IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 20, 2020, accepted September 27, 2020, date of publication October 5, 2020, date of current version October 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3028121

Hybrid Feature Selection Method Based on Harmony Search and Naked Mole-Rat Algorithms for Spoken Language Identification From Audio Signals

SAMARPAN GUHA[1], AANKIT DAS[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, Bangi 43600, Malaysia
[4]Institute for Mathematical Research, Universiti Putra Malaysia, Seri Kembangan 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 This era is dominated by artificial intelligence and its various applications - one of which is Spoken Language Identification (S-LID) which has always been a challenging issue and an important research area in the domain of speech signal processing. This paper deals with S-LID to be used for Human-Computer Interaction (HCI) based applications by attempting to classify various languages from three multi-lingual databases namely CSS10: A Collection of Single Speaker Speech Datasets for 10 Languages, VoxForge and Indian Institute of Technology, Madras (IIT-Madras) speech corpus database by extracting their Mel-Spectrogram features and Relative Spectral Transform - Perceptual Linear Prediction (RASTA-PLP) features. A new hybrid Feature Selection (FS) algorithm have been developed using the versatile Harmony Search (HS) algorithm and a new nature-inspired algorithm called Naked Mole-Rat (NMR) algorithm to select the best subset of features and reduce the model complexity to help it train faster. This selected feature set is fed to five classifiers namely Support Vector Machine (SVM), k-Nearest Neighbor (k-NN), Multi-layer Perceptron (MLP), Naïve Bayes (NB) and Random Forest (RF). The evaluation measures used in this paper are precision, recall, f1-score, classification accuracy and number of selected features. An accuracy of 99.89% on CSS10, 98.22% on VoxForge and 99.75% on IIT-Madras speech corpus databases is achieved using RF. Furthermore, the proposed algorithm is found to outperform 15 standard meta-heuristic FS algorithms. The source code of this work is available at: https://github.com/CodeChef97dotcom/HS-NMR.git

INDEX TERMS Spoken language identification, feature selection, harmony search algorithm, naked mole-rat algorithm, RASTA-PLP features, Mel-spectrogram features.

I. INTRODUCTION

Spoken Language Identification (S-LID) is a process of identifying and classifying a digitized natural spoken language by performing computational linguistic methods on the given content or data [1]. This classification is made from a set of possible target languages [2], be it from a closed set where all possibilities are known or from an open set with unknown languages included in the test corpora. S-LID is an enabling technology that has seen a widespread use in a range of multilingual speech processing applications, such as spoken language translation [3], multilingual speech recognition [4],

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Asaduzzaman.

and spoken document retrieval [5]. In speech-based assistants also, like Apple Siri and Amazon Alexa, S-LID acts as the first step which chooses the corresponding grammar from a list of available languages for analyzing further the semantics of the languages [6]. S-LID has always been a challenging problem owing to the variations in the type of speech input and understanding how human beings comprehend and interpret speech in different conditions [7]. This makes it an important research topic in the field of speech signal processing.

Speech has various aspects associated with it that can be employed to represent the characteristics of a particular language. Unlike images, raw speech is complex and not at all suitable for feeding as an input to an S-LID model, hence the need for a good front-end arises. The task of this front-end is to convert the analog speech signal into its time-series counterpart after removing unwanted information such as background noise, etc. Next comes the task of identifying what features are to be extracted as they become extremely important in distinguishing among different languages. Features pertaining to speech are generally acoustic, prosodic or phonotactic. Prosody generally deals with the rhythm, stress, and intonation of speech, variation in syllable length, loudness, pitch, and the formant frequencies of speech sounds. These prosodic units are the actual phonetic 'spurts', or chunks of speech. Phonotactics are rules that govern permissible sequence of phonemes in speech signals, define permissible syllable structure, consonant clusters, and vowel sequences by means of phonotactical constraints. The acoustic features are the low-level features from which the prosodic and phonotactic features are derived. They deal with modelling those parameters which are obtained from digital signal processing techniques. All these features form the backbone of the S-LID model and are extremely crucial to its performance because based on these features a language can be easily identified.

Hence, it is clear from the aforementioned points that the set of features fed as input to the classifier after extracting the acoustic features from the audio signals has a significant impact on how accurately the model is able to identify the different languages under consideration. As the length of the audio files as well as the rate at which the audio signals are sampled increase, the number of features extracted for the purpose of classification also increases. However, training a S-LID model on such a large set of features requires high-end computing devices in term of hardware, thereby increasing the cost as well as computation time. Therefore, in this context, the necessity of all the features are questioned by researchers. Another speculation arises that whether a model can be developed by feeding a reduced set of features which will still be able to achieve a high classification accuracy. To this end, in this paper, a new approach to develop a feature selection (FS) [8] algorithm using a hybrid of Harmony Search (HS) and Naked Mole-Rat (NMR) algorithms for classifying spoken languages based on their Relative Spectral Transform - Perceptual Linear Prediction (RASTA-PLP) and mel-spectrogram features is explored. Through this work, it is inferred that the proposed FS algorithm is quite efficient in intelligently selecting a reduced subset of features that carry enough information about the input speech signals. The proposed FS algorithm is also found to significantly increase the classification accuracy of the S-LID model and at the same time decrease the model complexity as well as computation time.

The rest of the paper deals with the related works that are presented in Section II. The motivation behind this work is presented in Section III and proposed methodology is presented in Section IV. The experimental results are presented in Section V, and finally, the conclusion is presented in Section VI.

II. RELATED WORK

Research in S-LID from speech has a history extending back several years. Most of the early works were primitive in nature. In the work proposed by Leonard and Doddington [9], spectral features extracted from training data were scanned from regions of stability and regions of very rapid change. These regions were thought to be indicative of a specific language. Foil [10] used formant and prosodic feature-based classification technique for the purpose of language identification using recordings of noisy radio signals as a database. He achieved an accuracy of 64% with short speech segments after applying Markov modelling technique in his experiment. Goodman [11] extended Foil's work by developing a new distance metric, modifying the parameter sets, and adding a new correlation-based voicing statistic. This reduced the error rate obtained by Foil by more than 50%. Cimarusti *et al.* [12], in his work extracted 100-element Linear Predictive Coding (LPC) features from a dataset of 8 languages and then used a polynomial classifier for the purpose of classification. His work resulted in an overall classification accuracy of 84% where American English was identified with the least accuracy of 76.8% while Korean was identified with the highest accuracy of 93.4%. Zissman [13] used hidden Markov models (HMMs) to perform automatic language classification and detection of speech messages. He obtained an accuracy of 71% on the Spoken Language Library (SLL) database using jackknifing, 73% on the Rome Laboratory (RL) database using the Sanders 50/50 training/test convention and one HMM per language, and 92% on the RL database using jackknifing and one HMM per speaker. Sugiyama [14] in his work had proposed two algorithms for the purpose of language recognition. The first algorithm is based on the standard Vector Quantization (VQ) technique where every language was characterized by its own VQ codebook. The second algorithm was based on a single universal VQ codebook, for all languages, and its occurrence probability histograms. The database used consisted of 20 languages where each language consisted of 16 sentences uttered twice by 4 males and 4 females. From the results he concluded that the first algorithm had a recognition rate of 65% while the second one had a recognition rate of 80%.

TABLE 1. A brief overview of the past methods proposed for developing S-LID system.

Work Ref.	Method Used	Number of languages	Accuracy
Cimarusti et al. [12]	Polynomial classification using LPC features	8	84%
Jerry T. Foil [10]	Formant and prosodic feature-based language identification	3	64%
Marc A. Zissman [13]	HMM based language identification	20	92%
M. Sugiyama [14]	Vector Quantization Technique based language identification	20	80%
Gazeau et al. [15]	HMM based language identification	4	70%
Revay et al. [16]	log-Mel spectra based DNN approach for language identification	6	89%
Bartz et al. [6]	CRNN based language identification	6	91%
Mukherjee et al. [19]	MFCC-2 based language identification	3	98.09%

Nowadays, in an attempt to move beyond low-level spectral analysis, several attempts have been made for better and meaningful feature extraction techniques that rely heavily upon deep learning models for language identification purposes. Gazeau *et al.* [15], in their study, used Neural Network (NN), Support Vector Machine (SVM) and HMM models to recognize 4 different languages namely French, English, Spanish and German. The dataset was prepared by using voice samples from Shtooka, VoxForge and Youtube. Out of the 3 models used, HMM yielded the best accuracy of almost 70% for all the languages. Revay *et al.* [16] calculated log-Mel spectra of each audio file and then used ResNet50 [17] architecture for classifying 6 languages namely English, French, Spanish, Russian, Italian and German. They achieved an overall accuracy of 89%. Bartz *et al.* [6] identified the language of a given audio sample by using a hybrid Convolutional Recurrent Neural Network (CRNN). They had collected their datasets from speeches and statements from the European Parliament and News broadcast channels hosted on YouTube. Their work included identification of 6 languages namely English, German, French, Spanish, Russian and Mandarin Chinese. Their model achieved an overall accuracy of 91.0% on Youtube News Dataset. Sarthak *et al.* [18] used an attention based deep learning model for classifying 6 languages based on their log-Mel spectrogram features. The audio files were obtained from VoxForge dataset and the accuracy achieved was 95.4%. Mukherjee *et al.* [19] proposed a new second level Mel frequency cepstral coefficient (MFCC-2) based features to overcome the large and uneven dimensionality of MFCC [20]. This method was used to identify 3 languages namely English, Bangla and Hindi and the dataset was prepared using 12,000 utterances of numerals and 41,884 clips extracted from YouTube videos. The highest and average accuracies achieved were 98.09% and 95.54% respectively. For easy referencing, a summary of the work done on developing S-LID systems is presented in Table (1).

