IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 3, 2020, accepted September 28, 2020, date of publication October 1, 2020, date of current version October 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3028241

A Hybrid Meta-Heuristic Feature Selection Method for Identification of Indian Spoken Languages From Audio Signals

AANKIT DAS[1], SAMARPAN GUHA[1], PAWAN KUMAR SINGH[2], (Member, IEEE), ALI AHMADIAN[3,4], (Member, IEEE), NORAZAK SENU[4], AND RAM SARKAR[5], (Senior Member, IEEE)

[1]Institute of Radio Physics and Electronics, University of Calcutta, Kolkata 700009, India
[2]Department of Information Technology, Jadavpur University, Kolkata 700106, India
[3]Institute of IR 4.0, The National University of Malaysia (UKM), Bangi 43600, Malaysia
[4]Institute for Mathematical Research, Universiti Putra Malaysia, Serdang 43400, Malaysia
[5]Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, India

Corresponding author: Ali Ahmadian (ahmadian.hosseini@gmail.com)

This work was supported in part by the Fundamental Research Grant Scheme (FRGS) provided by the Ministry of Education, Malaysia, under Project FRGS/1/2018/STG06/UPM/02/2, and in part by Universiti Putra Malaysia.

ABSTRACT With the recent advancements in the fields of machine learning and artificial intelligence, spoken language identification-based applications have been increasing in terms of the impact they have on the day-to-day lives of common people. Western countries have been enjoying the privilege of spoken language recognition-based applications for a while now, however, they have not gained much popularity in multi-lingual countries like India owing to various complexities. In this paper, we have addressed this issue by attempting to identify different Indian languages based on various well-known features like Mel-Frequency Cepstral Coefficient (MFCC), Linear Prediction Coefficient (LPC), Discrete Wavelet Transform (DWT), Gammatone Frequency Cepstral Coefficient (GFCC) as well as a few deep learning architecture based features like i-vector and x-vector extracted from the audio signals. After comparing the initial results, it is observed that the combination of MFCC and LPC produces the best results. Then we have developed a new nature-inspired feature selection (FS) algorithm by hybridizing Binary Bat Algorithm (BBA) with Late Acceptance Hill-Climbing (LAHC) to select the optimal subset from the said feature vectors in order to reduce the model complexity and help it train faster. Using Random Forest (RF) classifier, we have achieved an accuracy of 92.35% on Indic TTS database developed by IIT-Madras, and an accuracy of 100% on the Indic Speech database developed by the Speech and Vision Laboratory (SVL) IIIT-Hyderabad. The proposed algorithm is also found to outperform many standard meta-heuristic FS algorithms. The source code of this work is available at: https://github.com/CodeChef97dotcom/Feature-Selection

INDEX TERMS Spoken language identification, feature selection, binary Bat algorithm, late acceptance hill climbing algorithm, MFCC and LPC features.

I. INTRODUCTION

Speech is one of the most innate human capabilities. When we speak with one another, we use not just words but also associated emotions and sentiments to convey meaning and get our opinions across. There are many features associated with spoken language that allow us to deliver information that go far beyond just our words. Spoken language involves the actual use of speech or related utterances that convey meaning to share the thoughts or other information. Processing of spoken languages involves human-computer interaction (HCI) which has significantly improved over the last decade. Automatic language identification plays a vital role in a wide range of services. Nowadays, almost everyone is equipped with smartphones which makes life much easier. People can now control their daily activities like calling someone, turning on

The associate editor coordinating the review of this manuscript and approving it for publication was K. C. Santosh.

or off electrical appliances, making financial transactions and various other activities just by instructing their smartphone through oral commands. This is made possible only due to the sophisticated speech or language recognition algorithms that come with smartphones. However, when it comes to multi-lingual countries like India, the complexity of classifying these languages becomes extremely difficult owing to the language-specific syntaxes, dialects, idiomatic expressions, and accents, etc.

India houses more than 19,500 languages or dialects according to an analysis of a census release in 2018. The Indian constitution recognizes 23 major languages what is known as ''the 8th Schedule'' of the Constitution. They also happen to be the major literary languages in India, with a considerable volume of writing in them. Besides Sanskrit, 22 other modern Indian languages are Assamese, Bangla, Bodo, Dogri, Gujarati, Hindi, Kashmiri, Kannada, Konkani, Maithili, Malayalam, Manipuri, Marathi, Nepali, Oriya, Punjabi, Tamil, Telugu, Santali, Sindhi, English, and Urdu. Each language has its own acoustic, prosodic, and phonotactic features. Therefore, the task of recognizing [1] these languages with their own vernacular, cadence, semantics, and varied ambience becomes extremely complex especially if the number of features extracted to distinguish different languages is very large. There are even some features that do not carry useful information about the audio signals, and therefore including these features for the purpose of classification proves to be counterproductive as they often bring down the performance of the model both in terms of accuracy as well as efficiency. Hence, it is imperative to use the appropriate feature subset that carries enough information about the signal for the learning model to identify the different languages accurately. This reduces the model complexity both in terms of hardware requirements as well as computation time. In this paper, we explore a new approach to develop a feature selection (FS) algorithm using a hybrid of Binary Bat Algorithm (BBA) and Late Acceptance Hill-Climbing (LAHC) algorithm for classifying Indian languages based on their Mel-frequency Cepstral Coefficient (MFCC) and Linear Prediction Coefficient (LPC) features. Additionally, we have evaluated recently proposed features such as i-vector [2], x-vector [3], a fusion of Discrete Wavelet Transform (DWT) and MFCC features [4], and a combination of MFCC and GFCC [5] for feature extraction and have also evaluated the performance of the proposed FS algorithm on these features.

Lately, India has seen a major surge in Artificial Intelligence (AI) based applications. People nowadays rely much more on electronic gadgets for their day-to-day lives. Applications like Apple Siri, Google Assistant, Amazon Alexa, etc. have only made lives easier by automating various jobs like performing online transactions, controlling various home appliances, and so on, all by simple voice commands. In these speech-based assistants, spoken language identification acts as the first step which chooses the corresponding grammar from a list of available languages for analyzing further the semantics of the languages. However, in the multi-lingual scenario prevailing in India, the performance of traditional algorithms, which do not make proper use of features extracted from audio signals, are found to be quite low which makes them unreliable for language identification. Our proposed work is one step closer to resolving this particular issue.

In the rest of the paper, the related works are presented in Sect. II. The motivation for our work is presented in Sect. III and the proposed methodology is presented in Sect. IV. The experimental results are presented in Sect. V. A comparative study of our work is presented in Sect. VI and the conclusion is presented in Sect. VII.

II. RELATED WORK

Koolagudi *et al.* [6] had identified 15 Indian languages using MFCC features and Gaussian Mixture Model (GMM). The training set and the test set were divided into speech data which comprised 5 min and 1.5 min of speech data for each language respectively. The validation set comprised of clips of 2 seconds from each language. The final accuracy obtained from this system was 88%. Zissman and Singer [7] obtained an accuracy of 79.2% for 10 languages on the OGI multiple language telephonic speech corpus. Tang *et al.* [8], in 2017, applied deep learning-based techniques to identify 4 languages, namely Bangla, Gregorian, Turkish, and Assamese, from the Babel corpus. They had used phonetic information coupled with Recurrent Neural Network (RNN). The results obtained had better accuracy than the baseline RNN system. Gunawan *et al.* [9] identified 5 languages, namely English, Malay, Korean, Chinese, and Arabic using MFCC features coupled with vector quantization on a database put together with the help of 10 volunteers. Their obtained individual accuracies were 90% for English, 80% for Malay, 80% for Korean, 60% for Chinese, and 100% for Arabic. Bekker *et al.* [10] used a different approach to identify spoken languages. They coupled various intra-cluster training strategy with deep learning and used this technique on the NIST 2015 language identification dataset. Cluster Munition Coalition (CMC) clustering method achieved the lowest misclassification rate of 13.5% compared to other clustering methods used. Mukherjee [11] identified 4 different languages using both MFCC and LPC features. They applied Artificial Neural Network (ANN) for the purpose of classification on a database composed of speech of 20 people and obtained an accuracy of 88%. 4 Indian languages were also distinguished by Mohanty [12] using 1 manner feature and 10 place features using a Support Vector Machine (SVM) classifier. Their system was based on a corpus of 500 words for every language. Average recognition accuracy of 89.75% at word level was obtained for the system. Bartz *et al.* [13] had used a hybrid Convolutional RNN which operated on spectrogram images of the provided audio snippets. They had collected their datasets from speeches and statements from the European Parliament and news broadcast channels hosted on YouTube. Their work included the identification of 6 languages namely English, German, French,

Spanish, Russian, and Mandarin Chinese. They noted that Mandarin Chinese outperformed every other language with a top accuracy of 96%. Sarthak *et al.* [14], in their work, proposed attention-based deep learning model for language identification using log-Mel spectrogram images as input for identification of 6 languages. They classified 6 languages with an accuracy of 95.4% and 4 languages with an accuracy of 96.3% obtained from the VoxForge dataset. Mukherjee *et al.* [15] proposed line spectral frequency-based features for modeling the 7 Indian languages taken from IIIT-H Indic Speech Database. They achieved the highest accuracy of 99.71% using ensemble learning-based classification technique. Revay and Teschke [16] used a deep learning approach on spectral images of audio signals for multi-class language identification. They obtained an average accuracy of 89% for multi-class classification of 6 languages. Mukherjee *et al.* [17] developed a lazy learning-based language identification technique using MFCC-2 features for the classification of 3 Indian languages. They achieved the highest accuracy of 98.09% and an average accuracy of 95.5% respectively. In a recent paper, Mukherjee *et al.* [18] used LPC-based feature and ensemble learning technique for classification of top-seven world languages namely Chinese, Spanish, English, Hindi, Arabic, Bangla and Portuguese and the highest possible accuracy of 96.95% was received on more than 200h of real-world YouTube data.

III. MOTIVATION

Some relevant work has already been done in the field of spoken language identification, however, to the best of our knowledge, not much concrete research has been done on how to optimize the derived features before feeding the same to the language classification model. Researchers always find it difficult to identify the relevant features from a high dimensional feature vector and remove the irrelevant or less important features which do not contribute much to the target variable in order to achieve better accuracy for the learning model under consideration [19]–[22]. This problem becomes even more apparent if the number of features is very large. Datasets with higher dimensions pose computational problems especially in cases where achieving a near perfect classification accuracy score is of primary concern, since the system has constraints in terms of computation time and memory requirement. Each and every feature might not be useful for a learning algorithm. Rather, only the features which are really important can significantly improve the performance of a machine learning model. Moreover, using FS algorithms [23] to reduce dimensionality [24] can enable the machine learning algorithm to train faster. It also improves the accuracy of a model if the right subset is chosen while reducing overfitting.

To this end, in this paper, a novel hybrid nature-inspired FS algorithm has been developed using BBA-LAHC with the goal to classify spoken Indian languages. Even though there exists a plethora of literature based on the meta-heuristic FS methods, such as Particle Swarm Optimization (PSO) [25], Genetic algorithm (GA) [26] etc., in this work, BBA is chosen as the FS method over other algorithms due to the following reasons:

- BBA uses echolocation and frequency tuning to solve optimization problems. The use of frequency variations to mimic the true function provides some functionality that is similar to the key feature used in PSO. Therefore, BBA incorporates the advantageous characteristics of other swarm-intelligence-based algorithms.
- BBA has the capability of automatically zooming into a region where promising solutions have been found. As a result, BBA has a quicker convergence rate, at least at early stages of the iterations which gives this algorithm an edge over other meta-heuristic algorithms.
- Many meta-heuristic algorithms make use of fixed parameters by means o some pre-tuned algorithm-dependent parameters. In contrast, BBA uses parameter control, which varies the values of parameters accordingly as the iterations proceed. This provides a way to automatically switch from exploration to exploitation when the optimal solution is approaching.

