

# Raga classification using enhanced spatial bound whale optimization algorithm

Bettadamadahally Shivakumaraswamy Gowrishankar<sup>1</sup>, Nagappa U. Bhajantri<sup>2</sup>

<sup>1</sup>Department of Information Science, Vidyavardhaka College of Engineering, Karnataka, India

<sup>2</sup>Department of Computer Science, Government College of Engineering, Karnataka, India

## Article Info

### Article history:

Received Oct 27, 2022

Revised Jan 18, 2023

Accepted Jan 26, 2023

### Keywords:

Audio feature extractor  
Enhanced spatial bound whale optimization algorithm  
Mel frequency cepstrum coefficients  
Short time energy  
Spectral centroid features  
spectral flux

## ABSTRACT

A raga is a unique set of notes with certain rules that carefully followed, retain and protect its purity and produce amazing musical effects. An automated raga transcription and identification is important for computational musicology, which is an important step for musicology for indexing, classifying, and recommending tunes. In the present research, the audio features such as mel frequency cepstrum coefficients (MFCCs), spectral flux, short time energy, audio feature extractor, and spectral centroid features are used for the prediction of a raga. The model showed more complexity which means it required lots of training data. The proposed enhanced spatial bound whale optimization algorithm (ESBWOA) is used that overcome the feature selection problem of high dimensional features. In addition to this, a weighted salp swarm algorithm (SSA) is used for selecting the tone-based features from the ragas based on amplitude or each raga sample. The features were fed for bidirectional long short-term memory (Bi-LSTM) network, which enhanced the success rate for raga identification and classification. The present research uses CompMusic dataset in the research work where 9 classes for Carnatic music and 7 classes in Hindustani music are considered for the classification of ragas.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Bettadamadahally Shivakumaraswamy Gowrishankar  
Department of Information Science, Vidyavardhaka College of Engineering  
Mysore, Karnataka, India  
Email: gowrish.vvce@gmail.com

## 1. INTRODUCTION

Indian classical music (ICM) based raga classification is important for music information retrieval. Various existing studies have shown that machine learning techniques are conceivable because it has been considered a lot of musical data to process through the internet [1], [2]. Significant work has been performed on the multimedia sources like text or video also audio processing in the developing phase. The ICM is categorized broadly into Hindustani and Carnatic music [3], [4]. It is having a wide range of following in the way Carnatic music has higher complexity which means the notes were arranged and rendered. ICM is based on the Talam and Raga which are considered to be equivalent for melody in music [5]. Thus, an automated music transcription is performed for the Carnatic music which would be challenging due to the variations happening in the swara note frequencies. Thus, the variations help to detect raga and transcription of music is difficult [6].

The deep learning model like LSTM based model was used for performing automated transcription for the Carnatic music. The detection of the note is solved with the problem of image classification [7], [8]. The ragas classification showed complexity that is more compared with western music in the context of scale and melody. The ragas notes are arranged sequentially which invokes the emotion of a song [9]. Thus, each

note is set with the frequency that is associated with them. Carnatic music is associated with the Talam and it is a beat belonging to Western music [10]. The placement of the syllables and the music temp are signified by the composition of the music. Due to the complexity of a note, the existing LSTMs were sensitive to different random weight initializations that showed the problem of overfitting [11], [12]. Therefore, hyper parameter optimization and the problem of tuning are performed by setting the optimal hyper parameters during the learning of an algorithm. The hyper parameter is having values that are controlled by the process of learning [13]. By contrast, the values of other parameters' typical node weights are learned [14]. The proposed technique has built a model that has a possible combination with overall hyper parameters that are provided also the model evaluated and selected the architecture for producing better results [15]. Kaur and Kumar [16] developed a fuzzy-based hierarchy-based pattern matching model for classification based on melody. The improved pattern matching technique was used for the classification by using fuzzy analytical hierarchy process. The developed model conducted results on the datasets that were having a wide range of melodies belonging to a classical background which showed an improvement in classification rate.

The developed model needed to discover the pattern further which was integrated with fuzzy probabilistic models. Kiran [17] developed Dragonfly algorithm applied to the neural network for performing the ICM classification. The developed model utilized an effective raga recognition model in Carnatic music genre for classification that identified the model effectively. The neural network was used as an adaptive classifier that exploited the learning feature set that was used for classification. The results obtained by the developed model analyzed the results precisely which needed superiority for raga identification. Sharma *et al.* [18] developed a deep learning model to perform the ICM based classification based on the time series matching. Various deep learning algorithms were implemented such as recurrent neural network (RNN)-LSTM, support vector machine (SVM), deep neural network (DNN) classifiers were implemented. The features such as scalogram, spectrogram, mel frequency cepstrum coefficient (MFCC), were fed for the visual geometry group (VGG)-16, convolutional neural network (CNN) (1st layer, 2nd layer, 3rd layer), ResNet-50 layers. The layered RNN-LSTM and CNN obtained the best values compared to other approaches. The model needed to be required to focus on tempo and melody range shape to cover the wide range of Indian concerts. John *et al.* [19] developed an automated raga classification to process the audio signals and the recognition of the raga is automatically performed by using a deep learning model. The music signals were synthesized which managed the audio dataset for music therapy. The developed model has used the pitch contour to select the raga from the Carnatic music as it extracted the key features that are applied on CNN. The developed model showed a challenge in pattern recognition as the CNN failed to understand the sequences using Parsel-mouth library.

Sarkar *et al.* [20] utilized Hindustani and Carnatic classical music for raga identification from the audio signals for identifying the properties. The existing models performed an automated raga recognition that consumed time and overcame the problem. The co-occurrence matrix was used for overcoming the problem and the audio clip features extracted identified the properties of raga features. The features that were obtained were fed for the support vector machine the type of raga to which it belonged. However, the error that occurred in the classification required manual identification. Shah *et al.* [21] developed the raga deep learning model for recognition in ICM. The developed deep learning and signal processing based approach was presented in the research work that recognized the raga based on the audio spectrograms. The developed model decreased the performance when faster temp parts were used. The separation of sounds affected the model accuracy. The source was separated from the audio which showed better results for the original audio signals. The spectrograms worked well in predicting the ragas with the preprocessing steps and entirely were not correct at the point. The developed model required more investigation because deep learning models were just at one end. However, the separation of the unit was not developed for ICM instruments in that particular area. Krishnaiah and Divakarachari [22] developed hybrid spectral features for the automated classification of mood using Multi-SVM (MSVM). The identification and extraction of features were required to be considered which was a major issue in the existing models.