III. MOTIVATION AND CONTRIBUTIONS

Over the years, several techniques have been adopted for the purpose of acoustic feature extraction and language identification as mentioned above. Many S-LID systems have been developed which have made use of MFCC, LPC, Gaussian Mixture Model (GMM) [21], i-vector based features etc. Promising results are also obtained when log-Mel Spectrogram is used as a feature set. Although all these features carry some information about the audio signal, it is to be noted that as the length of the audio signal and the sampling rate increase, this number of features subsequently also increases. Now feeding all the features to the model for language identification purpose is not a feasible option as some of the features might not be relevant to the model at all. This opens up a whole new field of research where optimization of the learning process is required to enable a comprehensive capturing of the extracted features' embedded knowledge. One competent way to tackle such problem is to use meta-heuristic FS method [23-33] which intelligently selects only the relevant features without loss of any valuable information. This method assumes that this reduced set of features carries significant information about the audio signal and is enough for the model to identify the different spoken languages while maintaining a high accuracy level.

Meta-heuristic algorithms typically tend to find better solutions to an optimization problem by trial and error method. This means that there is no guarantee to find the best or the optimal solution, however, metaheuristics can often find good solutions with less computational effort than optimization algorithms, iterative methods, or simple heuristics. After exploring a myriad of meta-heuristic algorithms, HS have been chosen as the base algorithm due to the following reasons:

- HS allows the user to control the explorability and exploitability by altering the internal parameters.
- HS generates new solutions by stochastic randomization. This brings the level of its efficiency at least up to the same level of the other meta-heuristic algorithms.
- Unlike other evolutionary strategies, HS is less sensitive to the chosen parameters. This means that it is not required to perform exhaustive experiments to fine-tune the parameters to get quality solutions.
- HS allows the usage of multiple harmonic groups parallelly as it is a population-based meta-heuristic algorithm.

Although HS is a higher-level optimization strategy which works well under appropriate conditions, however, like any other meta-heuristic algorithm, there is always room for improvement. Hybridization of HS algorithm with NMR algorithm addresses some of the important issues relating to the lack of a theoretical framework for meta-heuristics to provide some analytical guidance [33]. After analyzing

the NMR algorithm, it can be understood that this particular algorithm:

- has the capability to avoid local minima
- has a quick convergence rate
- is highly consistent in finding the global minima

Keeping the above mentioned points in mind, the hybridization of HS with NMR helps in improving the efficiency for a given optimization problem, providing a better rate of convergence, or ensuring that the global optima has been reached. Some real life applications of HS are Water-Related Applications like the hydraulic simulator called EPANET, Structural Designing, Soil Stability Analysis, Information Technology Applications, Thermal and Energy Applications and so on. Since NMR is a relatively new algorithm, not much work has been done on its applications, however, as discussed in the original paper [45], the binary version of it can be used in an electroencephalogram (EEG) to reduce the total number of sensors. It can also be used for thinning of antennas and placement of nodes in wireless sensor networks. Therefore, to further dig deep into the scope of this algorithm, in this paper, a novel hybrid FS method have been proposed by taking into consideration these two meta-heuristic FS algorithms namely, HS and NMR.

IV. METHODOLOGY

In this paper, the work is presented in three main sub-sections namely Pre-processing (IV-A), Feature Extraction (IV-B) and FS (IV-C). After FS is completed, five classifiers namely SVM, k-NN, MLP, NB and RF are used for the task of classification. The flowchart for the proposed experimental setup is given in Figure (1).

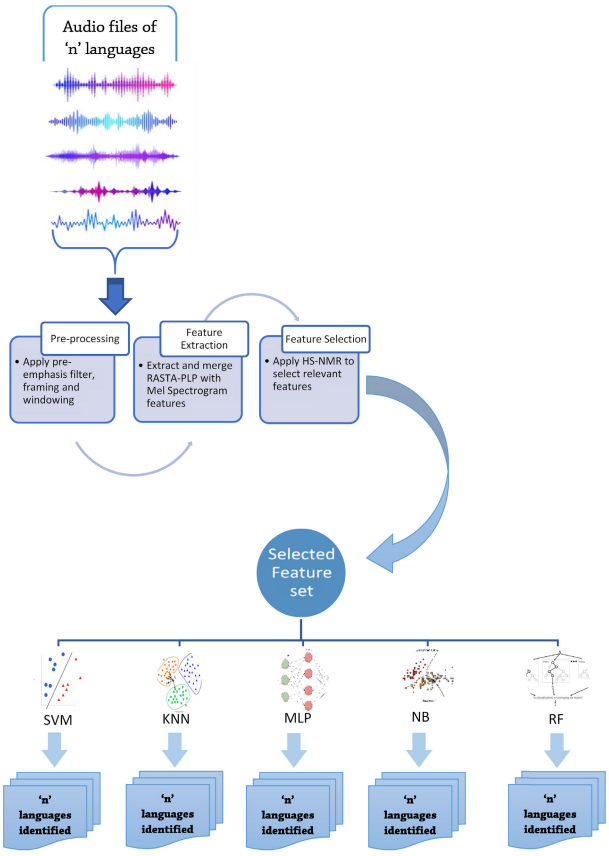


FIGURE 1. Flowchart of the proposed experimental setup used for identification of spoken languages from audio signals.

A. PRE-PROCESSING

Pre-processing of audio signals includes tasks like resampling audio files to a consistent sample rate, removing regions of silence, and trimming audio to a consistent duration. A detailed discussion on the steps involved are discussed below.

1) PRE-EMPHASIS

The pre-processing of speech signals has a great importance, because it converts the speech waveform to some type of parametric representation. Pre-emphasis [34] is a very simple signal processing method which increases the amplitude of high frequency bands and decrease the amplitudes of lower bands of a signal. In simple form it can be implemented as Eqn. (1).

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

where, $y(t)$ is the output signal and $x(t)$ is the input to the pre-emphasis filter.

Pre-emphasis reduces the dynamic range of the speech spectrum, helps in dealing with the DC offset of the signal, thus enabling to estimate the parameters of the signal more accurately. Pre-emphasis also helps in improving the signal-to-noise ratio (SNR). Typically, pre-emphasis is applied as a time-domain Finite Impulse Response (FIR) filter with one free parameter as is given by Eqn. (2).

$$P(z) = 1 - 0.68\, z^{-1} \tag{2}$$

The working of a pre-emphasis filter with filter coefficient $\alpha = 0.97$ on a speech audio signal is given in Figure (2).

2) FRAMING AND WINDOWING

Due to the differences in spectral features of phonemes, and changes in prosody due to random variations in the vocal tract, speech is a non-stationary signal. This means that the spectral features of the signal do not remain constant over its entire time span. This makes analyzing speech signals as a whole a very tedious job. Therefore, to overcome this, framing is used which decomposes the signal into a series of overlapping frames. Generally, for practical purposes, the pre-emphasized signal is blocked into 200 frames each 25ms long with 10ms frame shift. Moreover, to minimize the spectral distortions and signal mismatch when blocking the speech signal, each frame is further multiplied with a Hamming window [35], shown in Figure (3). The Hamming window is mathematically illustrated in Eqn. (3).

$$w(m) = 0.54 - 0.46cos(2\pi m/M - 1) \tag{3}$$

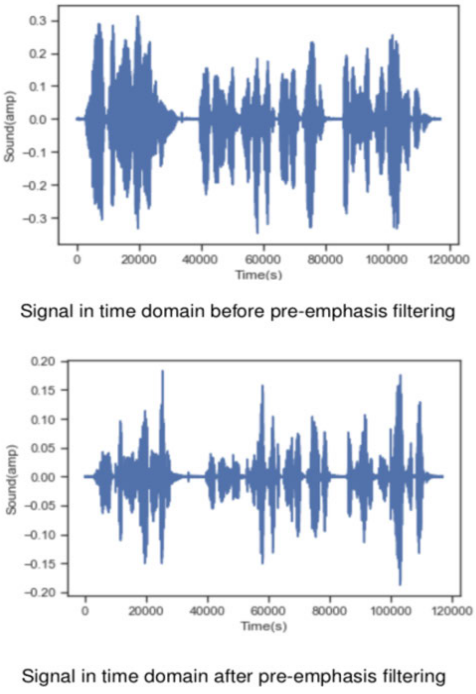


FIGURE 2. Effect of pre-emphasis filtering on audio signals.

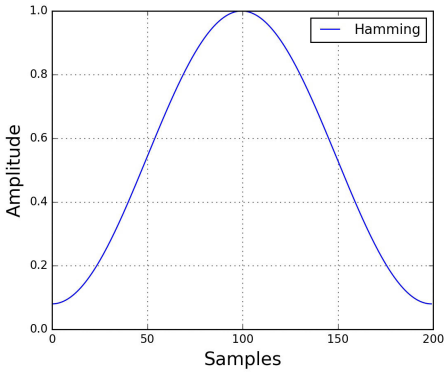


FIGURE 3. Hamming window used as a pre-processing step to minimize spectral distortion.

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

B. FEATURE EXTRACTION

Extraction of features from audio signals is a very important step in order to analyze and model the signals. Just converting the signal into its time series counterpart and doing frequency analysis does not provide much insightful knowledge about the signal, therefore, it is imperative that useful features which carry relevant information about the signal is extracted such that the model is able to pick up subtle differences between two languages and able to differentiate them properly.

Choosing which features to extract from speech is the one of the most significant part in language identification. Some popular features are: MFCC, LPC, PLP, RASTA-PLP, Line Spectral Frequencies (LSF), Discrete Wavelet Transform (DWT) etc. In this work, the primary focus is on RASTA-PLP and Mel-spectrogram features. Here is a brief overview of these features.

1) RASTA-PLP BASED FEATURES

PLP, which is a combination of both spectral analysis and linear prediction analysis, is a feature extraction method developed by Hermansky [36] in 1990. This is based on the short-term spectrum of voice signals and combines techniques like critical bands, intensity-to-loudness compression and equal loudness pre-emphasis to extract important information from speech.

Human beings perceive speech like sounds depending on the spectral difference between the current sound and the preceding sound. Humans are relatively insensitive to slow varying changes in the frequency characteristics of an audio signal and therefore, based on this evidence, suppressing these slow varying changes to model human auditory perception makes good sense. RASTA, much like human listeners, isolates not only the speech components but also the relative spectral changes in order to reduce the steady state factors (noise). Therefore, RASTA-PLP [37] is used for filtering which replaces the conventional critical-band short term spectrum in PLP with a spectral estimate in which each frequency channel is band-pass filtered with a filter that has a sharp spectral zero at zero frequency. RASTA filtering overcomes the drawbacks of PLP by smoothening out the short-term noise variations and suppressing the undesirable frequencies in speech, thereby making the technique more robust. This technique is widely used to model signals which have environmental noise and is very efficient in capturing signals with low modulation.