Even though the BBA is efficient, there is still room for improvement and enhancement of the convergence of the algorithm. Therefore, to address this problem, we have incorporated LAHC as a local search algorithm with BBA.

In this work, we find that the proposed hybrid FS algorithm works better particularly in the field of spoken language identification and also has an accuracy score better than some standard meta-heuristic FS algorithms. The algorithm is able to intelligently select the optimal features from a dataset containing almost 1000 features and is also able get a better accuracy score while reducing over-fitting.

IV. METHODOLOGY

The audio clips are first passed through a pre-emphasis filter. Then the pre-emphasized audio signals are pre-processed using framing and windowing techniques. The Fourier Transform (FT) and Power spectrum of the audio signals are computed followed by the application of Filter banks. Finally, the MFCC and LPC features are extracted. Then, we apply FS algorithm using hybrid BBA-LAHC for selecting those features from the dataset which contribute most to our prediction value or output in which we are interested in. After FS is completed, four classifiers namely SVM, Random Forest (RF), Naïve Bayes and Multi Layer Perceptron (MLP) are applied for the task of classification. Here, in this section, our work is presented in three sub-sections namely Pre-processing (IV-A), Feature Extraction (IV-B), Feature Selection (IV-C). A flowchart for our proposed experimental setup is given in Figure (1).

A. PRE-PROCESSING

1) PRE-EMPHASIS

The signal is first passed through a pre-emphasis filter to amplify the energy in the high frequencies. A pre-emphasis filter is useful as it helps in balancing the frequency spectrum

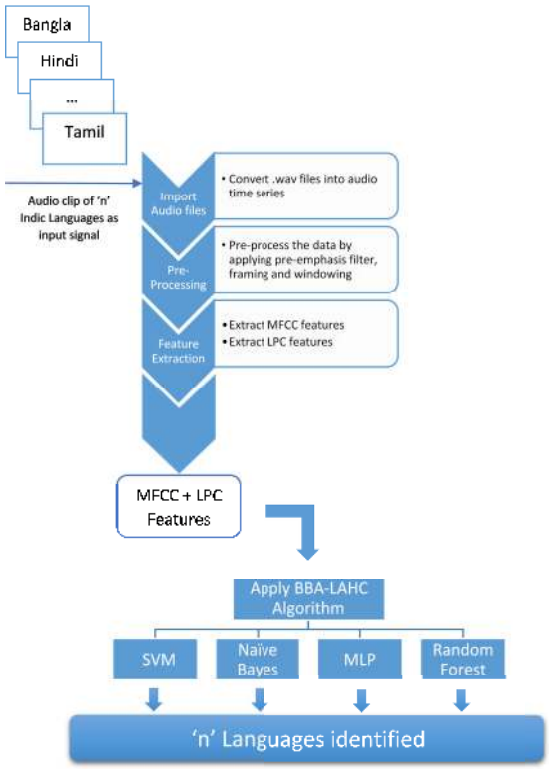


FIGURE 1. Flowchart of our proposed experimental setup for identification of spoken Indian languages.

since high frequencies usually have smaller magnitudes compared to lower frequencies. Pre-emphasis helps in avoiding problems faced during the numerical computation of the FT and also helps in improving the Signal-to-Noise Ratio (SNR).

The pre-emphasis filter can be applied to a signal x(t) using the first order filter as shown in the Eqn. (1):

$$y(t) = x(t) - \alpha x(t-1) \tag{1}$$

where the value of α is set to 0.97.

2) FRAMING

Analyzing speech signals as a whole is a difficult job mostly because the spectral characteristics of the signal does not remain the same over the entire span. Frequencies in a signal change over time, so in most cases it is not sensible to do the FT across the entire signal, otherwise we would lose the frequency contours of the signal over time [27], [28]. Hence, we assume that frequencies in a signal are stationary over a very short period of time. Therefore, by doing a FT over this short-time frame, we can obtain a good approximation of the frequency contours of the signal by concatenating adjacent frames.

3) WINDOWING

Windowing involves the slicing of the audio waveform into sliding frames. Windowing mainly gets rid of any jitters and ensuring a smoother transition from one frame to the next. Windowing is applied in spectral estimation to produce continuity at the edges, with some windows introducing up to 2^{nd} order continuity. Without it signal mismatch can occur which leads to undesirable results. In this experiment, Hamming window [29], shown in Figure (2), is chosen to deal with the aforementioned problem and mathematically it is illustrated in Eqn. (2):

$$w(m) = 0.54 - 0.46cos(2\pi m/M - 1) \tag{2}$$

where w(m) is the function of hamming window; m ranges from start to end of the frame (0–255 here); and M is the frame size (256 here).

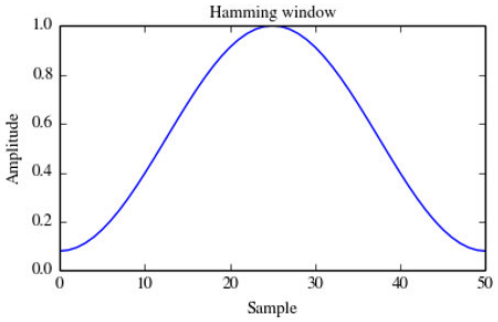


FIGURE 2. Hamming window used as a preprocessing step to reduce spectral leakages.

4) FOURIER TRANSFORM AND POWER SPECTRUM

The N-point Fast FT (FFT), which is also called Short-Time-Fourier-Transform (STFT), is applied on each frame to calculate the frequency spectrum of the sign. The Power Spectrum (periodogram) is calculated using Eqn. (3):

$$P = \frac{|FFT\ (x_i)\ |^2}{N} \tag{3}$$

where x_i is the i^{th} frame of the signal.

5) FILTER BANKS

Filter banks, shown in Figure (3), are arrangements of low pass, band pass, and high pass filters used for the spectral decomposition and composition of signals. They play an important role in many modern signal processing applications such as audio and image coding. The reason for their popularity is the fact that they easily allow the extraction of spectral components of a signal while providing very efficient implementations. Since most filter banks involve various sampling rates, they are also referred to as multi-rate systems. In this experiment, the filter banks are computed by applying triangular filters on Mel-scale to the power spectrum to extract the frequency bands. The Mel-scale aims to mimic the non-linear human ear perception of sound, by being more discriminative at lower frequencies and less discriminative at higher frequencies. The approximation of Mel-frequency [30] from physical frequency can be expressed as Eqn. (4) and Eqn. (5):

$$f_{mel} = 2595\ log_{10}\left(1 + \frac{f}{700}\right) \tag{4}$$

$$f = 700\ (10^{\frac{m}{2595}} - 1) \tag{5}$$

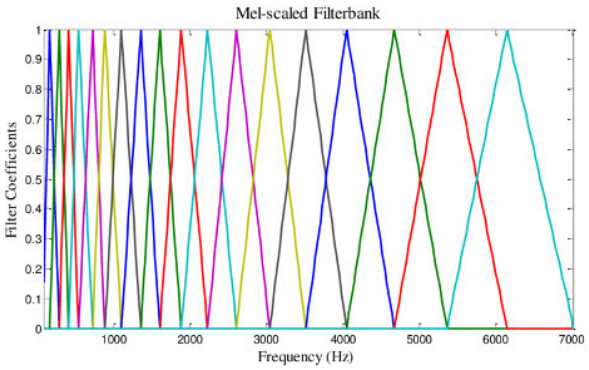


FIGURE 3. Mel-scaled filter bank with its triangular band pass frequency response used to mimic human ear perception of sound.

B. FEATURE EXTRACTION

The fundamental difficulty of speech recognition [31] is that the speech signal is highly variable due to the different speakers, speaking rates, contents and acoustic conditions. Theoretically, it should be possible to recognize speech directly from the digitized waveform. However, because of the large variability of the speech signal, it is better to perform some feature extraction that would reduce that variability. Particularly, eliminating various source of information, such as whether the sound is voiced or unvoiced and, if voiced, it eliminates the effect of the periodicity or pitch, amplitude of excitation signal and fundamental frequency etc.

Choosing which features to extract from speech is the most significant part of speaker recognition and language identification task. Some popular acoustic features that we have taken into consideration in this work are MFCC, LPC, GFCC, DWT along with some relatively new deep learning architecture based features like i-vector and x-vector. After experimenting with different combinations of these features, we have come to the conclusion that the combination of MFCC and LPC works best for the task of language identification. This claim is also supported in [32]–[33] where this combination has proved to be very effective in improving the performance of the model. Moreover, we have found from our experiments that this combination outperforms some new feature extraction techniques like i-vector, x-vector, fusion of DWT and MFCC feature warping and combination of MFCC with GFCC. These comparisons are discussed in detail in the subsequent sections. Due to the aforementioned reasons, we have used this combination of MFCC and LPC as the primary feature set for our work. Here is a brief overview of MFCC and LPC features.

1) MFCC BASED FEATURES

Cepstral features are computed by taking the FT of the warped logarithmic spectrum, and they contain information about the rate of changes in the different spectrum bands. The lower order coefficients contain most of the information about the overall spectral shape of the source-filter transfer function. The zero-order coefficient indicates the average power of the input signal. The first-order coefficient represents the distribution of spectral energy between low and high frequencies. Even though higher order coefficients represent increasing levels of spectral details, depending on the sampling rate and estimation method, 12 to 20 cepstral coefficients are typically optimal for speech analysis. Selecting a large number of cepstral coefficients results in more complexity in the models. For example, if we intend to model a speech signal by a GMM and a large number of cepstral coefficients is used, we typically need more data in order to accurately estimate the parameters of the GMM.

The idea of MFCC [34] is to convert audio in time domain into frequency domain so that we can understand all the information present in speech signals. Every sound signal consists of a set of distinct frequency components of variable intensities which is known as its spectral envelope. Bhalke *et al.* [35] showed that MFCC feature representation of sound signals is an effective way of understanding and representing the shape of their presumed spectral envelopes. But just converting time domain signals into frequency domain is not very optimal. We can do more than just converting time domain signals into frequency domain signals. Our ear has cochlea which basically has more filters at low frequency and very few filters at higher frequency. This can be mimicked using Mel filters. So the idea of MFCC is to convert time domain signal into frequency domain signal by mimicking cochlea function using Mel filters. Mukherjee *et al.* [36] extracted 19 MFCC features (1st level) for every frame with the aid of Mel Filter bank [37] and the power spectrum of the frames. Therefore, MFCC is nothing but a Discrete Cosine Transform (DCT) of a real logarithm of the short-term energy displayed on the Mel-frequency scale [38]. MFCC feature is used in a variety of industries like identifying airline reservation, numbers spoken into a telephone and voice recognition system for security purpose and is finding increased uses in music information retrieval applications such as genre classification, audio similarity measures, etc [39].

