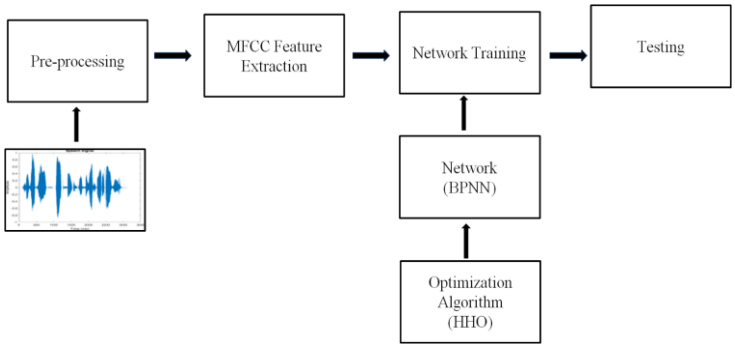

Fig 1:Block Diagram for Speaker Recognition System

Pre-Processing:

Preprocessing refers to the manipulation and transformation of raw data into a format that is more suitable for analysis, modelling, or other computational tasks.

Pre-emphasis:

The first step is to apply pre-emphasis to the signal to amplify higher frequencies. This is typically done using a first-order FIR filter:

$$y[n] = x[n] - \alpha . x[n-1]$$

where:

$x[n]$ is the input signal.

$y[n]$ is the output signal after pre-emphasis.

α is the pre-emphasis coefficient (around 0.95-0.97)

IV.FEATURE EXTRACTION

Mel-Frequency Cepstral Coefficients (MFCC) are a widely used feature extraction technique in speech and audio processing. They aim to capture the essential characteristics of the audio signal in a compact and efficient manner. MFCCs are particularly useful for speaker recognition tasks due to their ability to represent the unique vocal characteristics of individuals.

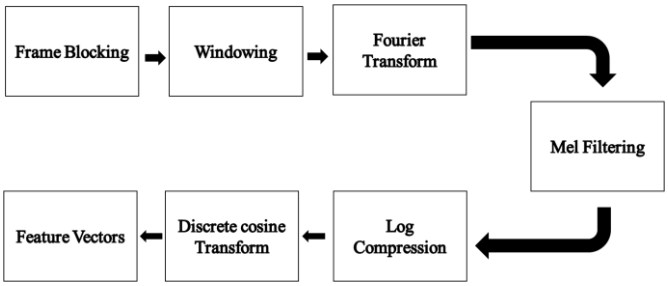

Fig 2:Block Diagram for MFCC Feature Extraction

1.Framing:

Next, the pre-emphasized signal is divided into short overlapping frames of typically 20-30 milliseconds. Common settings include 25 ms frame length with 10 ms overlap.

2.Windowing:

Each frame is windowed using a window function such as the Hamming window to reduce spectral leakage:

$$w[n] = 0.54 - 0.46cos\ (\frac{2\pi n}{N-1})$$

3.Fast Fourier Transform (FFT):

The windowed frames are passed through an FFT to convert the signal from the time domain to the frequency domain. This provides a spectral representation of the signal.

4.Mel Filter Bank:

The power spectrum of the signal is passed through a Mel filter bank to extract frequency bands that are spaced according to the Mel scale, which approximates the human auditory system's response to different frequencies. The filter bank typically consists of triangular filters.

$$m(f) = 2595log_{10}(1 + \frac{f}{700})$$

5.Logarithic Compression:

The log of the energy in each filter output is taken. This step helps in replicating the logarithmic perception of loudness by humans.

6.Discrete Cosine Transform (DCT):

Finally, Discrete Cosine Transform is applied to the logarithmically scaled filter bank outputs to decorrelate the features and obtain the final MFCCs.

7.Feature Vectors:

These steps result in a set of MFCC coefficients that capture the essential spectral characteristics of the input signal. Usually, only the lower-dimensional coefficients are kept (e.g., first 12-13 coefficients), as they contain most of the information about the spectral shape of the signal. these coefficients are commonly used as features for speech and audio processing tasks such as speech recognition, speaker identification, and emotion recognition.