Thus, the problem was overcome by using the hybrid spectral feature extraction model that includes the combination of spectral features from audio including spread, skewness, centroid, MFCCs, and linear prediction coefficients (LPCs). These features were required to reduce the complexity that enhanced the success rate of MSVM for performing the classification of moods. The distinct types of moods were required to be used for the classification. Therefore, in the present research work, the audio features such as MFCCs, spectral flux, short time energy, audio feature extractor, and spectral centroid features are used that accurately predict the raga but also make the complex model. The proposed enhanced spatial bound whale optimization algorithm (ESBWOA) is used to overcome the problem of high dimensional features as the feature selection algorithm selects and computes the features space without losing the feature properties. The bidirectional long short-term memory (Bi-LSTM) classifier for classification enhanced the success rate that classifying the model into 9 classes for Carnatic music and 7 classes in Hindustani music. Where Carnatic music includes bagesri, bhairavi, nata, kalyani, madhyamvati, sindhubhairavi, yamankalyani, purvikalyani. Whereas, Hindustani music

includes bagesri, bhairavi, hamsadhvani, madhyamavati, mulkauns, todi, and yamankalyani. The contributions of the proposed research work are: (i) Developed an ESBWOA for overcoming the problem of feature dimensions and (ii) proposed a weighted salp swarm algorithm (SSA) for the selection of features that are fed for the Bi-LSTM classifier to perform the classification of Carnatic and Hindustani music. The research paper is organized as follows: section 2 discusses the enhanced feature selection algorithm and section 3 describes the results and discussion of the proposed research evaluated qualitatively and quantitatively. The conclusion and future work for the proposed research work is given in section 4.

## 2. MATERIAL AND METHOD

The block diagram of the research work is shown in Figure 1 which consists of compmusic dataset, feature extraction process that includes MFCC, spectral flux spectral centroid, short-time energy, audio feature extractor. The process of feature selection is performed by using ESBWOA and hyper parameter optimization uses weighted SSA. The features that were obtained were undergone for the process of classification using Bi-LSTM.

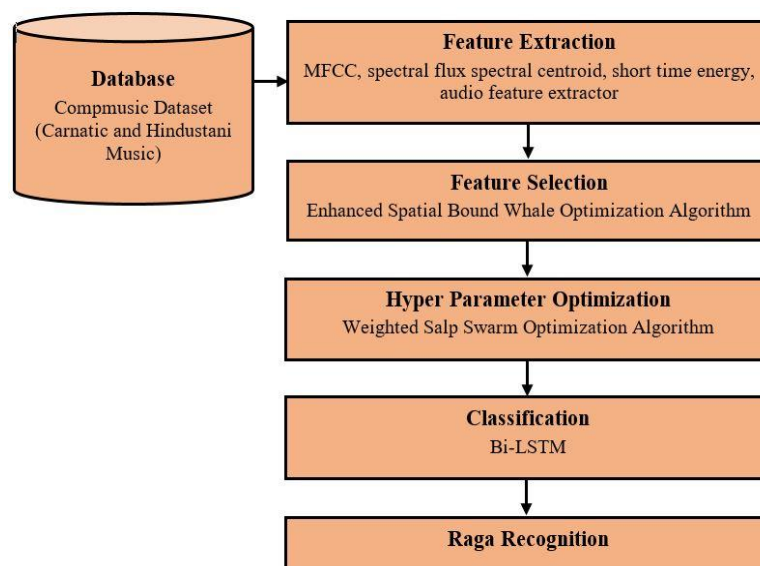


Figure 1. Block diagram of the proposed research

### 2.1. Dataset

The Comp music dataset has both Carnatic and Hindustani music datasets that include 124 hours which are 480 recordings present in the mp3 format. Each raga recording is having a full length that is present with the dataset consisting of 40 distinct ragas. The diverse ragas are distinct and their presence constitutes numerous compositions. The Hindustani music consists of a total of 116 hours of audio recordings and among those 300 recordings are consisting of 30 types of ragas that has 10 distinct record types for the single raga. The diversity refers to the attributes of melody and thus training evaluation for the model is performed using the dataset showed better results [23]. The data obtained undergoes audio transcription, which performed feature extraction using techniques such as spectral flux, spectral centroid, MFCCs, short-time energy, and audio feature extractor features are extracted.

### 2.2. Feature extraction

#### 2.2.1. Spectral flux and spectral centroid

Spectral flux (SF) is a measure of the variability of the spectrum over time. SF measures the power spectrum that calculates the signal for changing and comparing with one of the frames that are against the previous frame of the power spectrum. SF is expressed as shown in (1). Spectral centroid refers to the center of gravity that rates for frequently occurring parameters and determines the timber parameters. The spectral centroid is defined in each range of frequency energy belonging to the spectrum. The spectral centroid is ranging with the frequency energy components as they are dependent on each other. The spectral centroid is expressed as shown in (2).

$$SF = \frac{1}{KF} \sum_{i=2}^K \sum_{j=1}^F (\varepsilon_{i,j} - \varepsilon_{i-1,j})^2 \quad (1)$$

$$f_c = \frac{\sum_k S(k)f(k)}{\sum_k S(k)} \quad (2)$$

From the (1),  $\varepsilon_{i,j}$  is known as the spectral energy at the sub window  $i$  to  $K$  and the channel frequency is  $j$  to  $F$ . Where,  $f_c$  refers to the Centre Frequency,  $S(k)$  refers to the weighted mean having the bin ' $k$ ', and  $f(k)$  is known as the weighted frequency which are consisting of bin ' $k$ '. In (2), shows the centroid function which are belonging to the highest function. The normalized values are having the center which reduces the impact of the gamma in toning the audio features. The feature extracted indicates the loudness of the audio sample that has resulted in the spectral center having gravity. The centroid calculation is evaluated for normalizing and balancing the signal throughout to equalize throughout. Thus, the feature is used for balancing the high spectral and low spectral centroid.