For the computation of MFCC, the first step is windowing the speech signal to split the speech signal into frames of 25ms with an overlap of 10ms. Each frame is then multiplied with a Hamming window. After windowing, FFT is applied to find the power spectrum of each frame. The entire frequency range is divided into 'n' Mel filter banks, which is also the number of coefficients we want. Now the filter bank energies are calculated by multiplying each filter bank with the power spectrum and adding up the coefficients. We take the logarithm of these 'n' energies and compute its DCT to get the final MFCCs [40]. The formula used to calculate the mels for any frequency [41], [42] given by Eqn. (5).

The MFCCs are calculated using Eqn. (6):

$$\widehat{C_n} = \sum_{n=1}^{k} (log\ \widehat{S_k}) cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{k}\right] \tag{6}$$

where k is the number of mel cepstrum coefficients, $\widehat{S_k}$ is the output of filter-bank and $\widehat{C_n}$ is the final MFCC coefficients.

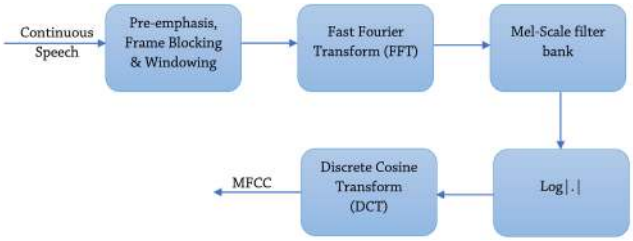


FIGURE 4. Steps involved in MFCC feature extraction.

The block diagram of the MFCC processor can be seen in Figure (4). It summarizes all the processes and steps taken to obtain the needed coefficients.

2) LPC BASED FEATURES

LPC is one of the most powerful speech analysis techniques and is a useful method for encoding quality speech at a low bit rate. The basic idea behind linear predictive analysis is that a specific speech sample at the current time can be approximated as a linear combination of past speech samples. The most important aspect of LPC is a linear predictive filter which allows the value of the next sample to be determined by a linear combination of its previous samples. LPC starts with the assumption that speech signal is produced by the glottis at the end of tube which is characterized by its intensity and frequency. LPC analyses the speech signal by estimating the formants, getting rid of its effects from the speech signal and estimating the concentration and frequency of the left behind residue. This process of removing the formants is called 'inverse filtering' and the remaining signal is called 'residue'. The coefficients of the difference equation characterize the formants and the numbers which describe the formants and the residue are stored. LPC is a powerful speech analysis method and it has gained fame as a formant estimation method [43].

Other features that can be deduced from LPC are Linear Predication Cepstral Coefficients (LPCC), Log Area Ratio (LAR), Reflection Coefficients (RC), Arcus Sine Coefficients (ARCSIN) [44] etc. LPC is generally used for speech reconstruction and applied in musical and electrical firms for creating mobile robots, in telephone firms, tonal analysis of violins and other string musical gadgets [45].

To understand LPCs, we must first understand the Autoregressive (AR) model of speech. Speech can be modelled as a p^{th} order AR process, where each sample is given by Eqn. (7):

$$x(n) = -\sum_{k=1}^{p} a_k x(n-k) + u(n) \tag{7}$$

Each sample at the nth instant depends on 'p' previous samples, added with a Gaussian noise u(n). This model comes from the assumption that speech signal is produced by a buzzer at the end of a tube, with occasional hissing and popping sounds.

LPC coefficients are given by α. To estimate the coefficients, we use the Yule-Walker equations which use the autocorrelation function R_l. Autocorrelation at lag l is given by Eqn. (8):

$$R(l) = \sum_{n=1}^{N} x(n)x(n-l) \tag{8}$$

The final form of the Yule-Walker equations is given by Eqn. (9) and Eqn. (10):

$$\sum_{k=1}^{p} \alpha_k R(l-k) = R(l) \tag{9}$$

$$\begin{bmatrix} R(0) & \cdots & R(p-1) \\ \vdots & \ddots & \vdots \\ R(p-1) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = - \begin{bmatrix} R(1) \\ \vdots \\ R(k) \end{bmatrix} \tag{10}$$

The final solution for α is given by Eqn. (11):

$$\alpha = -R^{-1} r \tag{11}$$

Figure (5) summarizes all the processes and steps taken to obtain the needed LPC coefficients.

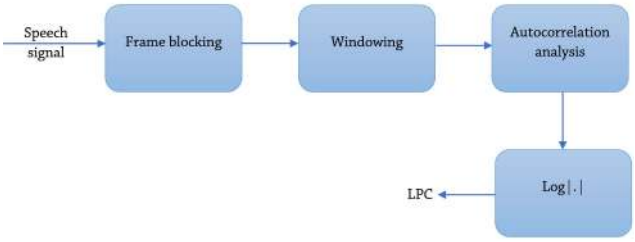


FIGURE 5. Steps involved in LPC feature extraction.

C. FEATURE SELECTION

1) BAT ALGORITHM: AN OVERVIEW

This algorithm [46] is inspired from the echolocation behavior of bats. Just like bats use natural sonar in order to find their prey, this algorithm uses a similar technique. In this algorithm, two main characteristics are adopted, viz., loudness and rate of emission of ultrasonic sound. Bats tend to decrease their loudness and increase the rate of emitted ultrasonic sound when their prey is nearer. In BA, a number of artificial bats with their own position vector X_i, velocity vector V_i and frequency vector F_i are initialized which are updated after the completion of each iteration. This exact behavior can be mathematically modelled [47] as Eqn. (12) and Eqn. (13):

$$V_i(t+1) = V_i(t) + (X_i(t) - Gbest)F_i \tag{12}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{13}$$

where Gbest is the best solution attained so far and F_i indicates the frequency of the i^{th} bat which is updated after each iteration as shown in Eqn. (14):

$$F_i = F_{min} + (F_{max} - F_{min})\beta \tag{14}$$

where β is a random number from the uniform distribution [0,1].

From Eqn. (12) and Eqn. (14), it can be concluded that different frequencies influence artificial bats to have diverse inclination to the best solution thus assuring the exploitability of the algorithm. Moreover, a random walk procedure has also been implemented to enhance the performance of the exploitability of the same as shown in Eqn. (15):

$$X_{new} = X_{old} + \varepsilon A^t \tag{15}$$

where ϵ is a random number in [-1,1] and A is the loudness of emitted sound. Since BA is a balanced combination of PSO and intensive local search, the balancing is controlled by the loudness (A) and pulse emission rate (r), which are updated using Eqn. (16) and Eqn. (17):

$$A_i(t+1) = \alpha A_i(t) \tag{16}$$

$$r_i(t+1) = r_i(0)[1 - exp(-\gamma t)] \tag{17}$$

where α is a constant analogous to the cooling factor in Simulated Annealing [48] and γ is another constant.

2) LATE ACCEPTANCE HILL CLIMBING

Hill climbing is a mathematical optimization technique which belongs to the family of local search. It is an iterative algorithm that starts with an arbitrary solution to a problem, then attempts to find a better solution by making an incremental change to the solution. If the change produces a better solution, another incremental change is made to the new solution, and so on until no further improvements can be found. Generally, this algorithm has the following advantages: easy to implement and computational time is rather small. In Hill Climbing, only one candidate which has the same or better cost function than the current candidate is accepted. To take into consideration limited acceptance of worsening moves, LAHC [49] takes the control parameter in the acceptance condition from the search history. In LAHC, a candidate solution is not only compared with the current one but also with that solution that was current several iterations before. It employs a list where the history of previous values of current cost function is stored. Now when a candidate is selected, its cost is calculated and then compared with the last element of the list. If the cost is better, then the candidate is accepted and the list is updated by adding this new current cost to the beginning of the list and removing the last element from the list. To summarize, LAHC chooses a solution immediately if its cost function is better and at the same time stores a list of worse performing solutions in order to improve them into better ones. The pseudo code of LAHC is provided in Algorithm (1).

Algorithm 1 Pseudo Code for LAHC

```
Produce an initial solution s
Calculate the initial cost function C(s)
Specify the length of the history list (L_h)
Initialize I = 0 and I_idle = 0
for (i < L_h) do
    Calculate the cost of the candidates
    Update cost in the fitness list f_v:=C(s)
end for
while (I < 10000) and ( I_idle > I * 0.02) do
    Construct a candidate solution s*
    Calculate a candidate cost function C(s*)
    if C(s*) > C(s) then
        Increment the idle iteration number. (I_idle = I_idle + 1)
        Else reset the idle iteration number. ( I_idle = 0)
    end if
    Calculate the virtual beginning v = ImodL_h
    if (C(s*) < f_v) or (C(s*) < C(s)) then
        Accept the candidate solution s = s*
        Else reject the candidate s = s*
    end if
    if C(s) < f_v then
        Update the fitness array f_v := C(s)
    end if
    Increment the number of iteration
end while
```

3) PROPOSED BBA-LAHC

BA is originally proposed for solving continuous optimization problems. So, this algorithm cannot be used for the purpose of FS where a solution or an 'agent' is represented using a binary vector. Here, 1 represents that the corresponding feature is selected and 0 represents that the corresponding feature is not selected. Therefore, to solve our purpose of FS, the position of an agent must be converted to a binary vector. In BBA, the artificial bats move around the search space utilizing position and velocity vectors (or their updated position and velocity vectors) in a binary domain ("0" and "1"). This is achieved by using a transfer function which maps the data from real continuous domain to binary domain. The suitable transfer function used in this paper is the V-shaped transfer function (as shown in Figure (6)) given by Eqn. (18):

$$V\ (v_i k\ (t)) = \left|\frac{2}{\pi} arctan(\frac{\pi}{2} v_i^k(t))\right| \tag{18}$$

where $x_i^k\ (t)$ and $v_i^k(t)$ are the position and velocity vectors of the i^{th} particle at iteration t in the k^{th} dimension, and $(x_i^k(t))^{-1}$ is the complement of $x_i^k\ (t)$.

Equation (18) is implemented in order to map the velocities of the bats to the probabilities of flipping their position vectors' elements. Also, the rules of equation (19) help in updating the position vectors of the bats.

$$x_i^k\ (t+1) = \begin{cases} (x_i^k(t))^{-1} & \text{if } rand < V(v_i^k\ (t+1)) \\ x_i^k\ (t) & \text{if } rand \geq V(v_i^k\ (t+1)) \end{cases} \tag{19}$$

After updating the position of an agent in each iteration, a local search is performed using LAHC to optimize the position of the agent in order to obtain better fitness values.

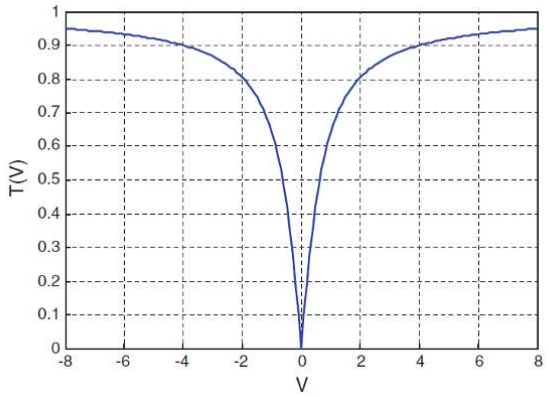


FIGURE 6. V-shaped transfer function used to convert continuous Bat algorithm search space to binary one.

This invariably eliminates the possibility of the algorithm to get stuck in a local minima and thereby increases the exploitability of the proposed hybrid BBA-LAHC FS method.

The pseudo code and flowchart for the proposed algorithm are provided in Algorithm (2) and Figure (7) respectively.