RASTA-PLP technique begins by calculating the critical-band power spectrum, followed by compression static non-linear transform. The time trajectory of each transformed spectral component is filtered and then this filtered speech representation is transformed through expanding static non-linear transformation. Now, equal loudness curve adjustment takes place by multiplying the loudness curve with the auditory spectrum. The value obtained is further exponentiated by a factor of 0.33 in order to truly mimic the human auditory system. Finally, the RASTA-PLP coefficients are computed from an all-pole model by consecutively performing Inverse Fourier Transform (IFT), linear predictive analysis [38] and cepstral analysis [39] on the output of the previous stage.

Figure (4) summarizes all the processes and steps taken to obtain the RASTA-PLP coefficients.

2) MEL-SPECTROGRAM BASED FEATURES

Audio signals are composed of a set of contrasting frequency components each of which is found in different proportions in it [40]. This set of components along with their frequencies is called the spectral envelope. It is this envelope that plays a

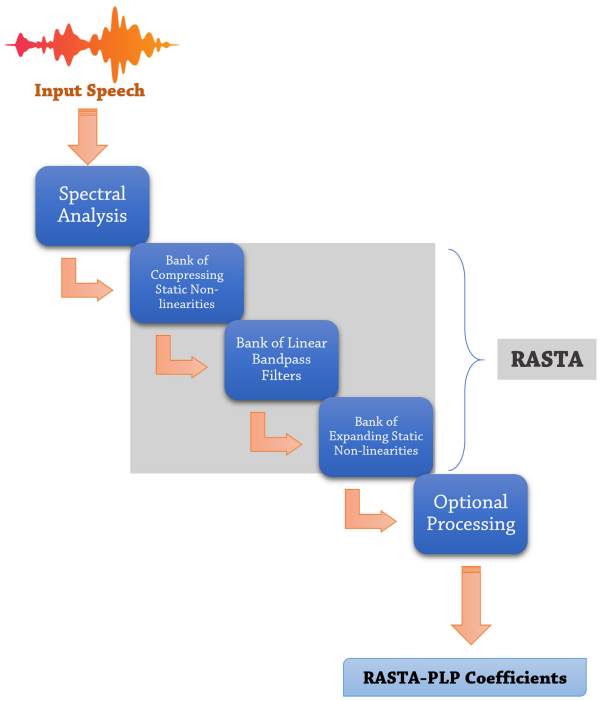


FIGURE 4. Steps involved in RASTA-PLP coefficient extraction.

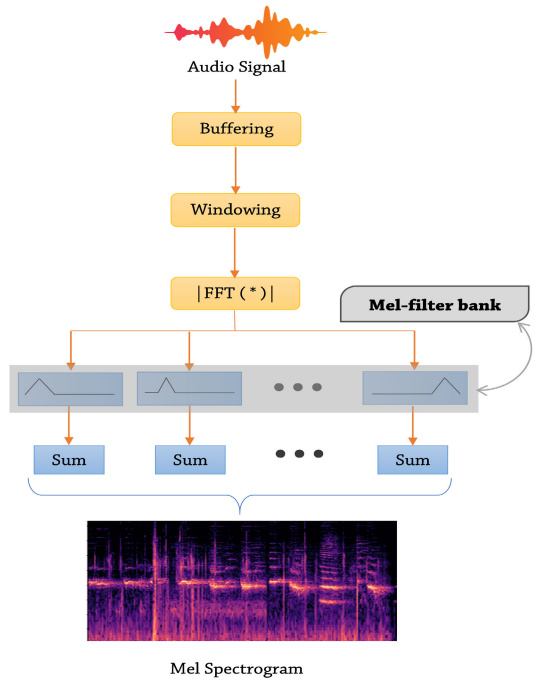


FIGURE 5. Steps involved in mel-spectrogram feature extraction.

significant role in determining what human beings actually hear, and this envelope is very well represented by spectrogram. It is a representation of frequencies changing with respect to time for given music signals. However, to model human hearing more accurately, mel-scale spectrogram have been used so that two pairs of frequencies separated by a delta in the mel scale are perceived by humans as being equidistant. The mel scale is nothing but a non-linear transformation of frequency scale based on the perception of speeches. The mel frequency scale for frequency f in Hertz is defined by Eqn. (4) and its inverse is defined by Eqn. (5).

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

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

To compute mel-spectrogram of a signal, the speech signal is first sampled into separate windows by applying framing and windowing techniques. The Fast Fourier Transform (FFT) [41] of each frame is computed to transform from time domain to frequency domain. Now a mel scale is generated by taking the entire frequency spectrum and separating it into evenly spaced frequencies. Finally, the mel-spectrogram is generated for each window by decomposing the magnitude of the signal into its components, corresponding to the frequencies in the mel scale.

Figure (5) summarizes all the processes and steps taken to obtain the mel-spectrogram coefficients.

C. FEATURE SELECTION

In this section, an exhaustive discussion of the popular HS algorithm and the relatively new NMR algorithm is done followed by the proposed hybridization of these two algorithms.

1) HS ALGORITHM: AN OVERVIEW

HS is a very popular and powerful meta-heuristic optimization algorithm, first proposed by Geem *et al.* [42] in 2001. It has gained world-wide attention due to its balanced combination of exploration and exploitation. Unlike existing heuristic methods based on natural phenomena, this algorithm is inspired from the artificial phenomenon of musical harmony; one of the most aesthetically satisfying process generated by human beings. HS algorithm is inspired by musical performance process, for example, the process of finding the better harmony by aesthetic estimation. HS mimics this action by seeking the best state or global minimum cost by objective function evaluation. One of the most important advantages of this algorithm is the ability to handle explorability and exploitability. The presence of three operators in this algorithm namely, random search, harmony memory considering rate (HMCR) and pitch adjustment rate (PAR) furthers this cause.

Although there exists many literature which claim that HS is equivalent to an evolution strategy (ES), Joong *et al.* [43] in 2016, proved the uniqueness of this algorithm. While the basic idea behind any evolution strategy lies on adaptation and evolution governed by two operators: recombination and mutation operators, HS, on the other hand, has three operators namely, HMCR (used for both exploration and exploitation), random search (used for the exploration), and PAR (used for the exploitation). Additionally, the operators in ES might be mandatory as in the case of (μ+1)-ES while in HS the

frequency of operation is controlled by the algorithm parameters - HMCR and PAR. The only similarity in HS and ES is that in both these algorithms, a new solution is generated at each iteration which can replace the worst solution in the Harmony Memory (HM) and population respectively.

Music consists of three main elements — melody, rhythm, and harmony. While the first two are typically accountable for making a piece of music memorable (the opening motif of Beethoven's Symphony No. 5), it's the third element, harmony, that can elevate a piece of music from common and predictable to challenging and sophisticated. Harmony is the composite product when individual musical voices group together to form a cohesive whole and it is analyzed as a series of chords. This method had been incorporated in the HS, and the steps in the procedure of HS as proposed by Geem *et al.* are as follows:

In Step 1, the objective function f(x), which is set up in accordance to the problem at hand, is optimized subject to the inequality constraint function g(x) and the equality constraint function h(x). The parameters are also specified in this step. These are the harmony memory size (HMS), or the number of solution vectors in the HM, HMCR, PAR, the number of decision variables (N), and the number of improvisations (NI), or stopping criterion. HM is the memory location where all the solution vectors are stored.

In Step 2, the HM matrix is filled with as many randomly generated solution vectors as the HMS as illustrated by Eqn. (6):

$$HM = \begin{bmatrix} x_1^1 & \cdots & x_N^1 \\ \vdots & \ddots & \vdots \\ x_1^{HMS} & \cdots & x_N^{HMS} \end{bmatrix} \tag{6}$$

In Step 3, a new harmony vector, $x_i' = (x_1', \ldots, x_N')$, is generated based on HMCR, PAR and random selection. This generation of new harmony is called 'improvisation' [44]. In HMCR, a new random number r_1 is generated in range [0,1]. If this randomly generated number is less than HMCR, then the first decision variable in the new vector x_{ij}^{new} is chosen randomly from the values in the current HM. The obtained decision variables from HMCR is further examined to determine if it needs pitch adjustment or not. For this reason, another random number r_2 is generated in range [0,1] and if this number is less than PAR, then the pitch adjustment decision variable is calculated as given in Eqn. (7):

$$x_{ij}^{new} = x_{ij} \pm rand(0,1) * Bandwidth\,factor \tag{7}$$

where, *Bandwidth factor* is one of the important factors for the high efficiency of the HS algorithm and can be potentially useful in adjusting convergence rate of algorithm to obtain optimal solution.

If the HCMR condition fails, then the new first decision variable in the new vector is randomly generated as illustrated by Eqn. (8):

$$x_{ij}^{new} = l_{ij} + (u_{ij} - l_{ij}) * rand\,(0,1) \tag{8}$$

Algorithm 1 Pseudo Code for HS

```
Initialize the parameters HMS, HMCR, PAR, BW and NI
Set t = 0
for (i ≤ HMS) do
    Generate initial population HMS
    Evaluate the fitness function of each harmony vector f(x_i)
end for
while (t < NI) do
    Generate a new solution x_i
    for (i ≤ HMS) do
        for (j < n) do
            if (r1 < HCMR) then
                x_ij^new = x_ij
                if (r2 < PAR) then
                    x_ij^new = x_ij ± rand(0,1) * BF
                end if
                if (x_ij^new < l_j) then
                    x_ij^new = l_j
                end if
            else
                x_ij^new = l_ij + (u_ij − l_ij) * rand (0,1)
            end if
        end for
    end for
    if (X_new < X_worst) then
        Update the HM as X_worst = X_new
    end if
    Set t = t+1
end while
Produce the best solution (harmony vector) X_new
```

where, u and l are the upper and lower bounds for the given problem respectively.

In Step 4, HM is updated. If the new harmony vector is better than the worst harmony in the HM, judged in terms of the objective function, the new harmony is included in the HM, and the existing worst harmony is excluded from the HM.