### 2.2.2. Mel frequency cepstrum coefficients and short time energy

MFCC is called the perceived components as the pitch consists of frequency components that calculate the rate of frequency. The changes in the lower level of ragas showed small changes when the audio features are insensitive to the ears. In MFCCs, the Mel Scale can understand the frequency or pitch for detecting the pure tone which can measure the actual frequency of the components. The small changes in the pitches are important for considering in the present research. The MFCC is used for processing speech and music as it is having the ability for modelling the subjective frequency that has also audio signal contents. The expression for an approximate Mel is expressed as shown in (3). The audio signal is time varying in nature and the energy is associated with it. Automatic processing of audio is the interest for automatic processing which gets to know the varied energy with respect to time and specific energy associated. The present energy components are voiced with the region that has shown a larger region when compared to the unvoiced region. The model has the silence region showed the least negligible energy. The total energy for the short time energy (SE) is expressed as shown in (4). In (4),  $S$  is known as the frame's length of audio. The short-term energy computation is considered for the vocal in terms of 60 sec.

$$f_{mel} = 2595 \log_{10} \left( 1 + \frac{f}{700} \right) \quad (3)$$

$$SE = \sum_{i=1}^S x_i^2 \quad (4)$$

### 2.2.3. Audio feature extraction, sample rate, and hamming window

The audio feature extraction is important for audio signal processing as it is a subfield in signal processing. The manipulation of audio signals is processed as it processes the signals. The unwanted noises are removed as they can balance the range of time-frequency to convert the digital signals into analog signals. The number of audio samples presents consisted of acoustic wave values with respect to the time specified. The audio sample is numbered which represented the acoustic wave having a specific point as a value. The sample rate is defined as shown in (5). The hamming window can form a taper by using a raised cosine having non-zero endpoints that are optimized for minimizing the nearest side lobe. The hamming window is represented using (6).

$$f_s = \frac{1}{T} \quad (5)$$

$$w(n) = 0.54 - 0.46 \cos \frac{2\pi n}{N-1} \quad (6)$$

Where,  $f_s$  is known as the samples obtained,  $\frac{1}{T}$  is known as the sample per time,  $0 \leq n \leq M - 1$ ,  $M = \frac{N}{2}$  for  $N$  even and  $\frac{N+1}{2}$  for  $N$  odd.

### 2.2.4. Overlap length, spectral flatness, and spectral kurtosis

The overlap length is performed among the hop length and the window length which evaluates the difference. The overlap length is represented as shown in (7). The spectral flatness is measured typically in terms of dB which provide a way to quantify the sound resemblance for a pure tone that is opposed to noise-like features. The spectral flatness is expressed using the following (8). Spectral kurtosis (SK) is a tool that

indicates the presence of transient series and their location with respect to the frequency domain. The classical power spectral density completely removes the non-stationary information. SK is expressed using the following (9). In the (9),  $f_k$  is known as the frequency in Hz that is corresponded to bin  $k$ ,  $b_1$  and  $b_2$  are the band edges, in bins that are calculating the spectral skewness,  $\mu_1$  is the spectral centroid and  $\mu_2$  is the spectral spread.

$$\text{Overlap length} = \text{Window length} - \text{Hop length} \tag{7}$$

$$\text{Spectral flatness} = (\prod_{i=1}^N Z[i])^{\frac{1}{N}} (\sum_{i=1}^N Z[i])^{-1} \tag{8}$$

$$SK = \frac{\sum_{k=b_1}^{b_2} (f_k - \mu_1)^4 s_k}{(\mu_2)^4 \sum_{k=b_1}^{b_2} s_k} \tag{9}$$

**2.3. Feature selection using enhanced spatial bound whale optimization**

The WOA is used for tackling the global optimization problem and the WOA is having the ability for tackling optimization problems. The WOA has advantages of premature convergence and local optima avoidance for overcoming the problem and it is important to update the whales. The update in the positions is performed based on the current whale location as best. The degradation of WOA improves the performance as it can employ the problem of high dimensional feature selection problem as it is a combined optimization approach [24], [25]. The complex combinatorial problems consist of an extensive number of optimums that can increase the rate of whales trapped based on the local solutions. The local optima avoidance is weak and the premature convergence provided is the main factor that caused degradation in the WOA performance. Thus, distinct strategies were included in the WOA for improving high-dimensional performances. The ESBWOA is used for improving the efficiency of the conventional WOA to overcome the Feature Selection problem of high dimensional features. The present research work uses a spatial bound mechanism for regulating the dimensions for each solution in a population. The Chaos map is used firstly to initialize the population. The initial population is significantly improved as it is having diversity in the algorithm that solves the optimization problems through WOA. The cubic mapping is used in the research because the values generated by cubic mapping are uniform. The cubic mapping is better and the cubic mapping expression is given by using (10).

$$\text{cubic mapping} = \begin{cases} y(n + 1) = 4 * y(n)^3 * y(n) \\ -1 \leq y(n) \leq 1, n = 0, 1, 2, \dots \end{cases} \tag{10}$$

Where  $y(n)$  in the sequence generated by the cube map is between -1 and 1,  $i$  is the initial iteration value of the cube starting with 0. The iteration value is initialized with a cube which is not getting the value as zero that would have shown Chaos as an occurrence. There is  $N$  number of particles that are in  $D$ -dimensional space which is generating the vectors randomly with the first particle. Each of the values of the components are ranging from (-1 to 1). Therefore, each dimension vector is replaced with the expression present in (11) working with  $N - 1$  number of iterations that resulted in  $N - 1$  number of particles. Yet, the value of  $y(n)$  in the sequence is generated with a cube map between -1 and 1. Thus, it is mapped to search for the particles present in the interval. The rules for the mapping are shown in (11).

$$x_{id} = min_d + (1 + y_{id}) \frac{(max_d - min_d)}{2} \tag{11}$$

Where  $i = 1, 2, \dots, N$ ;  $d = 1, 2, \dots, D$ .

where  $max_d$  and  $min_d$  is representing the upper and the lower limits with  $d^{th}$  dimensions in the search space.  $y_{id}$  is known as the coordinate of the  $i^{th}$  a particle which is generated using (11) and the  $i^{th}$  particle is coordinating with the  $d^{th}$  dimension in the search space. Once after the initialization of the population, the global search and the local exploitation showed non-linear convergence factors to be used. The process of optimization used for the WOA is known as the updated and complicated strategy to get the convergence factor if it was not reflecting the actual situation. Thus, the model has introduced the non-linear convergence factor where the formula has been expressed as shown in the following (12).

$$a_t = 2 - (a_{initial} - a_{end}) * \left[ \frac{\left( e^{\frac{t}{Max_{iter} - 1}} - 1 \right)}{e - 1} \right]^U \tag{12}$$

where  $t$  is known as the current number of iterations,  $Max_{iter}$  represents the maximum number of iterations,  $a_{initial}$  and  $a_{end}$  is known as the termination and initial values of  $a$  control parameters,  $u$  is known as the non-linear adjustment coefficient.  $y$  which enhances the capability of algorithm exploitation with accuracy and convergence speed. The feature numbers are selected randomly for obtaining a solution of two parameters such as maximum dimension rate and minimum dimension rate ( $d_{min}, d_{max}$ ). The selection of the number of features is larger or fewer than the minimum or maximum dimension. The minimum dimension is equalized to the value of 2 and thus additional features are chosen randomly from the feature vector that obtains the solution. The additional features are removed randomly to obtain the solution as the maximum dimension is determined at 4. A spatial pool is used for creating and storing the  $d_{min}$  and  $d_{max}$  value. The spatial pool is created for storing the  $d_{min}$  and  $d_{max}$  values. Thus, distinct numbers such as  $d_{max}$  and  $d_{min}$  numbers are expressed and stored with the spatial pool. The equation for the  $d_{min}$  is defined as shown in (13).

$$d_{min_j} = \frac{d_{max_j}}{2} \quad (13)$$

Where  $j = 1, 2, \dots, N_p$ , where  $N_p$  is the total number of  $d_{max}$  in pool. The (13) is applied for the spatial pool as constructed.