Algorithm 2 Pseudo Code for the Proposed FS Algorithm

Initialize: Bat population: $Xi(i = 1, 2, \ldots, n) =$ rand(0 or 1), Velocity $V_i = 0$ and Frequency $Q_i = 0$
Define pulse frequency F_i
Initialize pulse rates r_i and the loudness A_i
Create an empty solution matrix of dimension nXd
for $(i < n)$ **do**
  for $(i < n)$ **do**
    Fill the solution set with 0's and 1's according to random distribution
  end for
end for
Append the fitness value of each bat and find the current best
while $(t < max_iterations)$ **do**
  Adjusting frequency and updating velocities
  Calculate transfer function value using equation (18)
  Update positions using equation (19)
  if $(rand > r_i)$ **then**
    Select a solution (Gbest) among the best solutions randomly
    Change some of the dimensions of position vector with some of the dimensions of Gbest
  end if
  Generate a new solution by flying randomly
  if $(rand < A_i)$ and $(f(x_i) < f(Gbest))$ **then**
    Accept the new solutions
    Increase r_i and reduce A_i
  end if
  Rank the bats and find the current Gbest
  BestCandidate = LAHC (Gbest)
end while

V. RESULTS

A. DATASET DESCRIPTION

In this work, we have used the Indic TTS Database provided by IIT-Madras [50]. This is a special corpus of Indian languages covering 13 major languages of India. It comprises over 10,000 spoken sentences/utterances recorded by both male and female native speakers. Speech waveform files are available in.wav format along with the corresponding text. Out of the 13 Indian languages, we have used 10 most commonly spoken languages which are namely ''Bangla'', ''English'', ''Hindi'', ''Marathi'', ''Tamil'', ''Telugu'', ''Assamese'', ''Gujarati'', ''Kannada'' and ''Malayalam'' for identification purposes. We have also performed experiment on the database of 7 Indic languages [51], developed by Speech and Vision Laboratory (SVL) at IIIT-Hyderabad. This database consists of 1000 utterances for each of the 7 languages and each sentence is available as a separate audio clip in the database. These sentences span over 5000 most frequently used words in the text of the corresponding languages. The audios are recorded in a studio with a microphone connected to a zoom handy recorder. These 7 languages are ''Bangla'', ''Hindi'', ''Marathi'', ''Tamil'', ''Telugu'', ''Kannada'' and ''Malayalam''.

For our experiment, the numbers of utterances used from the Indic TTS database are: 650 for ''Bangla'', 550 for ''English'', 600 for ''Hindi'', 600 for ''Marathi'', 500 for ''Tamil'', 350 for ''Telugu'', 365 for ''Assamese'', 1000 for ''Gujarati'', 250 for ''Kannada'', and 300 for ''Malayalam''. The audio-clips for each of these 10 languages are 5 seconds long and the numbers of male and female speakers are found to be 3104 and 2061 respectively. We have allotted 80% of the total utterances to train the model and the remaining 20% for testing of the model. This implies that 4132 utterances fall in the training set and the remaining 1033 utterances fall in the testing set. Out of the 4132 utterances, 2480 utterances are by male speakers and the remaining 1652 utterances are by female speakers. Out of the 1033 utterances in the testing set, 624 utterances are by male speakers whereas the remaining 409 utterances are by female speakers. From IIIT-H database, we have used 1000 utterances for each of the 7 languages. In this case, each audio-clip is 5 seconds long, however, all the utterances are by male speakers. For this database, we have used 75% of the total utterances for training purpose, whereas the remaining 25% is allotted for testing purpose. This implies that 5250 utterances fall in the training set and the remaining 1750 utterances fall in the testing set.

B. EXPERIMENTAL SETUP

All the experiments are performed using a system equipped with a 5th generation dual-core Intel Core i5 processor clocked at a base frequency of 2.3 Ghz and 8 GB of memory.

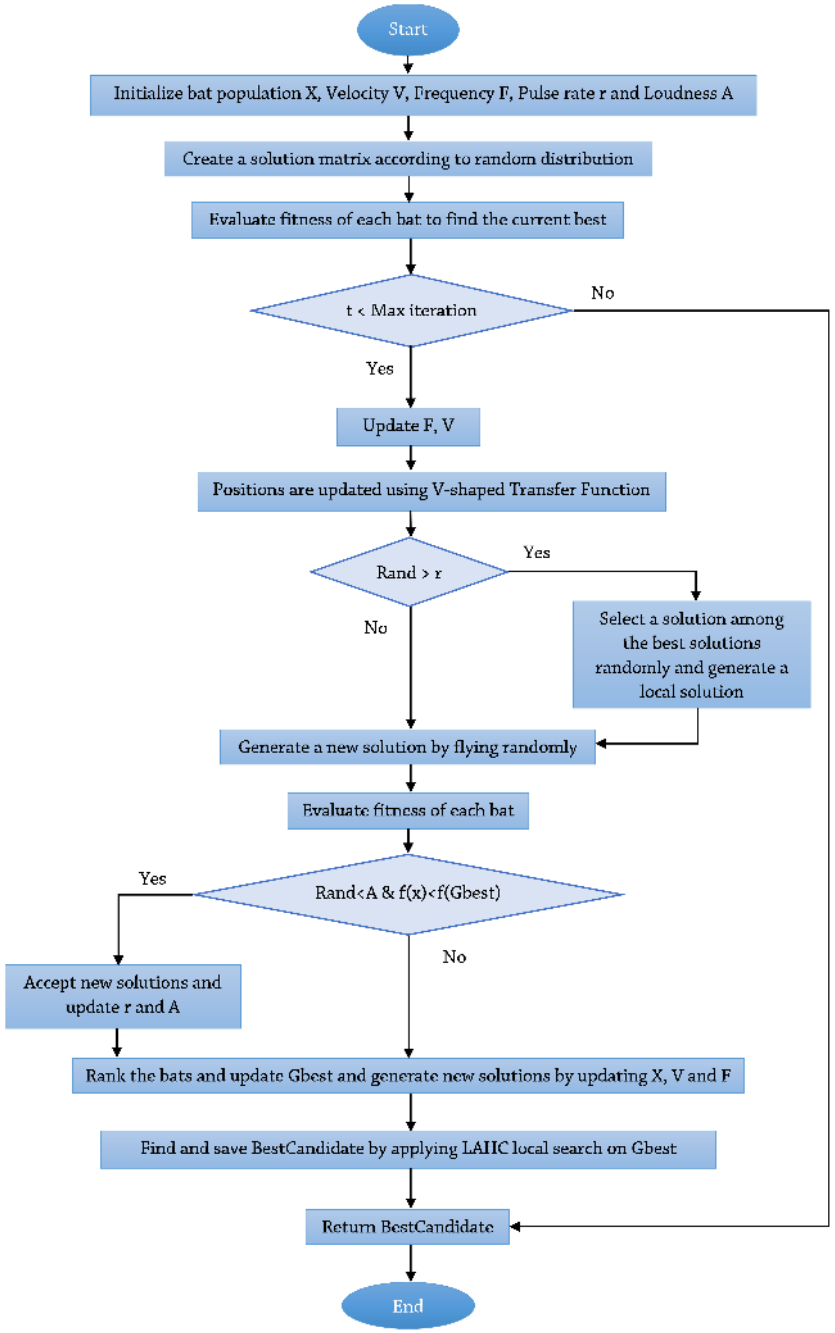


FIGURE 7. Flowchart for the proposed BBA-LAHC FS method.

The system is not equipped with an external GPU, rather, it runs on an integrated Intel Graphics card.

1) PARAMETER TUNING

In order to understand the effect of various parameters on the performance of the proposed FS algorithm, we have varied the values of one parameter while keeping the other parameters fixed and have tried to come up with optimal values of all the parameters involved. Therefore, we have evaluated the proposed BBA-LAHC based FS method for population sizes of [20, 30, 40, 50]. Figure (8) shows the effect of variation of population sizes on the classification accuracy of the proposed algorithm. From this figure, it can be noticed that the classification accuracy improves slowly with increase in the population size, reaches a maximum and then begins to decrease. Consequently, from Figure (8), we have set the population size to 30 and tried to improve the classification accuracy for both the datasets.

In order to find the value for optimum number of iterations, we have evaluated the proposed FS algorithm for iterations

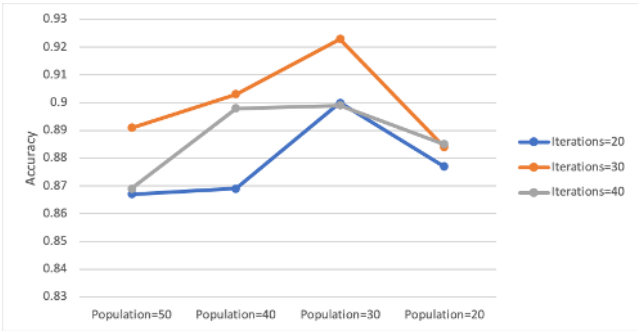


FIGURE 8. Effect of the variation of population sizes on classification accuracies of the proposed FS method on IIT-M database.

of 20, 30 and 40 while keeping all the other parameters fixed. Figure (9) shows the effect of variation of number of iterations on the classification accuracy of the proposed algorithm. A similar trend can also be seen here as the classification accuracy increases slowly, reaches a maximum and then decreases. Therefore, from Figure (9), we have set the number of iterations to 30.

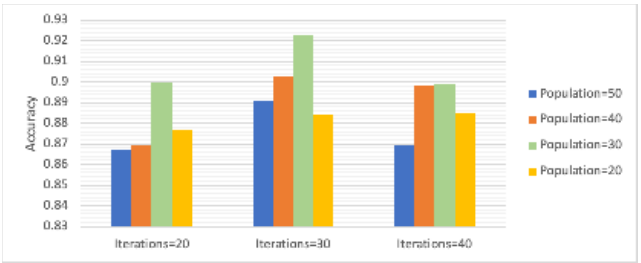


FIGURE 9. Effect of the number of iterations on classification accuracies of the proposed FS method on IIT-M database.

During the course of our experiment, we have also experimented with different values of A, r and α in the search space. As already mentioned in [52], we have also found that keeping the values of A and r to 0.9 and 0.2 respectively, the best classification result is achieved. Our experiments also come to the conclusion that the classification accuracy decreases marginally with increase in the value of α and hence, we have set the value of α to 0.7.

2) CLASSIFIERS USED

In this work, we have used 4 classifiers namely SVM, MLP, Naïve Bayes and RF for the purpose of classification. We have also experimented with the parameters of the different classifiers used. The variation of classification accuracies with the number of trees for RF classifier is presented in Figure (10) whereas the variation of classification accuracies with the number of hidden neurons for MLP classifier is presented in Figure (11). In accordance with the results obtained, we have set the number of trees to 40 for RF classifier and the number of hidden neurons to 300 for the MLP classifier. In addition to this, we have also found that using a Rectified Linear Unit (ReLU) activation function

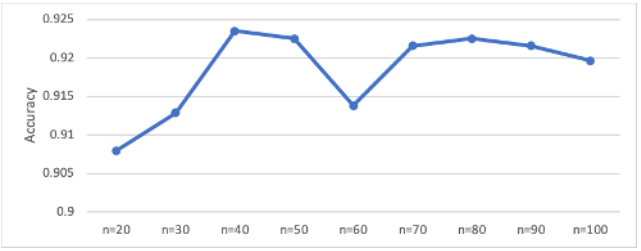


FIGURE 10. Effect of number of trees on the classification accuracies using RF classifier on IIT-M database.