In Step 5, stopping criterion (maximum number of improvisations) is checked and if it is satisfied, the computation is terminated. Otherwise, Steps 3 and 4 are repeated. The pseudo code of the HS algorithm is provided in Algorithm (1).

2) NMR ALGORITHM: AN OVERVIEW

The naked mole-rat, also known as the sand puppy, is a burrowing rodent which are known to be eusocial mammals, the highest classification of sociality. Only one female, the queen, and a group of males called the breeders, reproduce. The other group, called the workers, are sterile and they assist in various other jobs like construction, provisioning, maintenance and defense. All the females in the worker group fight until death to become the queen and establish their

dominance, hence there can be only one queen. This colonial lifestyle of a species, following a strict division of labor and having a single female for breeding is known as Eusociality.

Judging from the hierarchical pattern of the NMR without the presence of any central authority, it can be said that they follow self-organization and division of labor, thus having a close association with Swarm Intelligence (SI). This has led the researchers to develop SI-based algorithm by mimicking the mating patterns in NMR. Aforementioned claims along with the NMR algorithm has been discussed in [45].

The rules of mating have been idealized to propose the NMR algorithm. It is divided into three phases, initialization phase, worker phase and breeder phase. Detailed discussion of each phase is given below:

1) Firstly, a uniformly distributed random population of n NMR is generated where each NMR is a D-dimensional vector in the range [1, 2, . . . , n]. Each NMR is initialized as given by Eqn. (9):

$$NMR_{i,j} = NMR_{min,j} + (NMR_{min,j} - NMR_{max,j}) * rand(0, 1) \quad (9)$$

where, i ϵ [1,2,. . . ,n], j ϵ [1,2,. . . ,D], $NMR_{i,j}$ is the i^{th} solution in the j^{th} dimension, $NMR_{min,j}$ and $NMR_{max,j}$ are the lower and upper bounds of the problem function respectively.

2) Next, the population of NMR is iterated over many cycles of the search process of worker and breeder phases. In the worker phase, the workers tend to improve their fitness based on a pre-defined fitness function so that they can upgrade themselves to the breeder group and eventually mate with the queen. It is interesting to note that the workers in this phase generate new solutions based on the local information. After calculation of the fitness, the solution is updated with the new solution if the fitness is better, otherwise the old solution is retained. The final fitness of the solution is stored after all the worker rats complete their search process. The mathematical equation governing this behavior is given by Eqn. (10):

$$w_i^{t+1} = w_i^t + \lambda * (w_j^t - w_k^t) \quad (10)$$

where, w_i^t corresponds to the i^{th} worker in the t^{th} iteration, w_i^{t+1} is the new solution, λ is the mating factor generated from the uniform distribution $U(0, 1)$ and w_j^t and w_k^t are two random solutions chosen from the worker's pool.

3) In the breeder phase, the NMR tries to update themselves too in order to be selected for mating and to stay in the breeder group. The breeder NMRs are updated according to their breeding probability (bp) with respect to the overall best (d). The bp is a random number in the range of [0,1]. If the breeders are found to perform poorly, then they are then pushed to the worker class. The breeders modify their positions according to Eqn. (11):

$$b_i^{t+1} = (1 - \lambda)b_i^t + \lambda * (d - b_i^t) \quad (11)$$

where, b_i^t corresponds to the i^{th} breeder in the t^{th} iteration, λ is a factor that controls the mating frequency of breeders and helps in identifying new breeder b_i^{t+1} in the next iteration.

Algorithm 2 Pseudo Code for NMR

```
Initialize the parameters n, B = n/5, W = n − B, bp
while (iter < MaxIter) do
  for (i < numWorkers) do
    Find the new worker solution using w_i^{t+1} = w_i^t + λ * ( w_j^t − w_k^t )
  end for
  for (j < numBreeders) do
    if (rand (0, 1) > bp) then
      Find the new breeder solution using b_i^{t+1} = (1 − λ)b_i^t + λ * ( d − b_i^t )
    end if
  end for
  Combine the new worker and breeder population
  Evaluate the population and update the itercount
end while
Save and return the final best d
```

To make the algorithm simple, it is assumed that there is only one queen and the best among the breeders mates with the queen. This algorithm first identifies the breeders and workers among the pool of NMRs. Then the best breeder and worker are selected and the fitness of the workers is updated at every iteration so that their fitness improves and they may update themselves to the breeder class and eventually mate with the queen. On the other hand, the fitness of breeders is also updated based on their breeding probability. The breeders which become sterile are pushed back to the worker class. The best breeder among the population serves as the potential solution to the problem. The pseudo code of NMR is provided in Algorithm (2).

3) PROPOSED HS-NMR

Every FS algorithm is a binary optimization problem where '1' indicates that the corresponding feature is to be selected and '0' indicates that the feature is to be discarded. The main goal of FS algorithm is to reduce the number of 1's along with increasing the classification accuracy. The concept of optimization in a binary search space is different from that applied in a continuous search space. A binary search space can be considered as a hypercube. The search agents of heuristic algorithms have to jump to nearer and farther corners of this hypercube. A jump is usually performed by flipping various numbers of bits. For most binary optimization problems, two transfers functions namely, S-shaped transfer function [46] and V-shaped transfer function [46] are majorly used which are mathematically modelled by Eqn. (12) and

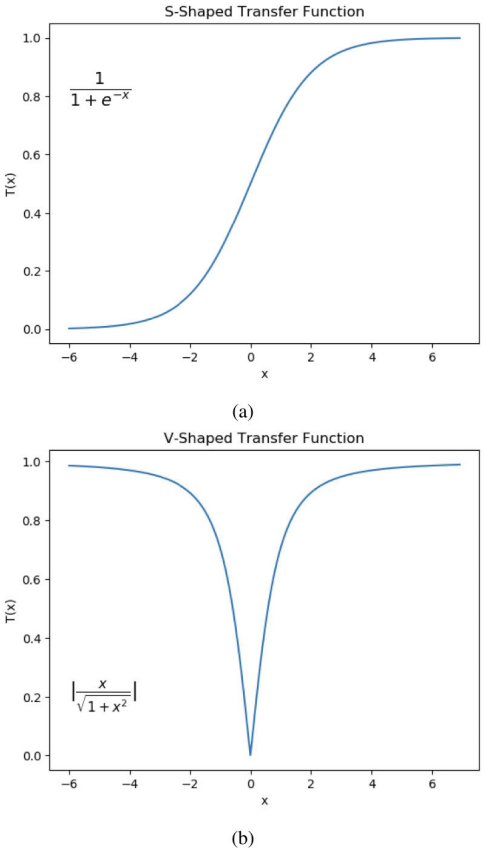


FIGURE 6. **(a) S-shaped and (b) V-shaped transfer functions used to convert continuous search space of any optimization problem into binary one.**

Eqn. (13) respectively. These transfer functions are also illustrated in Figure (6).

$$T_S(x) = \frac{1}{1+e^{x}} \tag{12}$$

$$T_V(x) = \left|\frac{x}{\sqrt{1+x^2}}\right| \tag{13}$$

It is to be noted that the V-shaped transfer function more often tends to form the complement of the variables. This mechanism promotes and guarantees changing the position of the search agents proportional to their velocities. In contrast, the S-shaped transfer function forces the search elements to take the value 0 or 1. Although the S-shaped transfer function, to some extent, hinders the possibility of the searching agents to explore towards global solution space, this drawback has been taken into account in this work by hybridizing HS with NMR which not only increases exploration of the global search space but also increases the exploitation of the same. For the purpose of comparison, experiments on both the S-shaped and V-shaped transfer functions have been done and amongst these two, the HS-NMR algorithm is found to perform marginally better with S-shaped transfer function. Moreover, the S-shaped transfer function is also found to reduce the dimension of the feature set more than the V-shaped transfer function for the problem domain. Keeping the facts in mind, S-shaped transfer function is used for the purpose of FS for the S-LID problem.

The problem of increasing classification accuracy while reducing the number of features might sound contradictory in nature. To do away with this dilemma, a fitness function has been proposed which takes into account both the driving candidates viz., the initial classification error rate and the fraction of selected features. This fitness function given by Eqn. (13):

$$\text{Fitness} = \omega * \text{f}(F') + (1-\omega) * (\text{Fraction of selected features}) \tag{14}$$

where f(F') represents classification error rate of the reduced feature set and ω acts like a weight whose value ranges between [0,1].

The dependency of HS algorithm on the values of HMCR and PAR has been discussed exhaustively in earlier sections. While HMCR mainly helps to find out favorable areas where global best may lie, thus, ensuring exploration, PAR on the other hand, helps to properly search the already discovered areas therefore ensuring exploitation. So, it might seem that HS alone can produce excellent convergence if the appropriate setting of these values is chosen, however, obtaining these ''appropriate'' values is a mammoth task in itself. Since HS is an iterative process involving some mathematical formulae and nested loops, it is seen that with increase in precision by 10% margin, the time requirement increases exponentially [47]. In order to address this problem, the hybridization of another FS algorithm with HS is proposed. The NMR is a versatile algorithm in the sense that its parameters give us the controllability over exploring and exploiting the search space. In NMR algorithm, exploration and exploitation are balanced by the worker and breeder phases. Here, the workers consist of a larger chunk of the total population (almost 80%) and, because of their large population size, perform exploration. The workers continuously update themselves in order to become a part of the breeder's group and in turn tend to explore different regions of the search space. The breeders belong to a smaller population group (about 2%) but with higher fitness than the workers and perform more intensive exploitation as they tend to improve their fitness in order to mate with the queen. The breeders tend to follow the best-known breeder and hence take small steps to improve the good solutions. Moreover, if a breeder does not improve fitness, it is pushed into the worker's group, so it avoids solutions getting stuck in local minima and of stagnation of population of breeders. This hybridization allows the HS algorithm to reduce its dependency on the initial values of HMCR and PAR to obtain global minima. The breeder population aids in the exploration process while the worker population improves the exploitation for a particular harmony.