### 2.3.1. Spatial value, spatial bounding process and spatial update

The spatial value has assigned initially at each set that are consisting of distinct rates of dimension. The generated spatial values are measured potentially and the set of the quality. The proposed scheme has shown a lower spatial value which also has the quality that should be set as high. The set with the lowest dimension rate has 0 as the spatial value and the largest spatial value as 1 for the highest dimension rates. The spatial pool consists of dimension rates of many sets and each of them provides the spatial value for selection. Each solution is provided to set the dimension rates for selecting based on the tournament. The set of dimension rates is chosen as it comes with a spatial bounding process. The solution is obtained as each whale that shows the maximum and minimum dimensions which are calculated as shown in (14) and (15).

$$\text{Maximum dimension} = \text{fix}(d_{max} \cdot D) \quad (14)$$

$$\text{Minimum dimension} = \text{ceil}(d_{min} \cdot D) \quad (15)$$

From the (14) and (15)  $D$  are known as the total number of features, and  $(\cdot)$  is the dot product of maximum dimensions and minimum dimensions. Maximum and minimum dimension rates are represented as  $d_{max}$  and  $d_{min}$  that have corresponded with a solution provided through the tournament selection. The  $\text{ceil}$  and  $\text{fix}$  are the two sorts of operators to round the operations. At each iteration, the spatial values are updated as they are important to run the main procedure when they are spatially updated. The spatial value has been updated with the mean spatial value which obtains the fitness function. The spatial value was updated by the mean value and its previous fitness function with the spatial values as a solution. If the best fitness value is selected, the updated spatial value has generated with the smaller values that are corresponded to the set. Therefore, the higher chances of selecting them are important for the next generation. The classification of error rate (CEE) is referred to as the objective or fitness function that evaluates the performances to select the features and provides solutions. The fitness function is evaluated using (16). From each of the datasets, the instances of the dataset are spitted into training and testing of 70% and 30% samples. The training samples for feature selection and testing samples are undergone for the testing session.

$$\text{Fit} = \text{CEE} = \frac{\text{Number of wrongly classified}}{\text{Total number of instances}} \quad (16)$$

### 2.3.2. Hyper parameter optimization using weighted salp swarm optimization algorithm

The weighted SSA algorithm showed the modification with the SSA algorithm. The SSA has been fascinated with the Salp Swarming of the Seas. The population in the Salp Swarm has been classified as followers and leaders [26], [27]. The first Salp is known as the leader and the rest are called the followers. The positions of the Salp are defined with the variables and the optimization problem is overcome. The double-dimensional matrix is named  $X$  that is expressed as shown in (17).

$$X = [x_{i1}, x_{i2}, x_{i3}, \dots, x_{ij}, \dots, x_{sv}] \quad (17)$$

From the (17),  $i \in [1,2,3, \dots V]$ ,  $S$  is called the population size,  $V$  is known as the number of variables. The Salp positions are initially referred to as  $x_{ij}$  in a matrix that is having a size of  $S$ ,  $V$  is determined using the (18):

$$x_{ij_{initial}} = lb_j + rand \times (ub_j - lb_j) \tag{18}$$

where  $lb_j$  and  $ub_j$  are called as the upper and the lower bound belonging to the  $j^{th}$  variable. The random number  $rand$  is generated with an interval of  $[0,1]$ . The positions of the Salp in each population are considered to be determined by the value of the objective function. If the problem has occurred, then the parameters which are considered are required for optimization. The parameters are optimized for obtaining the best solutions. The optimized parameters are set as the best population among the Salp Position matrix. The values are determined by reducing and the OF values are maximized. The best parameters are determined which are called the positions of the food source. The leader is updating their position based on (19).

$$x_{1j} = \begin{cases} f_j + a_1 \times (lb_j + a_2 \times (ub_j - lb_j)), & a_3 \geq 0.5 \\ f_j - a_1 \times (lb_j + a_2 \times (ub_j - lb_j)), & a_3 < 0.5 \end{cases} \tag{19}$$

From the (19), the terms  $f_j$  is known as the food source position having  $j^{th}$  variable.  $a_1 = 2e^{-\left(\frac{4t}{T}\right)}$ .  $t$  is known as the current iteration having total iterations as  $T$ .  $a_2, a_3$  are known as the random numbers which are present in the interval  $[0, 1]$ . Also, the followers are updated with their positions using the (20).

$$x_{ij} = \frac{1}{2} \times (x_{ij} + x_{i-1j}) \tag{20}$$

The SSA controls the search capability among the global and local search in the population. The controlling parameters improve the performances of the algorithms thus weighted SSA. The weighted SSA generates the dynamic weight element  $w$  that incorporated the followers' positions which are updated by using (21) and (22).

$$x_{ij} = \frac{1}{2} \times w_t \times (x_{ij} + x_{(i-1)j}) \tag{21}$$

$$w_t = w_{max} \times a_4 - \left(\frac{t}{T}\right) \times (w_{max} - w_{min}) \tag{22}$$

where  $w_t$  is the  $t^{th}$  iteration weight element.  $w_{min}$  and  $w_{max}$  are at the lower and upper limit which consisted of weight elements as  $w$  that are present between the interval of  $[0,1]$ . The proposed weighted SSA has validated the performances by implementing certain benchmark functions. Thus, the portion is included as per that and it is found that the literature has generated the value of  $w_{min} = 0.4$  and  $w_{max} = 0.9$ . The results obtained by the algorithm showed better accuracy. The random number  $a_4$  is generated in the interval as  $[0,1]$ .

The process of optimization is involving the following steps that follow:

- Step 1: The size of the population is represented as  $S$  that are having  $T$  are  $X$  initialized.
- Step 2: The population fitness is used in (16) and the positions are retained with the fittest population having the source position  $f_j$  which are known as the global initial population.
- Step 3: The populations' positions are updated. The position of the leader is updated according to (19) and the followers using (21).
- Step 4: The populations' fitness is evaluated and showed better than that of the position of the food source is replaced with new positions.
- Step 5: Steps 3 and 4 are repeated until the last iteration and the optimization algorithm performs the exploitation and exploitation phase. Here, the significant properties are higher in the searching process.