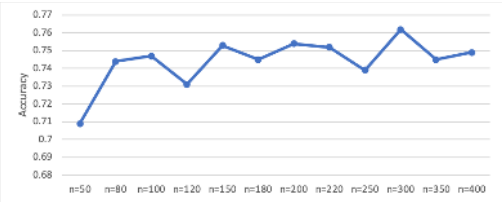


FIGURE 11. Effect of the number of hidden neurons on the classification accuracies using MLP classifier on IIT-M database.

along with the "Adam" optimizer for the MLP classifier gives the best results. In case of SVM classifier, we have used radial basis function ("rbf") kernel in order to obtain the best classification accuracy.

3) EVALUATION METRICES

The performance of our proposed BBA-LAHC based FS methodology has been measured using 4 evaluation metrices namely Classification accuracy, Precision, Recall and f1-Score. It is worth noting that a True Positive (TP) is an outcome where the model correctly predicts the positive class. Similarly, a True Negative (TN) is an outcome where the model correctly predicts the negative class. A False Positive (FP) is an outcome where the model incorrectly predicts the positive class. And a False Negative (FN) is an outcome where the model incorrectly predicts the negative class. These metrices are defined below as follows:

- **Accuracy** or **Classification accuracy** is defined as the ratio of the number of correct predictions to the total number of predictions. For this work, the accuracy in percentage is given by:

$$Accuracy = \frac{\text{No. of languages predicted correctly}}{\text{Total no. of predictions of languages}} \times 100\%$$

- **Precision** is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- **Recall** is is the proportion of correct positive classifications from the cases that are actually positive.

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

TABLE 1. Performance comparison of classification accuracies using 4 classifiers on IIT-M database.

Classifier	Accuracy without using FS	Accuracy after using BBA-LAHC
SVM	73.378%	75.992%
MLP	74.476%	75.702%
Naïve Bayes	85.299%	85.479%
RF	91.578%	**92.352%**

TABLE 2. Confusion matrix of 10 Indian languages (without FS) obtained using RF classifier on IIT-M database.

Language	Bangla	English	Hindi	Marathi	Tamil	Telugu	Assamese	Gujarati	Kannada	Malayalam
Bangla	104	1	0	2	0	0	3	1	0	0
English	19	39	14	4	3	0	4	4	0	0
Hindi	0	0	93	0	0	0	0	0	0	1
Marathi	1	1	0	90	0	0	1	1	0	0
Tamil	0	1	0	0	77	1	0	0	0	0
Telugu	0	0	0	0	1	64	0	0	0	0
Assamese	19	0	0	0	0	0	39	0	0	0
Gujarati	1	0	0	0	0	0	0	364	0	0
Kannada	1	0	0	0	1	1	0	1	39	0
Malayalam	0	0	0	0	0	0	0	0	0	37

TABLE 3. Confusion matrix of 10 Indian languages (with BBA-LAHC based FS method) obtained using RF classifier on IIT-M database.

Language	Bangla	English	Hindi	Marathi	Tamil	Telugu	Assamese	Gujarati	Kannada	Malayalam
Bangla	104	0	0	2	0	0	5	0	0	0
English	20	45	11	6	4	0	1	0	0	0
Hindi	0	0	94	0	0	0	0	0	0	0
Marathi	0	0	0	94	0	0	0	0	0	0
Tamil	0	0	0	0	79	0	0	0	0	0
Telugu	0	0	0	0	2	63	0	0	0	0
Assamese	17	1	0	3	0	0	37	0	0	0
Gujarati	0	0	0	0	0	0	0	365	0	0
Kannada	0	0	0	0	2	2	0	1	38	0
Malayalam	0	0	0	0	0	0	0	0	0	37

- **f1-Score** is the Harmonic Mean between precision and recall. The range for f1 Score is [0, 1]. It tells how precise the classifier is i.e., how many instances it classifies correctly, as well as how robust it is such that it does not miss a significant number of instances.

$$f1 - score = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

C. RESULTS OBTAINED

1) USING IIT-M DATABASE

The classification accuracies obtained using both original as well as optimal feature sets using all these four classifiers are presented in Table (1).

Primary analysis of our results reveals that higher accuracies are obtained for that dataset where we have used FS technique compared to its other counterpart. Out of the 4 classifiers used, the RF classifier achieves highest accuracy in both the cases, the accuracies being **91.578%** for the raw feature set and **92.352%** for the selected feature set respectively, thereby increasing the accuracy by an amount of **0.774%**. From the results, it can also be observed that the classification accuracy also increases significantly in case of selected feature set using both SVM and MLP classifiers. This result illustrates the robustness of this algorithm in this field.

The confusion matrix obtained without using any FS algorithm is presented in Table (2) and the one obtained by using BBA-LAHC based FS algorithm is presented in Table (3). The class-wise performance based on different evaluation metrics obtained from the raw feature set is juxtaposed with that from the selected feature set using BBA-LAHC based FS method in Table (4). The RF classifier is used for the classification purpose in both scenarios.

From Table (3), it can be seen that out of the 10 given languages, 5 languages namely "Malayalam", "Gujarati", "Tamil", "Marathi" and "Hindi" are identified with exactly **100%** accuracy when our proposed FS algorithm is used in contrary to just **2** languages being accurately identified when no FS algorithm is being used. Additionally, **2** of the remaining 5 languages namely "Kannada" and "Telugu" are identified with an accuracy well over **95%**. It is also found that after using our proposed FS algorithm, the classification accuracy in identifying "Bangla" language remains the same although the number of misclassified instances decreases from **4** to **2**. The same trend can also be seen is case of "English" where the number of misclassified instances decreases from **6** to **5**. Among all the 10 languages in this

TABLE 4. Accuracy report of 10 Indian languages obtained without FS and with BBA-LAHC based FS method using RF classifier on IIT-M database.

	Without FS			With BBA-LAHC		
Language	**Precision**	**Recall**	**f1-Score**	**Precision**	**Recall**	**f1-Score**
Bangla	0.7172	0.9369	0.8125	0.7324	0.9369	0.8221
English	0.9286	0.4484	0.6047	0.9783	0.5172	0.6767
Hindi	0.8692	0.9894	0.9254	0.8952	1.0000	0.9447
Marathi	0.9375	0.9574	0.9474	0.8942	0.9894	0.9394
Tamil	0.9390	0.9747	0.9565	0.9070	0.9873	0.9455
Telegu	0.9697	0.9846	0.9771	0.9545	0.9692	0.9618
Assamese	0.8298	0.6724	0.7429	0.8605	0.6379	0.7327
Gujarati	0.9811	0.9973	0.9891	0.9973	1.0000	0.9986
Kannada	1.0000	0.9070	0.9512	1.0000	0.8837	0.9383
Malayalam	0.9737	1.0000	0.9867	1.0000	1.0000	1.0000
Accuracy			**0.9158**			**0.9235**
Macro Average	**0.9146**	**0.8868**	**0.8893**	**0.9219**	**0.8922**	**0.8960**
Weighted Average	**0.9223**	**0.9158**	**0.9094**	**0.9315**	**0.9235**	**0.9185**

TABLE 5. Confusion matrix of 8 Indian languages (with BBA-LAHC based FS method) obtained using RF classifier on IIT-M database.

Language	Bangla	Hindi	Marathi	Tamil	Telugu	Gujarati	Kannada	Malayalam
Bangla	108	0	1	0	0	0	0	0
Hindi	0	91	0	0	0	0	0	0
Marathi	0	0	101	0	0	0	0	0
Tamil	0	0	0	87	0	0	0	0
Telugu	0	0	0	1	59	0	0	0
Gujarati	0	0	0	0	0	368	0	0
Kannada	0	0	0	1	1	0	37	0
Malayalam	0	2	0	0	0	0	0	33

dataset, it is found that ''Malayalam'' is commonly identified with an **100%** accuracy in both the cases.

It is interesting to note that some of the audio clips of ''Assamese'' language are misclassified as ''Bangla'' language and vice-versa. This is because both these languages share a lot of similarities. ''Assamese'' share a similar accent and dialect with ''Bangla'' and in fact, there are fair amounts of words that are almost identical in both the languages. Also, after analyzing the audio clips, it has been found the presence of commonly used ''English'' phrases in audio clips of other languages has mainly aided to the confusion, and has therefore resulted in misclassification.

Therefore, in order to investigate more, we have further experimented with 8 Indic languages by omitting ''English'' and ''Assamese'' audio features from the dataset. This has resulted in a much higher classification accuracy of **99.33%** using the RF classifier. The confusion matrix as well as language-wise classification accuracy for the same are presented in Table (5) and Table (6) respectively.

From both Table (4) and Table (6), it is evident that without the presence of ''Assamese'' language, the precision in identifying ''Bangla'' language increases from **73.24%** to **100%**. The same holds true for the remaining 7 languages as well, as all the languages can be identified with near perfect precision in absence of both ''English'' and ''Assamese'' languages.

It is worth noting that the total computation time required to evaluate our proposed BBA-LAHC based FS algorithm on this dataset is about 158 minutes. Once the optimal feature set consisting of 972 features is obtained, the training time required is reduced significantly to a couple of minutes and testing time is reduced to about 20 seconds using the RF classifier.

TABLE 6. Language-wise detailed performance on 8 Indian languages (after using BBA-LAHC based FS method) obtained using RF classifier on IIT-M database.

Language	Precision	Recall	f1-score
Bangla	1.0000	0.9908	0.9954
Hindi	0.9785	1.0000	0.9891
Marathi	0.9902	1.0000	0.9951
Tamil	0.9775	1.0000	0.9886
Telugu	0.9833	0.9833	0.9833
Gujarati	1.0000	1.0000	1.0000
Kannada	1.0000	0.9487	0.9737
Malayalam	1.0000	0.9429	0.9706
Accuracy			**0.9933**
Macro Average	**0.9912**	**0.9832**	**0.9870**
Weighted Average	**0.9934**	**0.9933**	**0.9932**

2) USING IIIT-H DATABASE

We have also performed experiment on the database consisting of 7 Indic languages developed by the SVL, IIIT-H. The same methodology is followed for feature extraction as well as for FS purposes. Finally, the reduced set of features is fed to SVM, RF, Naïve Bayes and MLP classifiers separately, for the classification purposes. The classification accuracy as well as the confusion matrix for this database using the 4 classifiers are presented in Table (7) and Table (8) respectively whereas the language-wise corresponding accuracy report is presented in Table (9).

TABLE 7. Performance comparison of classification accuracies using 4 classifiers on IIIT-H database.

Classifier	Accuracy without using FS	Accuracy after using BBA-LAHC FS method
SVM	93.8239%	99.8857%
MLP	91.6667%	99.9429%
Naïve Bayes	92.5369%	97.4857%
RF	94.0167%	**100%**

TABLE 8. Confusion matrix of 7 Indian languages (with BBA-LAHC based FS method) obtained using RF classifier on IIIT-H database.

Language	Bangla	Hindi	Marathi	Tamil	Telugu	Kannada	Malayalam
Bangla	250	0	0	0	0	0	0
Hindi	0	238	0	0	0	0	0
Marathi	0	0	273	0	0	0	0
Tamil	0	0	0	250	0	0	0
Telugu	0	0	0	0	252	0	0
Kannada	0	0	0	0	0	255	0
Malayalam	0	0	0	0	0	0	232

TABLE 9. Accuracy report of 7 Indian languages (using BBA-LAHC based FS method) obtained using RF classifier on IIIT-H database.