V.NETWORK TRAINING

In the Network Training we use classifiers. Classifiers are algorithms that are used to map input features to output categories. Neural networks are a powerful class of classifiers that can learn complex relationships between the input features and output categories. In this paper Back Propagation neural network are used and for weight Updating, Harris Hawk Optimization are used.

5.1 Back Propagation Neural Network(BPNN)

Back propagation Neural Network (BPNN) is a type of artificial neural network that is trained using the backpropagation algorithm. It is a type of feedforward neural network, where the data flows in one direction from input to output layer through multiple hidden layers.

The backpropagation algorithm is a supervised learning algorithm, which means that it requires labelled data for training. The algorithm is used to adjust the weights and biases of the connections between the neurons in the network in order to minimize the error between the predicted output and the actual output. The activation function used in BPNN is Sigmoid activation function.

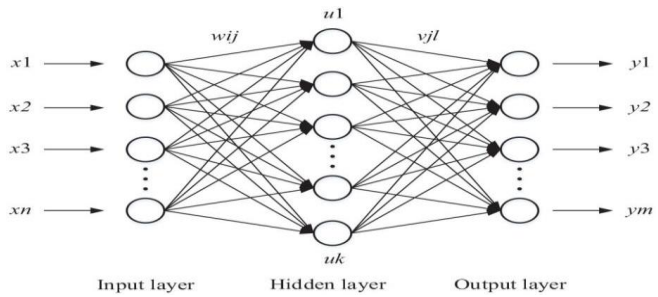

Fig.3: BPNN architecture

Input Layer: This layer consists of neurons that represent the input features. Each neuron corresponds to one feature of the input data.

Hidden Layers: These are intermediate layers between the input and output layers. Each neuron in a hidden layer is connected to every neuron in the previous layer and every neuron in the subsequent layer. The number of hidden layers and neurons in each layer is configurable and depends on the complexity of the problem.

Activation function: An activation function is a key component of artificial neural networks, serving as a non-linear transformation applied to the output of a neuron or a layer of neurons.

Output Layer: This layer produces the network's output. The number of neurons in the output layer depends on the type of problem being solved. For example, in a binary classification problem, there would be one neuron for each class, while in a regression problem, there would be a single neuron.

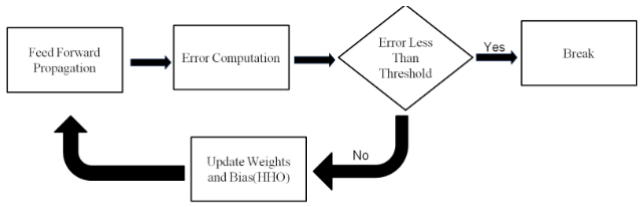

Fig.4: Proposed Model Algorithm

1.Initialization:
Initialize the weights and biases of the network randomly. Let us denote the weights between input layer and hidden layer as *Win*, where *i* represents the input neuron and *j* represents the hidden neuron. Similarly, denote the weights between hidden layer and output layer as *Vjk* , where *j* represents the hidden neuron and *k* represents the output neuron. Also, initialize biases *bj* and *ck* for the hidden and output layers, respectively.

2.Forward Pass:
For each input sample *x*, compute the activations of the hidden layer neurons

$$h_j = \sigma(\sum_{i=1}^{n} W_{ij}x_i + b_j)$$

Where σ is the activation function for the hidden layer.
Similarly, compute the activations of the output layer neurons

$$y_k = \sigma(\sum_{j=1}^{m} V_{jk}h_j + c_k)$$

Where σ is the activation function for the output layer.

3.Calculate Loss:
Compute the loss between the predicted output *yk* and the actual output *tk* using a suitable loss function such as Mean Squared Error (MSE):

$$E = \frac{1}{2}\sum_{k=1}^{p}(t_k - y_k)^2$$

4.Backward Pass:
Compute the gradient of the loss function with respect to the output layer activations:

$$\frac{\partial E}{\partial y_k} = y_k - t_k$$

Update the weights and biases between the hidden layer and the output layer using the gradient descent algorithm:

$$V_{jk} \leftarrow V_{jk} - \eta\frac{\partial E}{\partial V_{jk}}$$

$$c_k \leftarrow c_k - \eta\frac{\partial E}{\partial c_k}$$

Where η is the learning rate.
Similarly, compute the gradient of the loss function with respect to the hidden layer activations:

$$\frac{\partial E}{\partial h_j} = \sum_{k=1}^{p}\frac{\partial E}{\partial y_k}\cdot V_{jk}\cdot\sigma'(h_j)$$