In weighted SSA selects 38 features are selected among the 45 features that were extracted. An adaptive weight component are incorporated with the weight parameters which are adjusted dynamically based on the iteration numbers for adopting the new search environment. The balanced exploitation and exploration have resulted in higher global search capability. Thus, the next chance of getting accurate result is important. Thus, an effective enhanced optimization technique is performed. The weighted SSA is improving its performance by including the dynamic weight factor for updating the position in step 3. Once the weight factor is included, the position is updated by searching strategy made the weighted SSA which is promising as an optimization algorithm.

#### 2.4. Classification using Bi-directional long short-term memory

The LSTM model has the capacity for retaining important information in long term based on cell and the forget gate. The arrhythmia signal required data from the previous steps too and the LSTM model showed an advantage in handling long-term dependence problems in the hidden layer using the self-feedback method. The memory cell has three gates as forget gate, input gate, and output gate for storing the information in the LSTM model helped for handling long-term features problem [28]-[30]. Figure 2 shows the architecture of Bi-LSTM model.

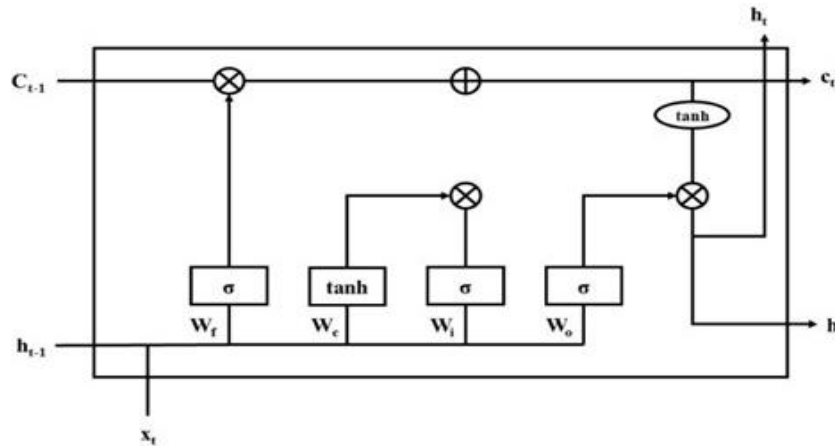


Figure 1. Architecture of Bi-LSTM network

The output from the LSTM cell is obtained which is represented as  $h_t$ ,  $c_t$  represents the memory cell value,  $h_{t-1}$  is the output of the LSTM cell. The LSTM unit has processed the following steps which are explained in steps:

- A. The memory cell is represented as  $\tilde{c}_t$  and the bias is represented as  $b_c$ , the weight matrix is represented as  $W_c$  which is expressed as shown in the (23).

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (23)$$

- B. The input gate is represented as  $i_t$ , the current input data update of the memory cell controls the state value by the input gate, the  $b_i$  represents the bias,  $W_i$  is denoted as the weight matrix, and the sigmoid function is  $\sigma$ . The input gate is represented as shown in (24).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (24)$$

- C. The  $f_t$  represents forget gate value which calculates, the state value of memory based on historical data. It is updated by controlling the forget gate,  $b_f$  represents the bias,  $W_f$  is known as the weight matrix. The forget gate equation is represented as shown in (25).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (25)$$

- D. The memory cell that is currently represented as  $c_t$  and the unit state value of the LSTM is represented as  $c_{t-1}$ , which is provided in (26):

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \quad (26)$$

from the (26), '\*' is known as the dot product,  $i_t$  is the input gate of the cell,  $f_t$  is the forget gate of the cell.

- E.  $o_t$  is the output gate that is calculated based on the memory cell state which is controlled by the output gate. The term  $b_o$  represents the bias and the  $W_o$  is represented as the weight matrix which is represented as shown in (27).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (27)$$

- F. The output  $h_t$  obtained from the LSTM is calculated which is given as shown in (28).



$$h_t = o_t * \tanh(c_t) \tag{28}$$

LSTM model update, reset, read and keep long-time information easily based on memory cell and control gates. The LSTM model sharing mechanism of internal parameters controls the output dimensions based on weight matrix dimensions' settings. Each token of the sequence is used by two LSTMs in Bi-LSTM based on the future and past context of the token [29], [30]. LSTM process the sequence from left to right (forward) and another one from right to left (backward). The hidden unit function  $\vec{h}$  of a hidden forward layer at each time step  $t$  is computed based on the input current step  $x_t$  and previous hidden state  $h_{t-1}$ . The hidden unit function  $\vec{h}$  of a hidden backward layer is computed based on input current step  $x_t$  and future hidden state  $\vec{h}_{t+1}$ . The backward and forward context representation generated by  $\vec{h}_t$  and  $\vec{h}_t$ , respectively are concatenated into a long vector. The classification of combined outputs of teacher-given target signals. The model classifies the model into 9 classes for Carnatic music and 7 classes in Hindustani music. Where Carnatic music includes bagesri, bhairavi, nata, kalyani, madhyamvati, sindhubhairavi, yamankalyani, purvikalyani. Whereas, Hindustani music includes bagesri, bhairavi, hamsadhvani, madhyamavati, mulkauns, todi, and yamankalyani.

### 3. RESULTS AND DISCUSSION

The proposed hybrid optimization algorithm is used for the classification of raga based on MATLAB 2018a which is operating at the i7 core processor. The memory is installed which is having the random access memory (RAM) is 16GB operating at a 4.20 GHz system frequency having 64-bit operating system (OS). The ragas present are divided with 10 segments to perform experimentation and the cross-validation is operated for the samples of the raga. The result section is segregated into quantitative and comparative analyses. The hyper parameter settings for the proposed research work are given as shown in Table 1.