Language	Precision	Recall	f1-score
Bangla	1.0000	1.0000	1.0000
Hindi	1.0000	1.0000	1.0000
Marathi	1.0000	1.0000	1.0000
Tamil	1.0000	1.0000	1.0000
Telugu	1.0000	1.0000	1.0000
Kannada	1.0000	1.0000	1.0000
Malayalam	1.0000	1.0000	1.0000
Accuracy			**1.0000**
Macro Average	**1.0000**	**1.0000**	**1.0000**
Weighted Average	**1.0000**	**1.0000**	**1.0000**

It is clear from Table (7) that BBA-LAHC has again performed very well in identifying all the languages present in the database. It is worth noting that the accuracy improvement of the model after using BBA-LAHC FS is significantly higher for the case of IIIT-H database as compared to IIT-M database. An accuracy increase of 5% or more, is observed for all the classifiers when FS is used, which in turn, adds strength to our claim that BBA-LAHC can be considered as effective algorithm in the field of spoken language identification from audio signals.

Even though the number of languages in this database is less than that of IIT-M, the number of utterances per language is much more. Therefore, the computation time required to evaluate BBA-LAHC on this database is 182 minutes. After the optimal feature set consisting of 1141 features is obtained, the training time is significantly reduced to around 2 minutes and testing time under a 10 seconds using the RF classifier.

VI. COMPARATIVE STUDY

A. WITH RECENT FEATURE EXTRACTION TECHNIQUES

In order add diversity to our experiments, we have also considered some new feature extraction techniques like i-vector [2], x-vector [3], a fusion of DWT and MFCC features [4], and combination of MFCC and GFCC [5].

For extracting i-vector, the speech segments are first modelled using a GMM-UBM system. The parameters of GMM are normally adapted from a previously trained UBM by applying maximum a posteriori (MAP) adaptation, whereas UBM is simply a single GMM trained with substantial amount of data from all the Indic language classes at hand. For our MFCC based i-vector framework, 20 cepstral coefficients are calculated and augmented with 20 deltas and 20 delta-deltas. After finishing the post-processing and finding i-vectors of the model, we use PLDA scoring for classification [53]. The number of Gaussian mixtures and dimension of total variability matrix (T) to train the UBM are tuned to get better results. Table (10) depicts accuracy of MFCC features based i-vector framework.

The x-vector system consists of a feed forward Deep Neural Network (DNN) that maps variable-length speech segments to embeddings that we call x-vectors. The x-vector networks are divided into three parts: an encoder network, which extracts acoustic features (here, MFCC), a pooling layer for producing a single vector per utterance and finally, a feed forward classification network for evaluating speaker class posteriors. Once extracted, the x-vectors are classified by the discriminatively trained Gaussian classifier. In our experiment, the network is implemented using the Kaldi Toolkit [54]. The standard time-delay neural network (TDNN) as described in [55], is employed, which

TABLE 10. Classification accuracy of MFCC feature based i-vector framework on IIT-M and IIIT-H databases.

Database	No. of Gaussian Mixture	T matrix Size	Accuracy (in %)	Computation time(in minutes)
Indic TTS, IIT-Madras	512	300	75.92	468
IIIT-Hyderabad	512	300	89.34	516

TABLE 11. Classification accuracy of MFCC feature based x-vector framework on IIT-M and IIIT-H databases.

Database	Accuracy (in %)	Computation time(in minutes)
Indic TTS, IIT-Madras	83.14	432
IIIT-Hyderabad	93.72	498

TABLE 12. Accuracy comparison between fusion of MFCC and DWT feature warping technique with combination of MFCC and GFCC features on IIT-M and IIIT-H databases.

Database	Features	Accuracy (in %)	Computation time (in minutes)
Indic TTS, IIT-Madras	MFCC+DWT (FW)	74.29	193
	MFCC+GFCC	91.06	205
IIIT-Hyderabad	MFCC+DWT (FC)	83.14	228
	MFCC+GFCC	93.60	237

TABLE 13. Class-wise performance based on different evaluation metrices of the proposed BBA-LAHC algorithm with 7 other meta-heuristics FS algorithms using RF classifier on IIT-M database.

	BBA		BCS		BDFA		BFFA		BGA		BGSA		BPSO		BBA-LAHC	
Language	Precision	f1 - Score	Precision	f1 - Score	Precision	f1 - Score	Precision	f1 - Score	Precision	f1 - Score	Precision	f1 - Score	Precision	f1 - Score	Precision	f1 - Score
Bangla	0.68	0.78	0.72	0.82	0.72	0.81	0.74	0.82	0.73	0.82	0.73	0.81	0.69	0.79	0.74	0.82
English	0.82	0.64	0.86	0.63	0.83	0.62	0.87	0.71	0.88	0.55	0.78	0.60	0.81	060	0.98	0.68
Hindi	0.86	0.92	0.90	0.94	0.88	0.94	0.89	0.93	0.88	0.93	0.88	0.93	0.86	0.92	0.90	0.94
Marathi	0.95	0.94	0.98	0.95	0.93	0.89	0.93	0.95	0.93	0.94	0.92	0.95	0.89	0.91	0.89	0.94
Tamil	0.89	0.93	0.90	0.93	0.90	0.92	0.89	0.93	0.94	0.94	0.90	0.93	0.88	0.89	0.91	0.95
Telugu	0.97	0.95	0.96	0.97	0.94	0.95	0.94	0.93	0.91	0.95	0.93	0.94	0.90	0.92	0.95	0.96
Assamese	0.87	0.69	0.82	0.66	0.85	0.75	0.88	0.73	0.87	0.78	0.90	0.76	0.88	0.71	0.86	0.73
Gujarati	1.00	1.00	0.95	0.97	0.97	0.98	0.99	0.99	0.96	0.97	0.98	0.99	0.99	1.00	1.00	1.00
Kannada	1.00	0.88	1.00	0.90	0.97	0.90	1.00	0.92	1.00	0.95	1.00	0.90	0.97	0.86	1.00	0.93
Malayalam	1.00	0.96	1.00	0.99	1.00	0.97	1.00	0.99	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00
Accuracy (in %)	90.51		90.51		90.22		91.86		90.70		90.90		89.64		**92.35**	
Computation time (in minutes)	144		149		144		147		139		141		143		**158**	

consists of 5-time delay layers and two dense layers. The embeddings are extracted after the first dense layer with a dimensionality of 512. Table (11) shows the classification accuracy and computation time required for classifying spoken languages from both IIT-M and IIIT-H databases.

Lastly, we have used a fusion of MFCC and DWT feature warping (FW) technique and a combination of MFCC and Gammatone Frequency Cepstral Coefficients (GFCCs) features. We have evaluated our proposed algorithm to reduce the feature set on both these techniques. The accuracies along with the corresponding computation time in presented in Table (12) for both the databases.

B. WITH STATE-OF-THE-ART ALGORITHMS

To establish the superiority of the proposed FS method, we have compared this algorithm with 7 state-of-the-art meta-heuristic FS algorithms, namely:

- Binary GA (BGA) [26]
- Binary PSO (BPSO) [25]
- Binary Gravitational Search Algorithm (BGSA) [56]
- Binary Cuckoo Search Algorithm (BCS) [57]
- Binary Dragonfly Algorithm (BDFA) [58]
- Binary Firefly Algorithm (BFFA) [59]
- BBA

This comparison is done on 10 Indian languages namely ''Bangla'', ''English'', ''Hindi'', ''Marathi'', ''Tamil'', ''Telugu'', ''Assamese'', ''Gujarati'', ''Kannada'' and ''Malayalam'' from the Indic TTS Database provided by IIT-M. For all of the above mentioned algorithms, we have again used the 4 classifiers namely SVM, RF, Naïve Bayes and MLP. In all these cases, RF classifier is found to outperform the remaining 3 classifiers, therefore, in the rest of the section we present our obtained results using RF classifier only.

From Table (13), it is observed that BBA-LAHC based FS method obtained the best accuracy of **92.35%** while the least accuracy is obtained in case of classification using BPSO which stands at a modest value of **89.64%**. Further it can be observed that BBA-LAHC based FS method helps in identifying 5 languages perfectly with an accuracy of **100%**, thereby, outperforming all the remaining FS algorithms for the purpose of identifying all the 10 Indian languages. The comparison (in terms of classification accuracy) of our proposed BBA-LAHC based FS method with 7 state-of-the-art meta-heuristic FS algorithms is presented in Figure (12). It is obvious from Figure (12) that the proposed FS algorithm undoubtedly performs better than all the FS algorithms mentioned above. Language-wise performance comparison of our proposed BBA-LAHC based FS method with 7 FS algorithms is also presented in Figure (13).

C. WITH PAST METHODS

We have compared our system with the one proposed by Gupta *et al.* [60] which is presented in Table (14). It is

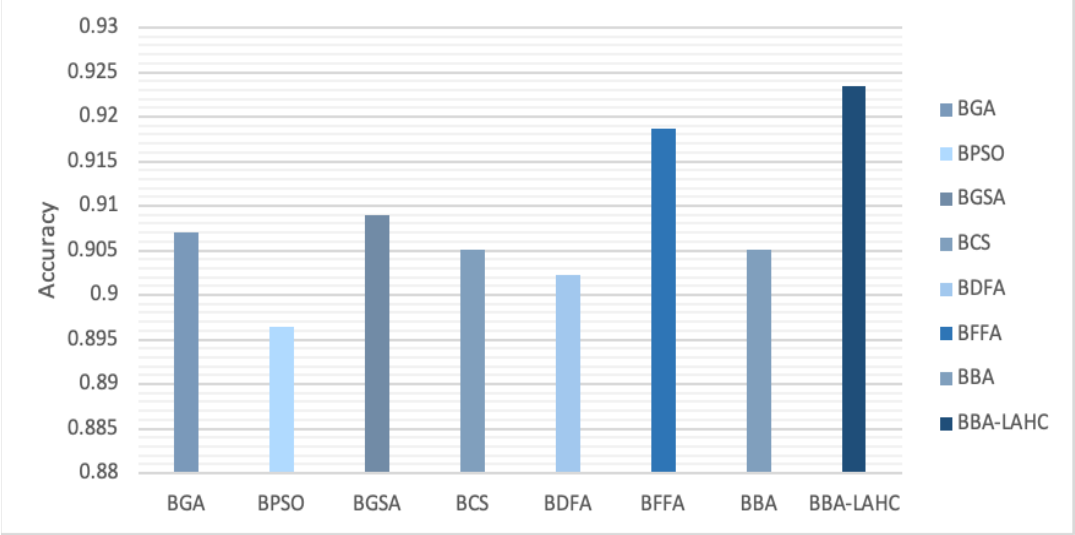


FIGURE 12. Performance comparison of the proposed BBA-LAHC based FS method with 7 other meta-heuristic FS algorithms on IIT-M database.