The analysis of the algorithm shows that its worst case time complexity is:

$$\mathcal{O}(MaxIter * (X_{new} * t_{fitness} + ((numWorkers + numBreeders) * t_{fitness}) + D))$$

where, *MaxIter* is the maximum number of iterations, X_{new} is the best HM, *numWorkers* and *numBreeders* are the number of workers and breeders, $t_{fitness}$ is the time requirement for calculating the fitness value of a particular agent using a given classifier and D is the dataset dimension. The flowchart of the proposed work is illustrated in Figure (7).

V. RESULTS

In Subsection V-A, the details about the dataset used is mentioned. In Subsection V-B and Subsection V-C, the experimental setup as well as the results obtained is discussed at great length. A comparative study of the proposed method with 15 state-of-the-art FS algorithms as well as with some past methods is provided under Subsection V-D and Subsection V-F respectively. Finally, a statistical significance test is also performed, the results of which is presented in Subsection V-E.

A. DATASET DESCRIPTION

In this paper, audio files from two foreign language databases namely CSS10 [48] and VoxForge [49] and one Indian language database namely Indic TTS database [50] provided by IIT-Madras have been used. CSS10 is a collection of single speaker speech dataset for 10 languages which consists of short audio clips from LibriVox audiobooks. The large audio files from LibriVox are processed and fragmented into many small audio files. The audio processing is discussed in [48]. The 10 languages used are ''German'', ''Greek'', ''Spanish'', ''Finnish'', ''French'', ''Hungarian'', ''Japanese'', ''Dutch'', ''Russian'', ''Chinese''. Unlike CSS10, the audio files of VoxForge database are not pre-processed. VoxForge is a free speech corpus and acoustic model repository for open source speech recognition engines. People from all over the world can submit their audio samples using any electronic equipment that are available to them which makes the audio qualities vary with every language. So, out of the 17 languages that are available now, 6 languages have been used namely, ''English'', ''French'', ''German'', ''Italian'', ''Russian'', ''Spanish''. This is because the quality of these audio files are relatively better than others and additionally the length and format of these audio files are appropriate for this experiment. Lastly, the Indic TTS database provided by IIT-Madras is a special corpus of Indian languages which has over 10,000 spoken sentences/utterances recorded by both male and female native speakers covering 13 major languages. Out of these, 10 spoken languages have been used namely, ''Bangla'', ''English'', ''Hindi'', ''Marathi'', ''Tamil'', ''Telugu'', ''Assamese'', ''Gujarati'', ''Malayalam'' and ''Kannada'' for experimentation. This is because these 10 languages are the most popular and widely used languages all over India. The aforementioned audio files from all three databases are available in.wav format which makes it easier to analyze in Python 3.

For the experiment, 700 utterances ranging between 2 to 10 seconds for each language from CSS10 database have been used. Owing to the quality variations related to VoxForge database, 150 utterances have been handpicked ranging between 3 to 15 seconds for each language which are relatively better in quality than others. In case of Indic TTS database, depending on availability, the number of utterances ranges from 200 to 400 for the 10 different languages. The duration of each utterance does not exceed 10 seconds. Throughout the course of this experiment, 75% data have been used for training purposes and the remaining 25% data was allotted for testing purposes.

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

1) PARAMETER TUNING

In any meta-heuristic based FS model, choosing the right set of values for the parameters involved, is of paramount importance as these are the parameters which control the learning process and contribute immensely to the success of the model. In this work, values for two very important parameters of the proposed algorithm namely, population size and number of iterations are decided through experimentation. More specifically, to truly understand the dependence of the algorithm on these parameters, many experiments with various permutations of population size and number of iterations have been performed while keeping the other parameters constant. The algorithm have been evaluated for iterations of [10, 20, 30, 40] on all the datasets and it is observed that even though the accuracy increases significantly with an increase in number of iterations from 10 to 30, the accuracy increases only by a small margin while increasing the same from 30 to 40. However, the time complexity increases exponentially with a small increase in number of iterations. Keeping this trade-off in mind, an iteration number of 30 for the rest of the experiments is used. Additionally, the population has been varied using size from 10 to 40 and it can be concluded that a population size of 20 has achieved the best results. During the course of the experiment, different values of bp (a parameter of NMR algorithm) is also varied and from the results it can be concluded that setting its value to 0.50 has obtained best results which is in accordance with the claims of [45].

2) CLASSIFIERS USED

Five state-of-the-art classifiers have been used namely, SVM, k-NN, MLP, NB and RF for the purpose of S-LID. In order to obtain optimal accuracy for all the classifiers used, experiments have been conducted with a variety of parametric values. For all the three datasets, the variation of classification accuracies have been calculated with the number of trees for RF classifier, with number of neighbors in k-NN classifier, and with the number of hidden neurons in MLP classifier. These variations are presented in Figure (8), Figure (9), Figure (10) respectively. The final set of values for these three

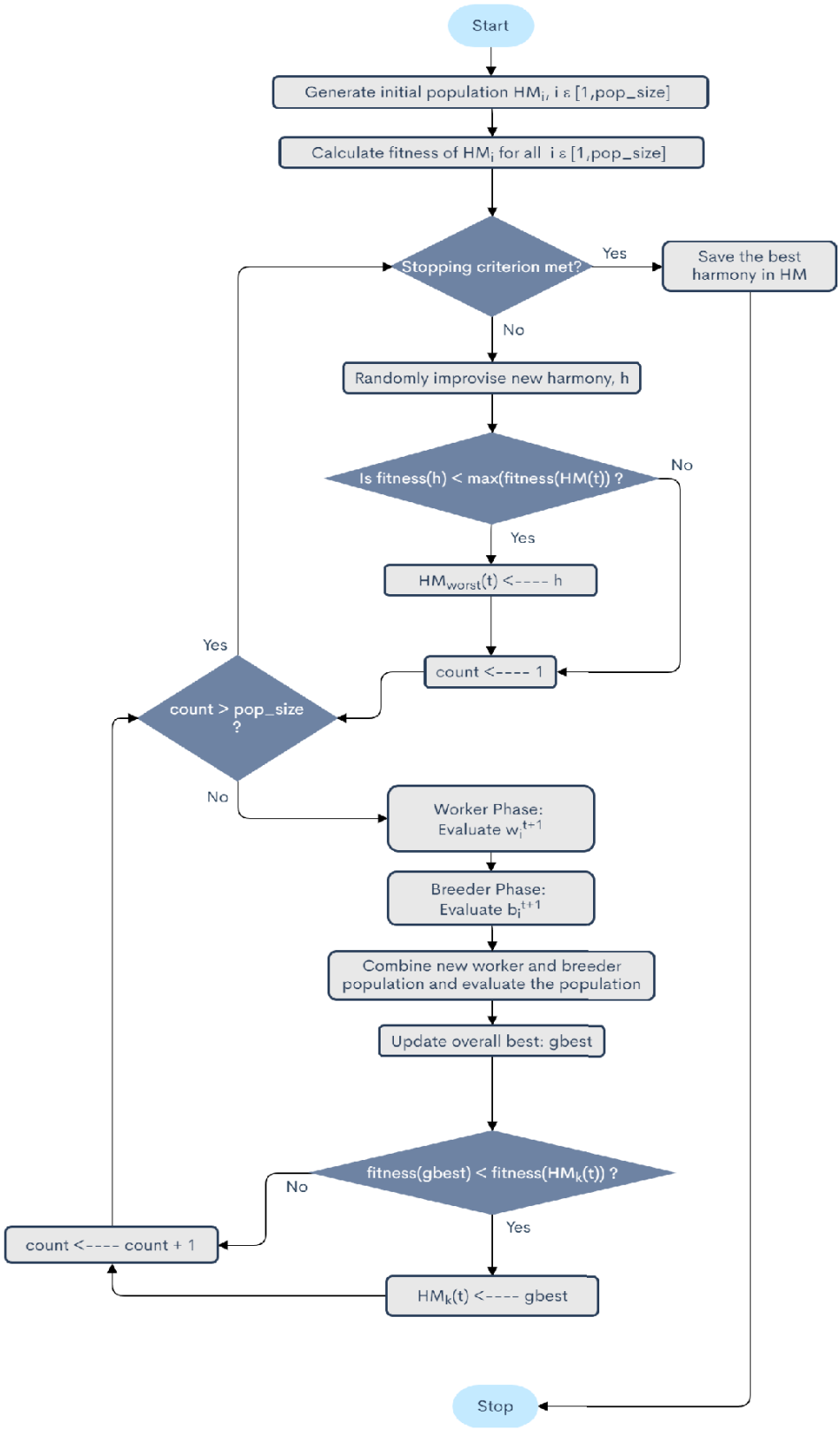


FIGURE 7. Flowchart for the proposed HS-NMR algorithm used for FS in S-LID.

classifiers for all the three datasets is presented in Table (2). In addition to this, it is also found that using a Rectified Linear Unit (ReLU) activation function along with the "Adam" optimizer for the MLP classifier gives the best results.

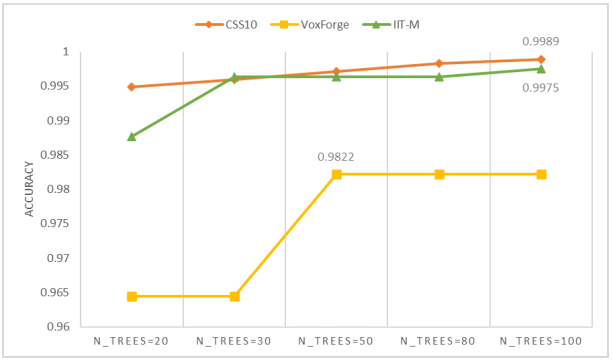


FIGURE 8. Effect of number of trees of the RF classifier on the classification accuracy over three S-LID datasets.

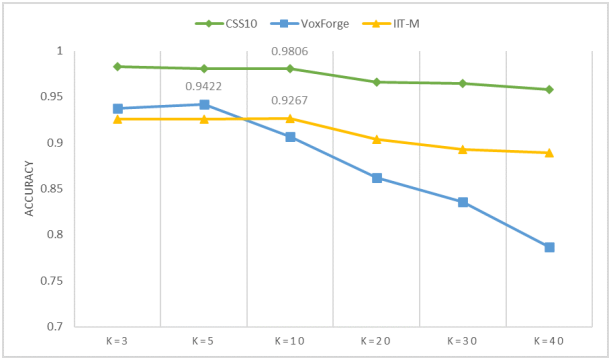


FIGURE 9. Effect of number of neighbors (k) of the k-NN classifier on the classification accuracy over three S-LID datasets.