Where σ' is the derivative of the activation function for the hidden layer.
Update the weights and biases between the input layer and the hidden layer:

$$W_{ij} \leftarrow W_{ij} - \eta\frac{\partial E}{\partial W_{ij}}$$

$$b_j \leftarrow b_j - \eta\frac{\partial E}{\partial b_j}$$

5.Repeat:
Repeat steps 2-4 for a fixed number of iterations or until convergence.

5.2 Harris hawk's optimization (HHO)
Harris Hawk Optimization (HHO) is a nature-inspired optimization algorithm developed based on the hunting behaviour of Harris's Hawks, a species of bird of prey. The key idea behind HHO is to mimic the collaborative hunting behaviour observed in Harris's Hawks. In the wild, these hawks engage in cooperative hunting strategies, where they work together to catch prey more efficiently. This collaboration involves various roles such as searching, chasing, and capturing prey.
In the context of optimization, HHO translates this collaborative behaviour into an iterative search process aimed at finding optimal solutions to optimization problems. Since BPNN is more sensitive to initial weights and thresholds, this research proposes the Harris hawk's optimization method for BPNN random weights and threshold optimization. The whole optimization process includes an exploration phase and exploitation phase.

HHO Algorithm:

1.Initialization: Start by initializing a population of Harris hawks. Each hawk represents a potential solution to the optimization problem. Initialize their positions randomly within the search space.

2.Evaluation: Evaluate the fitness of each hawk based on its position. This involves calculating the objective function value corresponding to each hawk's position. The objective function represents the problem being optimized.

3. Exploration Phase: During this phase, the hawks explore the search space individually, akin to scouting for prey. Each hawk updates its position based on its current position and velocity.

$$X(t+1)=\begin{cases} X_{rand}(t) - r_1 \cdot \left|X_{rand}(t) - 2 \cdot r_2 \cdot X(t)\right|, q \geq 0.5 \\ \left(X_{rabbit}(t) - X_b(t)\right) - r_3 \cdot \left(lb + r_4 \cdot (ub - lb)\right), q < 0.5 \end{cases}.$$

Where

$X(t+1)$ - Position Vector

$X(t)$ - current position

$X_{rabbit}(t)$ -Target Position

$X_b(t)$ - Average position of current Hawks population

r1,r2,r3,r4 and q → random numbers[0,1]

The upper and lower bounds of the variables are represented by ub and lb, respectively, while Xrand(t) is the hawk chosen at random from the population, which is updated in each iteration.

To Compute Average position of current Hawks population

$$X_b(t)=\frac{1}{N}\sum_{i=1}^{N} X_i(t)$$

$X_i(t)$ - Hawk current position

4. Exploitation Phase: In this phase, the hawks collaborate to exploit promising areas of the search space. This collaboration involves sharing information about the best solutions found so far and adjusting their positions accordingly.

$$E = 2E_0\left(1-\frac{t}{T}\right)$$

Where

E=Escaping Energy of the pray

E0=Initial state of it's energy inside the interval [-1,1]

i. E_0 decreases from 0 to -1 (rabbit is tired)

ii. E_0 increases from 0 to1 (rabbit is strengthening)

5.Update: Update the positions of the hawks based on their exploration and exploitation efforts. The update mechanism is typically guided by mathematical equations inspired by the hunting behavior of Harris's Hawks.

6.Termination: Repeat the exploration and exploitation phases until a termination criterion is met (e.g., a maximum number of iterations or a satisfactory solution is found).