Table 1. Hyper parameter settings

Parameters	Units
LearnRateDropFactor	0.1
InitialLearnRate	0.001
MaxEpochs	300
L2Regularization	0.001
Optimizer	Adam
Minibatch size	27
Hidden layer	100 units
Gradient threshold	1
Executive environment	Graphics Processing Unit (GPU)
LearnRateDropFactor	0.1
InitialLearnRate	0.001
MaxEpochs	300
L2Regularization	0.001

#### 3.1. Performance metrics

The performance measures are utilized for evaluating the results for classification which are as follows: Accuracy is the ratio of totally predicted observations to the total observations (29) represents the accuracy terms. Specificity is defined as the ratio of negatives that are correctly identified which is expressed as shown in (30). Specificity is defined as the ratio of positives to that of the correctly identified is expressed as shown in the following (31). PPV is called the positive terminal to the sum of true positive (TP) and false positive (FP) values which are expressed in (32). MCC is a correlation coefficient between the predicted values and the true values. The MCC equation is represented in (33). F1 score is the harmonic mean of precision and recall which is evaluated using (34). Where, TP is true positive, FP is false positive, TN is true negative and FN is false negative values.

$$Accuracy(\%) = \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100 \tag{29}$$

$$Specificity(\%) = \frac{TN}{TN+FP} \times 100 \tag{30}$$

$$Sensitivity(\%) = \frac{TP}{TP+FN} \times 100 \tag{31}$$

$$PPV(\%) = \frac{TP}{TP+FP} \times 100 \tag{32}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (33)$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (34)$$

### 3.2. Quantitative analysis

The hybrid spectral feature extraction technique has extracted the spectral features that were fed for the Bi-LSTM classifier to classify the raga. 5 spectral features were considered for evaluation and the results are tabulated as shown in Tables 2-7. Features named as: Spectral Centroid, Spread, Skewness, MFCC and LPC were extracted and fed for the classifiers. The obtained spectral features were combined into hybrid spectral features gave better results compared to each of the spectral features. The existing methods failed for considering the feature numbers for performing mood classification. The results obtained from the classifiers are given in Table 2, for without hyper parameter tuning and without weighted SSA feature selection. Table 3 shows results of the classifiers, with hyper parameter tuning and with weighted SSA feature selection. Table 4 shows the results obtained in terms of the performance metrics, by the proposed weighted SSA and various optimizers, by evaluating on Carnatic music. Table 5 shows the results obtained without hyperparameter tuning and without feature selection. Table 6 is the results obtained with hyper parameter tuning and with weighted feature selection. Table 7 is the results obtained by the proposed weighted SSA and various optimizers, by means of the performance metrics and the evaluation dataset used was Hindustani music. The Figure 3 shows the sample signals of Carnatic music and Figure 4 shows the sample raga signals from Hindustani music.

Table 2. Without hyper parameter tuning and without weighted SSA feature selection

Carnatic music dataset					
Classifier	Accuracy (%)	Sensitivity (%)	Specificity (%)	MCC (%)	F1-score(%)
KNN	74.00	75.08	73.57	74.21	74.64
RF	87.75	85.78	87.48	86.42	87.08
DNN	63.40	63.60	61.25	62.09	62.84
DE	85.26	87.14	83.26	82.37	85.75
Bi-LSTM	91.39	93.33	93.28	94.26	93.79

Table 3. With hyper parameter tuning and with weighted SSA feature selection

Classifier	Accuracy (%)	Sensitivity (%)	Specificity (%)	MCC (%)	F1-score (%)
KNN	76.34	75.81	76.72	74.24	74.64
RF	88.39	85.29	88.44	88.43	87.06
DNN	66.53	63.83	61.25	64.07	63.83
DE	87.95	87.45	89.21	88.37	87.73
Bi-LSTM	96.63	96.96	95.8	97.96	96.68

Table 4. Results obtained for the proposed weighted SSA with various performance metrics

Optimization Algorithm	Accuracy (%)	Sensitivity (%)	Specificity (%)	MCC (%)	F1-score (%)
PSO	90.76	91.71	92.82	91.44	89.49
GOA	83.74	84.03	85.69	86.38	84.45
ABC	85.39	84.28	86.32	88.77	85.65
Weighted SSA(Proposed)	96.63	96.96	95.80	97.96	96.68

Table 5. Results obtained without hyper parameter tuning and without feature selection

Hindustani music					
Classifier	Accuracy (%)	Sensitivity (%)	Specificity (%)	MCC (%)	F1-score (%)
RF	83.68	0.96	0.26	0.55	0.25
DNN	50.66	0.34	0.51	0.14	0.81
DE	83.16	0.59	0.70	0.15	0.24
Bi-LSTM	94.10	93.60	94.89	93.67	93.63

Table 6. Results obtained with hyper parameter tuning and with weighted feature selection

Classifier	Accuracy (%)	Sensitivity (%)	Specificity (%)	MCC (%)	F1-score (%)
KNN	67.14	66.76	69.96	63.33	65.00
RF	88.29	86.98	87.98	88.84	87.90
DNN	54.29	57.65	55.84	56.86	57.25
DE	86.57	85.87	87.57	84.43	85.14
Bi-LSTM	94.91	94.22	95.89	94.26	93.93

Table 7. Results obtained for the proposed weighted SSA with various performance metrics

Optimization Algorithm	Accuracy (%)	Sensitivity (%)	Specificity (%)	MCC (%)	F1-score (%)
PSO	89.20	90.83	91.76	88.05	89.57
GOA	88.25	89.59	90.75	87.53	88.47
ABC	85.62	84.55	86.38	85.78	85.01
Weighted SSA (Proposed)	94.91	94.22	95.89	94.26	93.93