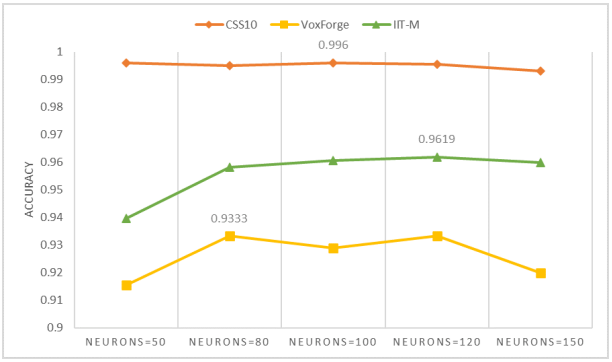


FIGURE 10. Effect of number of hidden neurons of the MLP classifier on the classification accuracy over three S-LID datasets.

In case of SVM classifier, radial basis function (''rbf'') kernel have been used in order to obtain the best classification accuracy.

3) EVALUATION METRICS

In this paper, four evaluation metrics [51] have been used for measuring the performance of the proposed FS algorithm. They are namely Classification accuracy, Precision, Recall and f1-Score. Classification accuracy is the ratio of correctly classified languages to the total number of languages available. Precision for a specific language is the number of true positives (i.e. the number of languages correctly labeled as belonging to the positive class) divided by the total number of languages labeled as belonging to the positive class (i.e. the sum of true positives and false positives, which are languages incorrectly labeled as belonging to the class). Recall is the number of true positives divided by the total number of languages that actually belong to the positive class (i.e. the sum of true positives and false negatives, which are languages which were not labeled as belonging to the positive class but should have been). The f1-Score is the harmonic mean of the precision and recall and can take values between 0 and 1.

TABLE 2. Parameters used for the three classifiers throughout the course of the experiment.

	n-trees(RF)	n-neighbors(k-NN)	neurons(MLP)
CSS10 database	100	10	100
VoxForge database	50	5	80
IIT-M Indic TTS database	100	10	120

TABLE 3. Performance Comparison in terms of classification accuracy using five classifiers on CSS10 database.

Classifier	Accuracy without using FS	Accuracy after using HS-NMR
SVM	98.40%	98.69%
k-NN	96.34%	98.06%
MLP	99.09%	99.60%
NB	98.11%	98.00%
RF	99.54%	**99.89%**

C. RESULTS OBTAINED

In this work, the five classifiers as mentioned earlier have been used for the purpose of S-LID. To validate the performance of the proposed FS algorithm, especially in the field of language identification, experiments were done based on both the original and the optimal feature sets. The results obtained from each of these three datasets using all the classifiers support the claim that FS is an effective step in the learning process of the model. An exhaustive discussion on these results are presented in the subsequent sections.

1) USING CSS10 DATABASE

Primary analysis of the results reveals the fact that higher accuracies are obtained when FS technique has been used compared to its other counterpart for majority of the classifiers. RF classifier has achieved the highest classification accuracy of **99.89%** for the selected feature set, thereby increasing the accuracy by **0.35%** from the raw feature set. The same trend is observed when SVM and MLP classifiers are used. A significant improvement of **1.72%** in accuracy is observed in case of the k-NN classifier. Only in case of NB classifier, there is a very slight decrease in accuracy by **0.11%**. A detailed description of the classification accuracies obtained is presented in Table (3).

The confusion matrix obtained using RF classifier for the raw feature set is presented in Table (4) and that obtained for the selected feature set is presented in Table (5). The

TABLE 4. Confusion matrix of 10 languages (without FS) obtained using RF classifier on CSS10 database.

Language	Chinese	Dutch	Finnish	French	German	Greek	Hungarian	Japanese	Russian	Spanish
Chinese	179	0	0	0	0	0	0	0	0	0
Dutch	0	161	0	0	0	0	0	0	0	0
Finnish	0	0	180	0	0	0	0	0	0	0
French	0	0	2	172	0	0	0	0	0	0
German	0	0	0	0	166	1	0	0	0	0
Greek	0	0	0	0	0	163	2	0	0	0
Hungarian	0	0	0	0	1	0	191	0	0	0
Japanese	0	0	0	0	0	0	0	178	0	0
Russian	0	1	0	0	0	1	0	0	177	0
Spanish	0	0	0	0	0	0	0	0	0	175

TABLE 5. Confusion matrix of 10 languages (with HS-NMR based FS method) obtained using RF classifier on CSS10 database.

Language	Chinese	Dutch	Finnish	French	German	Greek	Hungarian	Japanese	Russian	Spanish
Chinese	179	0	0	0	0	0	0	0	0	0
Dutch	0	161	0	0	0	0	0	0	0	0
Finnish	0	0	180	0	0	0	0	0	0	0
French	0	0	0	173	0	0	0	1	0	0
German	0	0	0	0	166	1	0	0	0	0
Greek	0	0	0	0	0	165	0	0	0	0
Hungarian	0	0	0	0	0	0	192	0	0	0
Japanese	0	0	0	0	0	0	0	178	0	0
Russian	0	0	0	0	0	0	0	0	179	0
Spanish	0	0	0	0	0	0	0	0	0	175

TABLE 6. Accuracy Report of 10 Languages obtained without FS and with HS-NMR based FS method using RF classifier on CSS10 database.

	Without FS			**With HS-NMR**		
Language	**Precision**	**Recall**	**f1-Score**	**Precision**	**Recall**	**f1-Score**
Chinese	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Dutch	0.9938	1.0000	0.9969	1.0000	1.0000	1.0000
Finnish	0.9890	1.0000	0.9945	1.0000	1.0000	1.0000
French	1.0000	0.9885	0.9942	1.0000	0.9943	0.9971
German	0.9940	0.9940	0.9940	1.0000	0.9940	0.9970
Greek	0.9879	0.9879	0.9879	0.9940	1.0000	0.9970
Hungarian	0.9896	0.9948	0.9922	1.0000	1.0000	1.0000
Japansese	1.0000	1.0000	1.0000	0.9944	1.0000	0.9972
Russian	1.0000	0.9888	0.9944	1.0000	1.0000	1.0000
Spanish	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Accuracy			**0.9954**			**0.9989**
Macro Average	**0.9954**	**0.9954**	**0.9954**	**0.9988**	**0.9988**	**0.9988**
Weighted Average	**0.9955**	**0.9954**	**0.9954**	**0.9989**	**0.9989**	**0.9989**

class-wise performance based on different evaluation metrics obtained from the raw feature set is juxtaposed with that from the selected feature set using HS-NMR based FS method in Table (6). Comparing Table (4) and Table (5), it can be concluded that after using FS method, the model is able to identify **8** languages with **100%** compared to **5** languages when FS method is not used. Further, it can also be seen that mis-classification instances of all the languages has also reduced from **8** to **2**.

It is worth noting that the total computation time required to evaluate the proposed HS-NMR based FS algorithm on this database is about 236 minutes. Once the optimal feature set consisting of 612 features (out of 1146 features) is obtained, the training time required is reduced significantly to about a couple of minutes and testing time is reduced to less than 10 seconds using the RF classifier.

2) USING VoxForge DATABASE

After analyzing the results obtained on this database, it is again observed that FS leads to better results compared to original feature set. However, contrary to the results obtained from CSS10 database, here the improvement is much more significant. Here also, RF classifier leads to the best overall classification accuracy of **98.22%** for selected feature set which is a significant increase of **2.66%** from its raw feature counterpart. Additionally, SVM, k-NN and MLP classifiers have performed exceptionally well on the selected feature set. The classification accuracy increased by **3.11%** for SVM classifier, **10.22%** for k-NN classifier and **7.55%** for MLP classifier. However, no improvement is observed when NB classifier is used for both the cases. A detailed description of the classification accuracies obtained is presented in Table (7).

TABLE 7. Performance Comparison of classification accuracies using five classifiers on VoxForge database.

Classifier	Accuracy without using FS	Accuracy after using HS-NMR
SVM	74.67%	77.78%
k-NN	84.00%	94.22%
MLP	85.78%	93.33%
NB	77.78%	77.78%
RF	95.56%	**98.22%**

TABLE 8. Confusion matrix of 6 languages (without FS) obtained using RF classifier on VoxForge database.

Language	English	French	German	Italian	Russian	Spanish
English	31	0	0	0	0	0
French	0	38	0	1	1	0
German	0	0	38	1	0	0
Italian	1	4	0	32	0	2
Russian	0	0	0	0	38	0
Spanish	0	0	0	0	0	38

TABLE 9. Confusion matrix of the six languages (with HS-NMR based FS method) obtained using RF classifier on VoxForge database.

Language	English	French	German	Italian	Russian	Spanish
English	31	0	0	0	0	0
French	0	39	0	0	0	1
German	0	0	39	0	0	0
Italian	1	1	0	36	0	1
Russian	0	0	0	0	38	0
Spanish	0	0	0	0	0	38

The confusion matrix obtained using RF classifier for the raw feature set is presented in Table (8) and that obtained for the selected feature set is presented in Table (9). The class-wise performance based on different evaluation metrics for both the cases using HS-NMR based FS method is presented in Table (10). From Table (8) and Table (9), it can be concluded that total number of mis-classification instances for ''Italian'' language decreased from **7** to **3**. The same trend is seen for both ''French'' and ''German'' languages. It is also observed that three languages namely ''English'', ''Russian'' and ''Spanish'' are identified with **100%** accuracy in both the cases.

Since the total number of utterances in this database is the least out of the three databases used, the computation time required to evaluate HS-NMR on this database is 47 minutes. After the optimal feature set consisting of 413 features (out of 972 features) is obtained, the training time is significantly reduced to below 40 seconds and testing time under 5 seconds using the RF classifier.