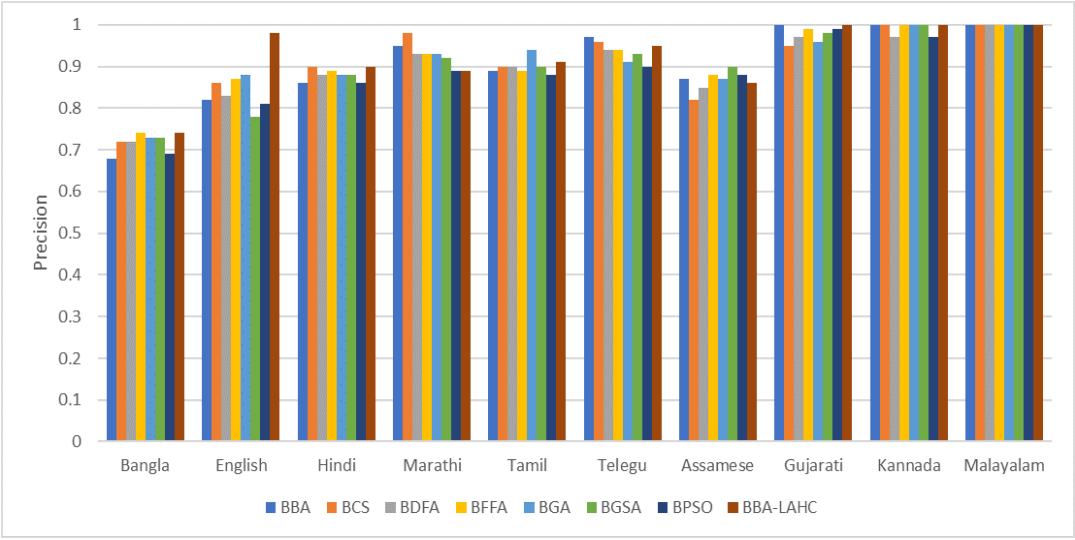


FIGURE 13. Language-wise comparison of the proposed BBA-LAHC based FS method with 7 other meta-heuristic FS algorithms on IIT-M database.

TABLE 14. Comparison of our method with past methods on the IIIT-H dataset.

Gupta et al.		**Our method**	
Classifier/Method	Accuracy without FS	Classifier/Method	Accuracy with BBA-LAHC FS method
RF with **300** trees	92.60%	RF with **40** trees	**100%**
SVM	90.60%	SVM	**99.8857%**

worth noting that they experimented with only 6 languages of the dataset and used only 2 classifiers namely SVM and RF.

We have also compared our system with the one proposed by Mukherjee *et al.* [61] and Revathi *et al.* [62]. Revathi *et al.*, in their work, had used Mel-frequency Perceptual Linear Predictive Cepstrum (MFPLPC) based clustering approach for classification of 7 Indian languages, while Mukherjee *et al.* had used a spectrogram based classification approach using Convolutional Neural Network (CNN). The comparison of our results with that obtained from their work is presented in Table (15).

TABLE 15. Comparison of our method with past methods on the IIIT-H dataset.

Revathi et al.		**Mukherjee et al.**		**Our method**	
Classifier/Method	Accuracy	Classifier/Method	Accuracy	Classifier/Method	Accuracy
MFPLPC based clustering approach	99.4%	Spectrogram based classification using CNN	99.96%	RF classifier using BBA-LAHC FS method	**100%**

TABLE 16. Language-wise comparison of the proposed system with some past methods on IIIT-H database.

Method	**Bangla**	**Hindi**	**Marathi**	**Tamil**	**Telugu**	***Kannada***	**Malayalam**	**Total**
Revathi et al.	99.78	100	99.22	98.89	99.44	98.67	99.67	99.38
Gupta et al.	93.86	90.02	95.01	91.54	91.67	-	92.05	92.60
Mukherjee et al.	100	100	99.90	100	100	100	99.80	99.96
Our method	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

The language-wise performance of our system is also compared with the ones presented by Revathi *et al.*, Gupta *et al.*, and Mukherjee *et al.* respectively and is presented in Table (16).

In all the above cases, we find that our proposed system obtains the best results.

VII. CONCLUSION

Over the years, researchers have proposed several methods in the field of spoken language identification. Among these, most of it have mainly focused on either developing new feature extraction algorithms or building robust classification models making use of state-of-the-art machine learning or deep learning architectures. However, not much research has been done on how to optimize the feature set so that the training time as well as the hardware requirement can be sufficiently reduced. To this end, in this paper, we explore a different approach to classify Indian languages from speech signals based on MFCC and LPC features by applying a new hybrid nature-inspired FS algorithm. The proposed algorithm works by incorporating the BBA with LAHC and it is found to outperform many state-of-the-art standard meta-heuristic FS algorithms. Apart from using MFCC and LPC, we have also considered some recent feature extraction techniques like i-vector, x-vector, a fusion of DWT and MFCC feature warping, and combination of MFCC and GFCC on both IIT-M and IIIT-H databases. Our proposed work is found to have outperformed these approaches as well. We have also compared our system with some past methods and it produces results with minimal error. Our main goal in this paper is to give a new approach to spoken language identification by applying a new FS algorithm prior to the task of classification. Like any new algorithm, this too is open to further intensive study for performance enhancement. This hybridization of two optimization algorithms provides tremendous scope for investigation and research, some of which include:

- Study of the effect of different transfer functions and cost functions on the performance of BBA-LAHC.
- Application of this algorithm to identify not only these 10 languages but also other languages which we have not considered here.

It is to be noted that since we have used databases from open sources, therefore, discussion on the long-term and short-term dependencies as well as time dependent variations in the speech signals is beyond the scope of this work. However, more research can be done considering these aspects and we need to develop database accordingly. We are already making progress in developing our own speech corpus which takes into consideration all the aforementioned factors and our future works will definitely have a comprehensive overview of the real-time applications of our proposed BBA-LAHC based FS algorithm keeping these factors in mind.

To the best of our knowledge, this is the first work which primarily focuses on implementing an FS algorithm for classifying Indian speech languages. We hope that our work would certainly help in bridging the digital divide among various native language speaking communities.

REFERENCES

[1] A. M. Ahmad, S. Ismail, and D. F. Samaon, "Recurrent neural network with backpropagation through time for speech recognition," in *Proc. IEEE Int. Symp. Commun. Inf. Technol. (ISCIT)*, vol. 1, Oct. 2004, pp. 98–102.

[2] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Proc. 12th Annu. Conf. Int. Speech Commun. Assoc.* 2011, pp. 1–5.

[3] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using X-vectors," in *Proc. Odyssey Speaker Lang. Recognit. Workshop*, Jun. 2018, pp. 105–111.

[4] A. K. H. Al-Ali, D. Dean, B. Senadji, V. Chandran, and G. R. Naik, "Enhanced forensic speaker verification using a combination of DWT and MFCC feature warping in the presence of noise and reverberation conditions," *IEEE Access*, vol. 5, pp. 15400–15413, 2017.

[5] W. Burgos, "Gammatone and MFCC features in speaker recognition," Ph.D. dissertation, Florida Inst. Technol., Melbourne, FL, USA, 2014.

[6] S. G. Koolagudi, D. Rastogi, and K. S. Rao, "Identification of language using mel-frequency cepstral coefficients (MFCC)," *Procedia Eng.*, vol. 38, pp. 3391–3398, Sep. 2012.

[7] M. A. Zissman and E. Singer, "Automatic language identification of telephone speech messages using phoneme recognition and N-gram modeling," in *Proc. ICASSP. IEEE Int. Conf. Acoust., Speech Signal Process.*, vol. 1, Apr. 1994, pp. I–305.

[8] Z. Tang, D. Wang, Y. Chen, Y. Shi, and L. Li, "Phone-aware neural language identification," in *Proc. 20th Conf. Oriental Chapter Int. Coordinating Committee Speech Databases Speech I/O Syst. Assessment (O-COCOSDA)*, Nov. 2017, pp. 1–6.

[9] T. S. Gunawan, R. Husain, and M. Kartiwi, "Development of language identification system using MFCC and vector quantization," in *Proc. IEEE 4th Int. Conf. Smart Instrum., Meas. Appl. (ICSIMA)*, Nov. 2017, pp. 1–4.