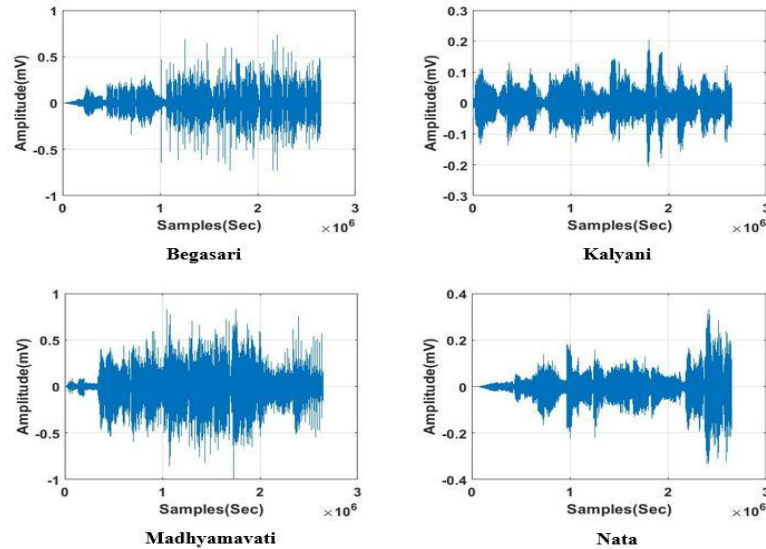


Figure 3. Sample Carnatic music

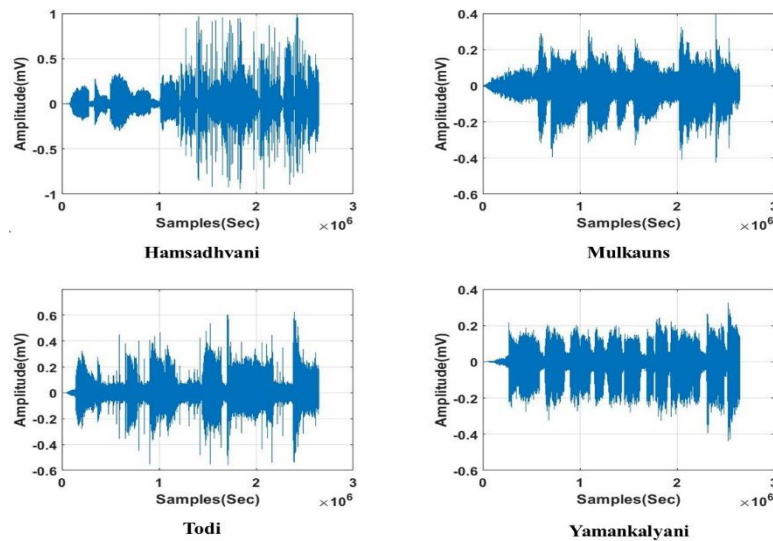


Figure 4. Sample Hindustani music

### 3.3. Comparative analysis

In Table 8, the comparative analysis of the proposed and the existing models' results are provided. The existing fuzzy analytical hierarchy process-based approach obtained sensitivity or recall of 77% of specificity of 53.61%, PPV or Precision of 78.12%, F-score of 71.20%. Similarly, the neural network based dragonfly algorithm for the Carnatic music dataset obtained 68% of sensitivity, specificity of 72%, and PPV or Precision of 67%. The neural network based dragonfly algorithm for the Carnatic music obtained sensitivity of 68%, specificity of 72%, precision of 67%. The CNN model obtained 54% of specificity for the Carnatic music and SVM obtained 70.52 % of specificity. However, the proposed ESBWOA-weighted SSA Hindustani music obtained accuracy of 94.91%, sensitivity of 94.22%, specificity of 95.89%, and F-score of 93.93%. The accuracy of 96.63%, 96.96% of sensitivity, specificity of 95.80%, and F1-score of 96.68% for Carnatic music.

Table 8. Comparative analysis

Authors	Methodology	Dataset	Accuracy (%)	Sensitivity or recall (%)	Specificity or true negative rate (%)	PPV or Precision (%)	F1-score (%)
Kaur and Kumar [16]	Fuzzy analytical Hierarchy process-based approach	Hindustani	-	77	53.61	78.12	71.20
Kiran [17]	Neural network based Dragonfly algorithm	Carnatic	68	72	67	-	30
Sharma <i>et al.</i> [18]	Time-series matching	Hindustani	78.51	92.77	-	77	84.15
John <i>et al.</i> [19]	Deep learning approach	Carnatic	-	-	-	-	-
Sarkar <i>et al.</i> [20]	Convolutional neural network	Carnatic	94	-	-	-	-
Proposed method	Support vector machine	Hindustani	70.52	-	-	-	-
	ESBWOA-weighted SSA	Hindustani	94.91	94.22	95.89	-	93.93
		Carnatic	96.63	96.96	95.80	-	96.68

#### 4. CONCLUSION

The identification of ragas is based on the tone or pitch levels where the emotion type is conveyed based on relations. The extraction or identification of music sampled features were needed to enhance the success rate to overcome the classification problem. The CompMusic dataset is used in the research for extracting the ragas. The white noises are generated by the audio signals as they were corrupted for raga classification. The process of normalization is performed for the removal of unwanted noises to process further. The proposed hybrid feature extraction technique from the signals are representing the combination of features that predict the raga based on the complex model. The proposed ESBWOA is used that overcome the Feature Selection problem of high dimensional features which reduces the feature space without losing the feature properties and the MSVM classifier for classification enhanced the success rate that classified the model into 9 classes for Carnatic music and 7 classes in Hindustani music. Where Carnatic music includes bagesri, bhairavi, nata, kalyani, madhyamvati, sindhubhairavi, yamankalyani, purvikalyani. The proposed method obtained accuracy of 94.09%, Sensitivity of 93.33%, Specificity of 93.28%, Precision of 94.26, and F-score of 93.79. Also, the results obtained for Carnatic music in terms of accuracy as 94.39%, the sensitivity of 93.6%, specificity of 94.89%, Precision of 93.67, and F-score of 93.63%. However, the usage of more number of features classifier makes decision-making a tricky process. In future, the complexity problem can be overcome by providing an efficient automatic system for raga recognition.





#### REFERENCES

- [1] S. Sharma *et al.*, "Indian classical music with incremental variation in tempo and octave promotes better anxiety reduction and controlled mind wandering—a randomised controlled EEG study," *Explore*, vol. 17, no. 2, pp. 115–121, 2021, doi: 10.1016/j.explore.2020.02.013.
- [2] M. S. Sinith, S. Tripathi, and K. V. V. Murthy, "Raga recognition using fibonacci series based pitch distribution in Indian Classical Music," *Applied Acoustics*, vol. 167, p. 107381, 2020, doi: 10.1016/j.apacoust.2020.107381.
- [3] S. Chapaneri and D. Jayaswal, "Deep Gaussian processes for music mood estimation and retrieval with locally aggregated acoustic Fisher vector," *Sādhanā*, vol. 45, p. 73, 2020, doi: 10.1007/s12046-020-1313-8.
- [4] S. Das, S. Satpathy, S. Debbarma, and B. K. Bhattacharyya, "Data analysis on music classification system and creating a sentiment word dictionary for Kokborok language," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2019, doi: 10.1007/s12652-019-01565-y.
- [5] D. Chaudhary, N. P. Singh, and S. Singh, "Automatic music emotion classification using hashtag graph," *International Journal of Speech Technology*, vol. 22, no. 3, pp. 551–561, 2019, doi: 10.1007/s10772-019-09629-2.
- [6] S. Mo and J. Niu, "A novel method based on OMPGW method for feature extraction in automatic music mood classification," *IEEE Transactions on Affective Computing*, vol. 10, no. 3, pp. 313–324, 2019, doi: 10.1109/TAFFC.2017.2724515.
- [7] B. Kumaraswamy and P. G. Poonacha, "Recognizing ragas of Carnatic genre using advanced intelligence: a classification system for Indian music," *Data Technologies and Applications*, vol. 54, no. 3, pp. 383–405, 2020, doi: 10.1108/DTA-04-2019-0055.
- [8] V. S. Pendyala, N. Yadav, C. Kulkarni, and L. Vadlamudi, "Towards building a deep learning based automated indian classical music tutor for the masses," *Systems and Soft Computing*, vol. 4, p. 200042, 2022, doi: 10.1016/j.sasc.2022.200042.
- [9] V. Chaturvedi, A. B. Kaur, V. Varshney, A. Garg, G. S. Chhabra, and M. Kumar, "Music mood and human emotion recognition based on physiological signals: a systematic review," *Multimedia Systems*, pp. 1–24, 2021, doi: 10.1007/s00530-021-00786-6.
- [10] R. B. Bouncken, Y. Qiu, N. Sinkovics, and W. Kürsten, "Qualitative research: extending the range with flexible pattern matching," *Review of Managerial Science*, vol. 15, no. 2, pp. 251–273, 2021, doi: 10.1007/s11846-021-00451-2.
- [11] S. John, M. S. Sinith, R. S. Sudheesh, and P. P. Lal, "Classification of Indian classical carnatic music based on raga using deep learning," In *2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pp. 110–113, 2020, doi: 10.1109/RAICS51191.2020.9332482.
- [12] A. K. Sharma *et al.*, "Classification of Indian classical music with time-series matching deep learning approach," *IEEE Access*, vol. 9, pp. 102041–102052, 2021, doi: 10.1109/ACCESS.2021.3093911.