3) USING INDIC TTS DATABASE

Contrary to the above mentioned databases which primarily contain audio files of foreign languages collected from different section of the world, this database is a repository of commonly spoken Indian languages. The results obtained on this database complies with the anticipation. Maximum classification accuracy of **99.75%** is obtained when the selected feature set is fed to the RF classifier, thus obtaining an improvement of **6.15%**. SVM, k-NN and MLP classifiers also performed exceptionally well after FS and the corresponding increase in accuracies are **9.22%**, **7.92%** and **6.03%** respectively. Unlike the previous trend of NB classifier, the classification accuracy, for this database, has increased strikingly from **71.59%** to **85.61%**, thus improving the accuracy by more than **14%**. A detailed description of the classification accuracies obtained is provided in Table (11).

The confusion matrix obtained using RF classifier for the raw feature set is presented in Table (12) and that obtained for the selected feature set is presented in Table (13). The class-wise performance based on different evaluation metrics for both the cases using HS-NMR based FS method is presented in Table (14). From Table (12) and Table (13), it is observed that ''Bangla'' is identified with **100%** accuracy when FS is applied. The number of misclassified instances for ''English'' is also reduced. A major improvement is observed in case of ''Hindi'' where the number of misclassified instances is reduced from **29** to **1**. In the case of other 7 languages, it is also observed that the number of misclassified instances reduced upon using FS method. It is interesting to note that even though both ''Bangla'' and ''Assamese'' languages share a similar dialect and there are fair amount of words that are common to both the languages, the model is able to successfully distinguish between ''Bangla'' and ''Assamese'' with no mis-classification.

This database consists for around 3200 utterances. The computation time required to evaluate HS-NMR on this database is 115 minutes. After the optimal feature set consisting of 527 features (out of 1036 features) is obtained, the training time is significantly reduced to around 1 minute and testing time under a 10 seconds using the RF classifier.

D. COMPARATIVE STUDY

To establish the superiority of the proposed FS method, this algorithm is compared with 15 state-of-the-art meta-heuristic FS algorithms, namely:

1) Binary Genetic Algorithm (BGA) [52]
2) Binary Particle Swarm Optimization (BPSO) [53], [28]
3) Binary Gravitational Search Algorithm (BGSA) [54]
4) Binary Cuckoo Search Algorithm (BCS) [55]
5) Binary Dragonfly Algorithm (BDFA) [56]
6) Binary Firefly Algorithm (BFFA) [57]
7) Binary Bat Algorithm (BBA) [58]
8) Binary Grey Wolf Optimizer (BGWO) [59]
9) Binary Whale Optimization (BWO) [60]
10) Binary NMR [45]
11) Binary Moth Flame Optimizer (BMFO) [61]
12) Binary Ant Lion Optimizer (BALO) [62]
13) Binary Krill Heard Algorithm (BKHA) [26], [63], [64]
14) Binary Sailfish Optimizer based on Adaptive β-Hill Climbing (AβBSF) [65]
15) Ring Theory based Harmony Search (RTHS) [66]

This comparison is done for all the databases namely CSS10, VoxForge and Indic TTS. For all of the above mentioned algorithms, the five classifiers have been again used

TABLE 10. Accuracy Report of 6 Languages obtained without FS and with HS-NMR based FS method using RF classifier on Voxforge database.

	Without FS			With HS-NMR		
Language	**Precision**	**Recall**	**f1-Score**	**Precision**	**Recall**	**f1-Score**
English	0.9688	1.0000	0.9841	0.9688	1.0000	0.9841
French	0.9048	0.9500	0.9268	0.9750	0.9750	0.9750
German	1.0000	0.9744	0.9870	1.0000	1.0000	1.0000
Italian	0.9412	0.8205	0.8767	1.0000	0.9231	0.9600
Russian	0.9744	1.0000	0.9870	1.0000	1.0000	1.0000
Spanish	0.9500	1.0000	0.9744	0.9500	1.0000	0.9744
Accuracy			**0.9556**			**0.9822**
Macro Average	**0.9565**	**0.9575**	**0.9560**	**0.9823**	**0.9830**	**0.9822**
Weighted Average	**0.9558**	**0.9556**	**0.9547**	**0.9828**	**0.9822**	**0.9821**

TABLE 11. Performance comparison of classification accuracies using five different classifiers on IIT-M database.

Classifier	Accuracy without using FS	Accuracy after using HS-NMR algorithm
SVM	85.49%	94.71%
k-NN	84.75%	92.67%
MLP	90.16%	96.19%
NB	71.59%	85.61%
RF	93.60%	**99.75%**

TABLE 12. 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	76	2	3	0	0	1	0	0	2	0
English	1	108	4	0	0	0	0	0	0	0
Hindi	3	16	55	0	7	1	0	0	2	0
Marathi	0	0	1	52	0	0	0	0	0	0
Tamil	0	0	2	0	85	0	2	0	0	0
Telugu	0	0	0	0	1	68	1	0	0	0
Assamese	0	0	0	0	1	0	78	0	0	0
Gujarati	0	0	0	0	0	0	0	84	0	0
Kannada	0	0	0	0	0	0	0	0	85	0
Malayalam	0	0	2	0	0	0	0	0	0	70

TABLE 13. Confusion matrix of 10 Indian languages (with HS-NMR based FS method) obtained using RF classifier on IIT-M database.

Language	Bangla	English	Hindi	Marathi	Tamil	Telugu	Assamese	Gujarati	Kannada	Malayalam
Bangla	84	0	0	0	0	0	0	0	0	0
English	1	112	0	0	0	0	0	0	0	0
Hindi	0	1	83	0	0	0	0	0	0	0
Marathi	0	0	0	53	0	0	0	0	0	0
Tamil	0	0	0	0	89	0	0	0	0	0
Telugu	0	0	0	0	0	70	0	0	0	0
Assamese	0	0	0	0	0	0	79	0	0	0
Gujarati	0	0	0	0	0	0	0	84	0	0
Kannada	0	0	0	0	0	0	0	0	85	0
Malayalam	0	0	0	0	0	0	0	0	0	72

TABLE 14. Accuracy report of 10 Indian languages obtained without FS and with HS-NMR based FS method using RF classifier on IIT-M database.

	Without FS			With HS-NMR		
Language	**Precision**	**Recall**	**f1-Score**	**Precision**	**Recall**	**f1-Score**
Bangla	0.9500	0.9048	0.9268	0.9882	1.0000	0.9941
English	0.8571	0.9558	0.9038	0.9912	0.9912	0.9912
Hindi	0.8209	0.6548	0.7285	1.0000	0.9881	0.9940
Marathi	1.0000	0.9811	0.9905	1.0000	1.0000	1.0000
Tamil	0.9043	0.9551	0.9290	1.0000	1.0000	1.0000
Telugu	0.9714	0.9714	0.9714	1.0000	1.0000	1.0000
Assamese	0.9630	0.9873	0.9750	1.0000	1.0000	1.0000
Gujarati	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Kannada	0.9551	1.0000	0.9770	1.0000	1.0000	1.0000
Malayalam	1.0000	0.9722	0.9859	1.0000	1.0000	1.0000
Accuracy			**0.9360**			**0.9975**
Macro Average	**0.9422**	**0.9382**	**0.9388**	**0.9979**	**0.9979**	**0.9979**
Weighted Average	**0.9352**	**0.9360**	**0.9341**	**0.9976**	**0.9975**	**0.9975**

TABLE 15. Parameter settings for 10 state-of-the-art FS methods used for comparison.

FS Algorithm	Parameters
BGA	popSize = 20 MaxIter = 25 Mutation and Crossover Rate = 0.05
BPSO	popSize = 20 MaxIter = 25 Move Rate, w1 = 0.5
BGSA	popSize = 20 MaxIter = 25 Elite Check, EC = 1
BCS	popSize = 20 MaxIter = 25 Arguments in levy fight, $\alpha = 0.1$ and $\beta = 1.5$ Probability of destroy inferior nest, param = 0.25
BDFA	popSize = 20 MaxIter = 25
BFFA	popSize = 20 MaxIter = 25 $\gamma = 1$, $\beta = 0.2$, $\alpha = 0.25$
BBA	popSize = 20 MaxIter = 25 Loudness, A = 0.25 Pulse Rate, r =0.4
BGWO	popSize = 20 MaxIter = 25 $\omega = 0.99$
BWO	popSize = 20 MaxIter = 25 $\omega = 0.99$
BNMR	popSize = 20 MaxIter = 30 Breeding Probability, bp = 0.5
BMFO	popSize = 20 MaxIter = 20 agents = 50 No. of flames = 6
BALO	popSize = 20 MaxIter = 20 agents = 5
BKHA	popSize = 20 MaxIter = 20 NR = 10 NK = 25
AβBSF	popSize = 20 MaxIter = 20 A = 4 $\kappa = 0.001$ omega = 0.9
RTHS	popSize = 5 Pm = 0.2 omega = 0.9

namely SVM, k-NN, MLP, NB and MLP. For all the three databases, RF classifier is found to outperform the remaining four classifiers, therefore, in the rest of the section, the obtained results is presented using RF classifier only. The parameter setting for these 10 FS methods is illustrated in Table (15)

From the results obtained from Table (16), it is observed that for CSS10 database, HS-NMR based FS method achieves the best classification accuracy of **99.89%** while BCS achieves the least accuracy of **98.11%**. For the remaining 2 databases also, the method outperforms the remaining 10 algorithms achieving accuracies of **98.22%** and **99.75%** for VoxForge and Indic TTS databases respectively. Moreover, the evaluation metrics obtained after using HS-NMR is also better than those obtained after using the remaining FS methods. The mis-classification instances are also significantly lowered when FS is done using the proposed method. The comparison (in terms of classification accuracy) between the aforementioned FS algorithms with the proposed method for all the three databases are illustrated in Figure (11).