[10] A. J. Bekker, I. Opher, I. Lapidot, and J. Goldberger, "Intra-cluster training strategy for deep learning with applications to language identification," in *Proc. IEEE 26th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Sep. 2016, pp. 1–6.
[11] H. Mukherjee, A. Dhar, S. Phadikar, and K. Roy, "RECAL—A language identification system," in *Proc. Int. Conf. Signal Process. Commun. (ICSPC)*, Jul. 2017, pp. 300–304.
[12] S. Mohanty, "Phonotactic model for spoken language identification in Indian language perspective," *Int. J. Comput. Appl.*, vol. 19, no. 9, pp. 18–24, Apr. 2011.
[13] C. Bartz, T. Herold, H. Yang, and C. Meinel, "Language identification using deep convolutional recurrent neural networks," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2017, pp. 880–889.
[14] S. S. B. Sarthak and G. Mittal, "Spoken language identification using ConvNets," in *Ambient Intelligence*, vol. 11912. Rome, Italy: Springer Nature, 2019, p. 252.
[15] H. Mukherjee, S. Das, A. Dhar, S. M. Obaidullah, K. C. Santosh, S. Phadikar, and K. Roy, "An ensemble learning-based language identification system," in *Computational Advancement in Communication Circuits and Systems*. Singapore: Springer, 2020, pp. 129–138.
[16] S. Revay and M. Teschke, "Multiclass language identification using deep learning on spectral images of audio signals," 2019, *arXiv:1905.04348*. [Online]. Available: http://arxiv.org/abs/1905.04348
[17] H. Mukherjee, S. M. Obaidullah, K. C. Santosh, S. Phadikar, and K. Roy, "A lazy learning-based language identification from speech using MFCC-2 features," *Int. J. Mach. Learn. Cybern.*, vol. 11, no. 1, pp. 1–14, Jan. 2020.
[18] H. Mukherjee, A. Dhar, S. M. Obaidullah, K. C. Santosh, S. Phadikar, and K. Roy, "Linear predictive coefficients-based feature to identify top-seven spoken languages," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 34, no. 6, 2020, Art. no. 2058006.
[19] B. Chatterjee, T. Bhattacharyya, K. K. Ghosh, P. K. Singh, Z. W. Geem, and R. Sarkar, "Late acceptance hill climbing based social ski driver algorithm for feature selection," *IEEE Access*, vol. 8, pp. 75393–75408, 2020.
[20] K. K. Ghosh, S. Ahmed, P. K. Singh, Z. W. Geem, and R. Sarkar, "Improved binary sailfish optimizer based on adaptive β-hill climbing for feature selection," *IEEE Access*, vol. 8, pp. 83548–83560, 2020.
[21] S. Ahmed, K. K. Ghosh, P. K. Singh, Z. W. Geem, and R. Sarkar, "Hybrid of harmony search algorithm and ring theory-based evolutionary algorithm for feature selection," *IEEE Access*, vol. 8, pp. 102629–102645, 2020.
[22] K. K. Ghosh, P. K. Singh, J. Hong, Z. W. Geem, and R. Sarkar, "Binary social mimic optimization algorithm with X-Shaped transfer function for feature selection," *IEEE Access*, vol. 8, pp. 97890–97906, 2020.
[23] E. Alpaydin, *Introduction to Machine Learning*. London, U.K.: MIT Press, 2010, p. 110.
[24] S. Sarangi, M. Sahidullah, and G. Saha, "Optimization of data-driven filterbank for automatic speaker verification," *Digit. Signal Process.*, vol. 104, Sep. 2020, Art. no. 102795.
[25] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.
[26] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks*. Cham, Switzerland: Springer, 2019, pp. 43–55.
[27] J. R. Deller, J. G. Proakis, and J. H. Hansen, *Discrete-Time Processing of Speech Signals*. Hoboken, NJ, USA: Wiley, 2000.
[28] J. Benesty, M. M. Sondhi, and Y. A. Huang, "Introduction to speech processing," in *Springer Handbook of Speech Processing*. Berlin, Germany: Springer, 2008, pp. 1–4.
[29] J. W. Picone, "Signal modeling techniques in speech recognition," *Proc. IEEE*, vol. 81, no. 9, pp. 1215–1247, Sep. 1993.
[30] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *J. Acoust. Soc. Amer.*, vol. 8, no. 3, pp. 185–190, Jan. 1937.
[31] T. Ganchev, N. Fakotakis, and G. Kokkinakis, "Comparative evaluation of various MFCC implementations on the speaker verification task," in *Proc. SPECOM*, vol. 1, 2005, pp. 191–194.
[32] K. R. Aida-Zade, C. Ardil, and S. S. Rustamov, "Investigation of combined use of MFCC and LPC features in speech recognition systems," in *World Academy of Science, Engineering and Technology*, vol. 19. Istanbul, Turkey: World Academy of Science, Engineering and Technology (WASET), 2006, pp. 74–80.
[33] T. Ratanpara and N. Patel, "Singer identification using MFCC and LPC coefficients from Indian video songs," in *Emerging ICT for Bridging the Future—Proceedings of the 49th Annual Convention of the Computer Society of India (CSI)*, vol. 1. Cham, Switzerland: Springer, 2015, pp. 275–282.
[34] M. Sahidullah and G. Saha, "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition," *Speech Commun.*, vol. 54, no. 4, pp. 543–565, May 2012.
[35] D. G. Bhalke, C. B. R. Rao, and D. S. Bormane, "Automatic musical instrument classification using fractional Fourier transform based- MFCC features and counter propagation neural network," *J. Intell. Inf. Syst.*, vol. 46, no. 3, pp. 425–446, Jun. 2016.
[36] H. Mukherjee, A. Dhar, S. Phadikar, and K. Roy, "RECAL—A language identification system," in *Proc. Int. Conf. Signal Process. Commun. (ICSPC)*, Jul. 2017, pp. 300–304.
[37] S. Singhal and R. K. Dubey, "Automatic speech recognition for connected words using DTW/HMM for English/Hindi languages," in *Proc. Commun., Control Intell. Syst. (CCIS)*, Nov. 2015, pp. 199–203.
[38] K. M. Ravikumar, B. Reddy, R. Rajagopal, and H. Nagaraj, "Automatic detection of syllable repetition in read speech for objective assessment of stuttered disfluencies," *Proc. World Acad. Sci., Eng. Technol.*, vol. 36, pp. 270–273, Oct. 2008.
[39] M. Müller, *Information Retrieval for Music and Motion*. vol. 2. Heidelberg, Germany: Springer, 2007.
[40] A. M. Ahmad, S. Ismail, and D. F. Samaon, "Recurrent neural network with backpropagation through time for speech recognition," in *Proc. IEEE Int. Symp. Commun. Inf. Technol. (ISCIT)*, vol. 1, Oct. 2004, pp. 98–102.
[41] S. Chakroborty, A. Roy, and G. Saha, "Fusion of a complementary feature set with MFCC for improved closed set text-independent speaker identification," in *Proc. IEEE Int. Conf. Ind. Technol.*, Dec. 2006, pp. 387–390.
[42] M. R. Hasan, M. Jamil, and M. G. R. M. S. Rahman, "Speaker identification using mel frequency cepstral coefficients," *Variations*, vol. 1, no. 4, pp. 1–4, 2004.
[43] R. Hibare and A. Vibhute, "Feature extraction techniques in speech processing: A survey," *Int. J. Comput. Appl.*, vol. 107, no. 5, pp. 1–8, Dec. 2014.
[44] R. Kumar, R. Ranjan, S. K. Singh, R. Kala, A. Shukla, and R. Tiwari, "Multilingual speaker recognition using neural network," in *Proc. Frontiers Res. Speech Music (FRSM)*, 2009, pp. 1–8.
[45] P. K. Kurzekar, R. R. Deshmukh, V. B. Waghmare, and P. P. Shrishrimal, "A comparative study of feature extraction techniques for speech recognition system," *Int. J. Innov. Res. Sci., Eng. Technol.*, vol. 3, no. 12, pp. 18006–18016, Dec. 2014.
[46] S. Mirjalili, S. M. Mirjalili, and X.-S. Yang, "Binary bat algorithm," *Neural Comput. Appl.*, vol. 25, nos. 3–4, pp. 663–681, 2014.
[47] X.-S. Yang, "A new meta-heuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Berlin, Germany: Springer, 2010, pp. 65–74.
[48] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
[49] E. K. Burke and Y. Bykov, "The late acceptance hill-climbing heuristic," *Eur. J. Oper. Res.*, vol. 258, no. 1, pp. 70–78, Apr. 2017.
[50] A. Baby, A. L. Thomas, N. L. Nishanthi, and TTS Consortium, "Resources for Indian languages," in *Proc. Text, Speech Dialogue*, 2016, pp. 1–4.
[51] K. Prahallad, E. N. Kumar, V. Keri, S. Rajendran, and A. W. Black, "The IIIT-H Indic speech databases," in *Proc. 13th Annu. Conf. Int. Speech Commun. Assoc.*, 2012, pp. 1–4.
[52] X.-X. Ma and J.-S. Wang, "Optimized parameter settings of binary bat algorithm for solving function optimization problems," *J. Electr. Comput. Eng.*, vol. 2018, pp. 1–9, May 2018.
[53] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 7649–7653.
[54] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hanneman, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.* Piscataway, NJ, USA: IEEE Signal Processing Society, 2011, pp. 7–21.
[55] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, Aug. 2017, pp. 999–1003.

[56] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "BGSA: Binary gravitational search algorithm," *Natural Comput.*, vol. 9, no. 3, pp. 727–745, Sep. 2010.
[57] D. Rodrigues, L. A. M. Pereira, T. N. S. Almeida, J. P. Papa, A. N. Souza, C. C. O. Ramos, and X.-S. Yang, "BCS: A binary cuckoo search algorithm for feature selection," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, pp. 465–468.
[58] M. M. Mafarja, D. Eleyan, I. Jaber, A. Hammouri, and S. Mirjalili, "Binary dragonfly algorithm for feature selection," in *Proc. Int. Conf. New Trends Comput. Sci. (ICTCS)*, Oct. 2017, pp. 12–17.
[59] Y. Zhang, X.-F. Song, and D.-W. Gong, "A return-cost-based binary firefly algorithm for feature selection," *Inf. Sci.*, vols. 418–419, pp. 561–574, Dec. 2017.
[60] M. Gupta, S. S. Bharti, and S. Agarwal, "Implicit language identification system based on random forest and support vector machine for speech," in *Proc. 4th Int. Conf. Power, Control Embedded Syst. (ICPCES)*, Mar. 2017, pp. 1–6.
[61] H. Mukherjee, S. Ghosh, S. Sen, O. Sk Md, K. C. Santosh, S. Phadikar, and K. Roy, "Deep learning for spoken language identification: Can we visualize speech signal patterns?" *Neural Comput. Appl.*, vol. 31, no. 12, pp. 8483–8501, Dec. 2019.
[62] A. Revathi, C. Jeyalakshmi, and T. Muruganantham, "Perceptual features based rapid and robust language identification system for various Indian classical languages," in *Computational Vision and Bio Inspired Computing*. Cham, Switzerland: Springer, 2018, pp. 291–305.

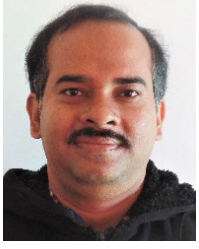
AANKIT DAS is currently pursuing the master's degree with the Institute of Radio Physics and Electronics, University of Calcutta, Kolkata, India. His research interests include machine learning, optimization, and data analytics.

SAMARPAN GUHA is currently pursuing the bachelor's degree with the Institute of Radio Physics and Electronics, University of Calcutta, Kolkata, India. His research interests include machine learning, optimization, and data analytics.

PAWAN KUMAR SINGH (Member, IEEE) received the B.Tech. degree in information technology from the West Bengal University of Technology in 2010, and the M.Tech. degree in computer science and engineering and the Ph.D. degree in engineering from Jadavpur University (JU), in 2013 and 2018, respectively. He is currently working as an Assistant Professor with the Department of Information Technology, JU. He has published more than 50 research papers in peer-reviewed journals and international conferences. His research interests include computer vision, pattern recognition, handwritten document analysis, image and video processing, feature optimization, machine learning, deep learning, and artificial intelligence. He is a member of The Institution of Engineers, India, and the Association for Computing Machinery (ACM) as well as a Life Member of the Indian Society for Technical Education (ISTE), New Delhi, and the Computer Society of India (CSI). He also received the RUSA 2.0 fellowship for pursuing his postdoctoral research at the JU in 2019.

ALI AHMADIAN (Member, IEEE) received the master's degree from Universiti Putra Malaysia (UPM) and the Ph.D. degree in early of 2014. He is currently a Fellow Researcher with the Institute of Industry Revolution 4.0, The National University of Malaysia (UKM). As a young researcher, he was dedicated to research in applied mathematics. In general, his primary mathematical focus is the development of computational methods and models for problems arising in AI, biology, physics, and engineering under fuzzy and fractional calculus (FC), and he has worked on projects related to drug delivery systems in this context, such as acid hydrolysis in palm oil frond, and carbon nanotubes dynamics, Bloch equations, and viscosity. He could successfully receive 15 national and international research grants and selected as the 1% top reviewer in the fields of mathematics and computer sciences recognized by Publons from 2017 to 2019. He has authored more than 80 research articles published in the reputed journals including IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Fuzzy Sets and Systems*, *Communications in Nonlinear Sciences and Numerical Simulations*, and *Computational Physics*. He also presented his research works in 38 international conferences held in Canada, Serbia, China, Turkey, Malaysia, and UAE. He was a Program Committee Member of a number of international conferences in fuzzy field at Japan, China, Turkey, South Korea, and Malaysia. He is also serving as a referee in more than 80 reputed international journals. He is a member of Editorial Board of *Progress in Fractional Differentiation and Applications* (Natural Sciences Publishing) and a Guest Editor of *Advances in Mechanical Engineering* (SAGE), *Symmetry* (MDPI), *Frontier in Physics* (Frontiers), and the *International Journal of Hybrid Intelligence* (Inderscience Publishers).

NORAZAK SENU is an Associate Professor with the Institute for Mathematical Research, Universiti Putra Malaysia. He has published more than 100 articles in peer-reviewed international journals. His main research interests are working on different types of differential equations and modeling real-world systems using such equations. He obtained several prizes for his research works from the Ministry of Education, Malaysia, and achieved a number of governmental grants to support his scientific works.



RAM SARKAR (Senior Member, IEEE) received the B.Tech. degree in computer science and engineering from the University of Calcutta in 2003, and the M.E. degree in computer science and engineering and the Ph.D. degree in engineering from Jadavpur University, in 2005 and 2012, respectively. He joined the Department of Computer Science and Engineering, Jadavpur University, as an Assistant Professor, in 2008, where he is currently working as an Associate Professor. He received the Fulbright-Nehru Fellowship (USIEF) for postdoctoral research from the University of Maryland, College Park, USA, from 2014 to 2015. His current research interests include image processing, pattern recognition, machine learning, and bioinformatics.