In Harris Hawks Optimization (HHO), the concepts of soft besiege, hard besiege, soft besiege with progressive rapid dives, and hard besiege with progressive rapid dives represent different exploration and exploitation strategies inspired by the hunting behaviour of Harris hawks.

i. Soft Besiege:

$$r \geq 0.5 \,\&\, |E| \geq 0.5$$

In this case, the rabbit still has enough energy, and the hawk aim to exhaust its energy before launching the assault.

$$X(t+1) = \Delta X(t) - E\left|JX_{rabbit}(t) - X(t)\right|,$$

$$\Delta X(t) = X_{rabbit}(t) - X(t).$$

The above Equation expresses how the model updates the rabbit's position

Where,

$$J = 2(1-r_5)$$

r5-random number[0,1]

ii.Hard Besiege:

$$r \geq 0.5 \,\&\, |E| < 0.5$$

In this case , the Rabbit is tired and has low escaping energy.so Harris Hawk perform surprise attack.

$$X(t+1) = X_{rabbit}(t) - E\left|\Delta X(t)\right|.$$

The above Equation expresses how the model updates the rabbit's position

iii.Soft Besiege with Progressive Rapid Dives:

$$r < 0.5 \,\&\, |E| \geq 0.5$$

In this case,Rabbit has enough escaping energy and try to escape by many zigzag motions.so Harris Hawk perform levy-based short rapid dives around the rapid.

$$X(t+1) = \begin{cases} X_1, if\ fobj(x_1) < fobj(X(t) \\ X_2, if\ fobj(x2) < F(X(t) \end{cases}$$

$X_i = Xrabbit(t) - E|JXrabbit(t) - x(t)|$

X2=X1+r6(1,dim).*levy(dim)

The above Equation expresses how the model updates the rabbit's position

iv.Hard Besiege with Progressive Rapid Dives:

$$r < 0.5 \& |E| < 0.5$$

In this case,Rabbit has less escaping energy and try to escape by many zigzag motions.so Harris Hawk perform levy-based short rapid dives around the rapid.

$$X(t+1) = \begin{cases} X_1, if\ fobj(x_1) < fobj(X(t) \\ X_2, if\ fobj(x2) < F(X(t) \end{cases}$$

$$X_1 = Xrabbit(t) - E|JXrabbit(t) - xm(t)|$$

X2=X1+r7(1,dim).*levy(dim)

The above Equation expresses how the model updates the rabbit's position

VI. RESULTS AND DISCUSSIONS

The Following Figure is Confusion Matrix.
A confusion matrix is a table that is used to evaluate the performance of a classification model by comparing the predicted and actual class labels of a set of test data. The matrix shows the number of true positives, true negatives, false positives, and false negatives, arranged in a tabular form. In a binary classification problem, a confusion matrix typically has two rows and two columns. The rows represent the predicted class labels (positive or negative), while the columns represent the actual class labels.

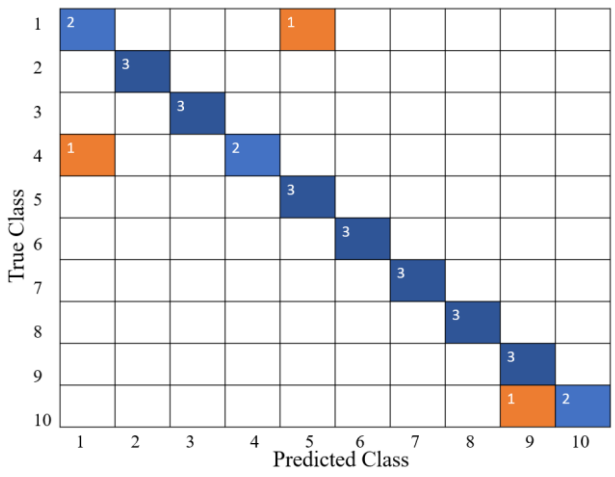

Fig.5: Confusion Matrix of MFCC-BPNN-HHO

These fig-5 is useful for calculating the True Positives, True Negatives, False Positives, and False Negatives. These Parameters are used for calculating the accuracy of the network.