- [13] A. Bhat, A. V. Krishna, and S. Acharya, "Analytical comparison of classification models for raga identification in carnatic classical instrumental polyphonic audio," *SN Computer Science*, vol. 1, no. 6, pp. 1–9, 2020, 10.1007/s42979-020-00355-0.
- [14] A. George, X. A. Mary, and S. T. George, "Development of an intelligent model for musical key estimation using machine learning techniques," *Multimedia Tools and Applications*, vol. 81, no. 14, pp. 19945–19964, 2022, doi: 10.1007/s11042-022-12432-y.
- [15] S. Nag, M. Basu, S. Sanyal, A. Banerjee, and D. Ghosh, "On the application of deep learning and multifractal techniques to classify emotions and instruments using Indian classical music," *Physica A: Statistical Mechanics and its Applications*, vol. 597, p. 127261, 2022, doi: 10.1016/j.physa.2022.127261.
- [16] C. Kaur and R. Kumar, "A fuzzy hierarchy-based pattern matching technique for melody classification," *Soft Computing*, vol. 23, no. 16, pp. 7375–7392, 2019, doi: 10.1007/s00500-018-3383-7.
- [17] B. K. Kiran, "Indian music classification using neural network based dragon fly algorithm," *Journal of Computational Mechanics, Power System and Control*, vol. 4, no. 3, pp. 32–40, 2021, doi: 10.46253/jcmps.v4i3.a5.
- [18] A. K. Sharma *et al.*, "Classification of Indian classical music with time-series matching deep learning approach," *IEEE Access*, vol. 9, pp. 102041–102052, 2021, doi: 10.1109/ACCESS.2021.3093911.
- [19] S. John, M. S. Sinith, R. S. Sudheesh, and P. P. Lalu, "Classification of Indian classical carnatic music based on raga using deep learning," *In 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pp. 110–113, 2020, doi: 10.1109/RAICS51191.2020.9332482.
- [20] R. Sarkar, S. K. Naskar, and S. K. Saha, "Raga identification from Hindustani classical music signal using compositional properties," *Computing and Visualization in Science*, vol. 22, no. 2, pp. 15–26, 2019, doi: 10.1007/s00791-017-0282-x.
- [21] D. P. Shah, N. M. Jagtap, P. T. Talekar, and K. Gawande, "Raga recognition in indian classical music using deep learning," *10th International Conference, Virtual Event*, pp. 248–263, 2021, doi: 10.1007/978-3-030-72914-1\_17.
- [22] A. Krishnaiah and P. B. Divakarachari, "Automatic music mood classification using multi-class support vector machine based on hybrid spectral features," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 5, pp. 102–111, 2022, doi: 10.22266/ijies2021.1031.10.
- [23] S. Gulati *et al.*, "Automatic tonic identification in Indian art music: approaches and evaluation," *Journal of New Music Research*, vol. 43, no. 1, pp. 53–71, 2014, doi: 10.1080/09298215.2013.875042.
- [24] Y. Zhang *et al.*, "WOCDA: A whale optimization-based community detection algorithm," *Physica A: Statistical Mechanics and its Applications*, vol. 539, p. 122937, 2020, doi: 10.1016/j.physa.2019.122937.
- [25] J. Gholami, F. Pourpanah, and X. Wang, "Feature selection based on improved binary global harmony search for data classification," *Applied Soft Computing*, vol. 93, p. 106402, 2020, doi: 10.1016/j.asoc.2020.106402.
- [26] S. B. Chaabane, A. Belazi, S. Kharbech, A. Bouallegue, and L. Clavier, "Improved salp swarm optimization algorithm: application in feature weighting for blind modulation identification," *Electronics*, vol. 10, no. 16, p. 2002, 2021, doi: 10.3390/electronics10162002.
- [27] E. Akbari *et al.*, "Improved salp swarm optimization algorithm for damping controller design for multimachine power system," *IEEE Access*, vol. 10, pp. 82910–82922, 2022, doi: 10.1109/ACCESS.2022.3196851.
- [28] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019, doi: 10.1162/neco\_a\_01199.
- [29] K. Smagulova and A. P. James, "A survey on LSTM memristive neural network architectures and applications," *The European Physical Journal Special Topics*, vol. 228, no. 10, pp. 2313–2324, 2019, doi: 10.1140/epjst/e2019-900046-x.
- [30] A. H. Bukhari, M. A. Z. Raja, M. Sulaiman, S. Islam, M. Shoaib, and P. Kumam, "Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting," *IEEE Access*, vol. 8, pp. 71326–71338, 2020, doi: 10.1109/ACCESS.2020.2985763.

## BIOGRAPHIES OF AUTHORS



**Bettadamadahally Shivakumaraswamy Gowrishankar**     is a faculty of Computer science, having keen interest in IOT, signal and image data processing, currently pursuing Ph.D from VTU. His areas of interest include IOT, data analytics, and signal data analytics. He can be contacted at email: gowrish.vvce@gmail.com.



**Nagappa U. Bhajantri**     is a Professor of Computer science, having keen interest in data analytics, signal and image data processing, currently a professor at Government College of Engineering. He has successfully guided many scholars for Doctoral Degree and has a vast experience in research and teaching. He can be contacted at email: bhajan3nu@gmail.com.