TP=27 FP=3 FN=3 TN=267

Performance Metrices :

Method	Formula	MFCC-BPNN-HHO
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	98.00%
Precision	$\frac{TP}{TP+FP}$	90.00%
Recall	$\frac{TP}{TP+FN}$	90.00%
Specificity	$\frac{TN}{TN+FP}$	98.89%
Sensitivity	$\frac{TP}{TP+FN}$	90.00%
F1 Score	$\frac{2*precision*recall}{precision+recall}$	90.00%

Table1:Results of MFCC-BPNN-HHO

Method	Accuracy	Error Rate
MFCC-KNN	93.40	0.33
MFCC-RF	92.80	0.36
MFCC-SVM	92.60	0.37
MFCC-BPNN	96.00	0.20
MFCC-BPNN-HHO	98.00	0.10

Table 2: Comparison of Accuracy and Error Rate for different methods

Comparison of Accuracy:

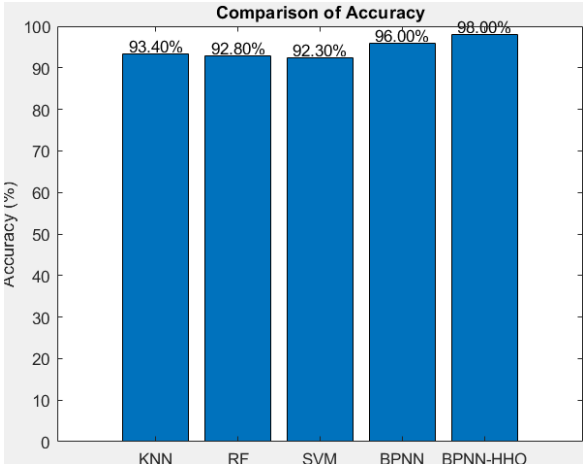

Fig.6: comparison of Accuracy Bar graph

VII.CONCLUSION

Based on the results obtained in this study, it can be concluded MFCC-BPNN-HHO is effective algorithm for Speaker recognition. The MFCC-BPNN-HHO algorithm outperformed the MFCC-BPNN algorithms in terms of accuracy, achieving a classification accuracy of 98.00%, compared to the 96.00% achieved by MFCC-BPNN. The confusion matrix analysis provided insight into the performance of the algorithms in each class. Both algorithms achieved high accuracy in classifying the majority of the Speakers, with MFCC-BPNN-HHO performing better than MFCC-BPNN in most cases. Furthermore, the use of Mel Frequency Cepstral Coefficients (MFCC) for dimensionality reduction proved to be effective in improving the accuracy of both algorithms.

REFERENCES

[1] Z. Liu and H. Chen, "An improved Harris Hawk optimization algorithm and its application to Extreme Learning Machine," 2023 3rd International Conference on Consumer Electronics and Computer Engineering (ICCECE), Guangzhou, China, 2023, pp. 842-846, doi: 10.1109/ICCECE58074.2023.10135354.

[2] N K Kaphungkuni,Adithya Bihar Kandali ”Text Dependent Speaker Recognition With Back Propagation Neural Network”(ISSN-2249-8958)

[3] A.Jose Albin, N.M.Nandhitha,S.Emalda Roslin ”Text Independent Speaker Recognition System Using Back Propagation Neural with Wavelet Features” (IEEE-2014)

[4] Mr.P,Kumar,Dr.S.L.Lahudkar ”Automatic Speaker Recognition using LPCC and MFCC”(ISSN:2321-8169)

[5] Jorge MARTINEZ*,Hector PEREZ,Enrique ESCAMILLA,Masahisa Mabo SUZUKI ”Speaker Recognition Using Mel- Frequency Cepstral Coefficients(MFCC) and Vector Quantization (VQ) Techniques”(IEEE2012)

[6] C.Sunitha,E.Chandra ”Speaker Recognition Using MFCC and Improved Weighted Vector Quantization Algorithm” International journal of Engineering and Technology(IJET)(ISSN:0975-4024)

[7] V. Wan and W. M. Campbell, "Support vector machines for speaker verification and identification," Neural Networks for Signal Processing X. Proceedings of the 2000 IEEE Signal Processing Society Workshop (Cat. No.00TH8501), Sydney, NSW, Australia, 2000, pp. 775-784 vol.2, doi: 10.1109/NNSP.2000.890157.

[8] S. Raghavan, G. Lazarou and J. Picone, "Speaker Verification using Support Vector Machines," Proceedings of the IEEE SoutheastCon 2006, Memphis, TN, USA, 2006, pp. 188-191, doi: 10.1109/second.2006.1629347.