E. STATISTICAL SIGNIFICANCE TEST

To determine the statistical significance of the proposed method, Friedman test [70-71] has been performed. It is a non-parametric statistical test that allows us to compare more than two population with a blocking variable without having to assume that the observations are normally distributed. Basically, it is the non-parametric version of the Two-Way Block ANOVA, where the rank of the observations or the location of the observations have been used in the dataset and not the actual values themselves. Here, the null hypothesis states that there is no significant difference among the results achieved by 16 FS methods. The test is done across three databases (n) using 16 FS methods (k) as mentioned. The FS methods are ranked row-wise based on their accuracy presented in Table (17). The Friedman statistic (χ_F^2) is calculated with the aid of Table (17) in accordance with Eqn. (15):

$$\chi_F^2 = \frac{12}{nk(k+1)} \sum_{j=1}^{k} R_j^2 - 3n(k+1) \tag{15}$$

where R_j is the sum of ranks obtained by each FS algorithm.

The critical value of (χ_F^2) at a significance level of $\alpha = 0.05$ with 15 degrees of freedom is found to be 24.9958. For this work, a value of (χ_F^2) equal to **29.0249** is obtained which is greater than the critical value. Therefore, the null hypothesis is rejected. The observed p-value is equal to **0.0238** which is less than the significance level. Therefore, from this test, it can be concluded that the results obtained by the proposed HS-NMR based FS method are statistically significant.

F. COMPARISON WITH PAST METHODS

The proposed method is compared to the one proposed by Jog *et al.* [69] and the one proposed by Chowdhury *et al.* [70] which is presented in Table (17). Jog *et al.*, in their work, had used a two-stage approach using Cochleagramn visual representation followed by their feature extraction using different texture descriptors. Finally, they had used an Artificial Neural Network (ANN) for classification of six Indian languages from the Indic TTS Database. Chowdhury *et al.*, in their work, extracted textural descriptors from spectrogram representations of the audio files and used BGWO based FS to get rid of irrelevant features. Finally, they used ANN to for the classification purpose.

The proposed method is compared to the one proposed by Montavon [71] who had used a spectrogram based deep Convolutional Neural Network (CNN) architecture for

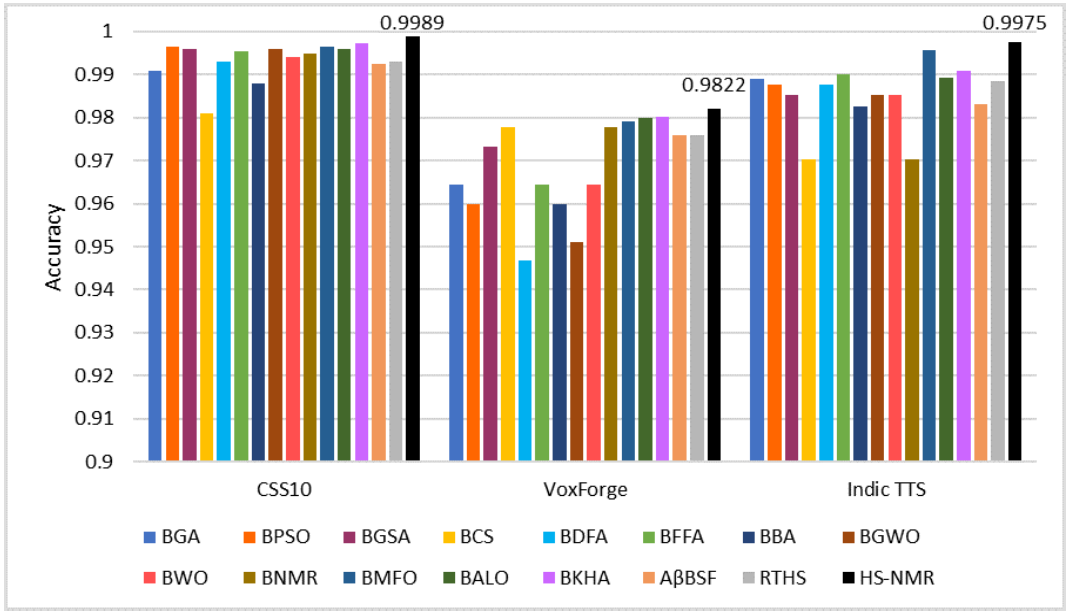


FIGURE 11. Performance comparison of the proposed HS-NMR based FS method with 15 other meta-heuristic FS algorithms on 3 databases.

TABLE 16. Comparison of the overall classification accuracy of the proposed HS-NMR algorithm with 15 other meta-heuristics FS algorithms using RF classifier on the three S-LID databases (The highest accuracy obtained, the lowest number of feature selected and the lowest computation time are highlighted in bold for the three databases).

Database	Parameters	BGA	BPSO	BGSA	BCS	BDFA	BFFA	BBA	BGWO	BWO	BNMR	BMFO	BALO	BKHA	AβBSF	RTHS	HS-NMR
CSS10	Accuracy	99.09%	99.65%	99.60%	98.11%	99.31%	99.54%	98.80%	99.60%	99.42%	99.48%	99.65%	99.6%	99.73%	99.24%	99.31%	**99.89%**
	Number of features selected	732	693	701	682	703	711	687	672	679	674	691	682	677	639	**609**	612
	Computation time(For FS in mins)	219	222	220	222	224	221	**216**	223	226	219	225	219	224	233	239	236
VoxForge	Accuracy	96.44%	96.00%	97.33%	97.77%	94.67%	96.44%	96.00%	95.11%	96.44%	97.77%	97.9%	98.00%	98.02%	97.59%	97.59%	**98.22%**
	Number of features selected	519	512	504	493	514	524	499	466	482	454	488	473	471	442	433	**417**
	Computation time(For FS in mins)	**34**	39	37	40	34	37	36	43	39	36	38	38	39	45	51	47
Indic TTS	Accuracy	98.89%	98.76%	98.52%	97.04%	98.76%	99.01%	98.27%	98.52%	98.52%	97.04%	99.56%	98.93%	99.10%	98.32%	98.84%	**99.75%**
	Number of features selected	663	622	639	613	599	609	588	572	579	558	596	563	571	538	533	**527**
	Computation time(For FS in mins)	**99**	104	102	100	103	103	102	107	104	107	101	105	103	110	119	115

TABLE 17. Rank Comparison of the proposed HS-NMR algorithm with 15 other state-of-the-art meta-heuristics FS algorithms after performing the Friedman Test (the highest rank obtained on the three S-LID databases is highlighted in bold).

Database	BGA	BPSO	BGSA	BCS	BDFA	BFFA	BBA	BGWO	BWO	BNMR	BMFO	BALO	BKHA	AβBSF	RTHS	HS-NMR
CSS10	3	13.5	11	1	5.5	9	2	11	7	8	13.5	11	15	4	5.5	**16**
VoxForge	6	3.5	8	11.5	1	6	3.5	2	6	11.5	13	14	15	9.5	9.5	**16**
Indic TTS	11	8.5	6	1.5	8.5	13	3	6	6	1.5	15	12	14	4	10	**16**
Sum of Ranks	20	25.5	25	14	15	28	8.5	19	19	21	41.5	37	44	17.5	25	**48**
Sum of rank squared	400	650.25	625	196	225	784	72.25	361	361	441	1722.25	1369	1936	306.25	625	**2304**

TABLE 18. Comparison of the method with some past methods on IIT-M database.

Jog et al.		Chowdhury et al.		Our method	
Method	Accuracy	Method	Accuracy	Classifier/Method	Accuracy
Cochleogram based texture descriptors using ANN classifier	95.36%	Spectrogram Image based textural descriptors using GWO based FS method and ANN classifier	96.97%	RF classifier using HS-NMR based FS method	**99.75%**

identifying "English", "French" and "German" using VoxForge database. Finally, it is also compared to the one proposed by Sarthak *et al.* [25]. They had used an attention based deep neural network for classifying six languages namely "English", "French", "German", "Italian", "Russian" and "Spanish" based on their log mel-spectrogram features. The comparison of the results with their obtained ones is presented in Table (18).

VI. CONCLUSION

Although some relevant work have already been done in the field of S-LID, not much concrete researches have been

TABLE 19. Comparison of the method with some past methods on VoxForge database.

Montavon		**Sarthak et al.**		**Our method**	
Method	Accuracy	Method	Accuracy	Classifier/Method	Accuracy
Spectrogram based CNN architecture for identifying 3 languages	80.10%	Deep Neural Network based approach using log Mel-spectrogram features for identifying 6 languages	95.40%	RF classifier using HS-NMR based FS method for identifying 6 languages	**98.22%**

done, to the best of our knowledge, on how to optimize the derived features before feeding the same to the language classification model. In the present work, a new hybrid HS-NMR based FS algorithm have been proposed and its performance is tested on three databases as mentioned above. The algorithm is also compared with 15 state-of-the-art meta-heuristic FS methods on all the three databases and this method has been found to outperform those in every case. In order to check for the statistical significance, Friedman test is performed and the proposed method is found to be statistically significant. Lastly, the proposed method is compared with some past methods and upon evaluating the different performance metrics, it can be concluded that this system produces best results with minimal error.

It is evident from the past research work [72] that in any deep learning architecture, as the number of hidden layers increases, so does the number of hyperparameters, thus making the model very complex and requiring much more computational power and execution time to train the model. Compared to this, the proposed model is much more efficient in terms of both computational power and time. Additionally, it has been able to outperform previous deep learning architectures used for the purpose of S-LID as illustrated in Table (18) and Table (19). Hence, it is logical to consider that HS-NMR based FS method is highly efficient and reliable, especially when it is applied to S-LID. However, since, the proposed method is relatively new, there is always room for improvement and therefore it requires further intensive study. From *No Free Lunch* (NFL) theorem [73], it is known that a particular algorithm cannot be efficient in solving every possible optimization problem because there may or may not be a relationship between how well an algorithm performs and the optimization problem on which it is run. There may be some classes of combinatorial optimization problems that are intrinsically harder or different than others, without much regard to the algorithm one uses. So, it is important to come up with problem specific solutions. After comparing with a multitude of FS algorithms based on their exploration and exploitation capabilities and applying them to the field of S-LID, it can be concluded that the proposed method works best for the optimization problem at hand as evident from Table (16). However, there may be some other optimization problems where the proposed method may fail to reach the global minima. Moreover, whenever this method is applied to other optimization problems, the parameters of the algorithm as well as the transfer function used have to be tuned again which may become tedious. As a future scope of the work, the proposed HS-NMR algorithm can be applied on other popular and interesting research problems, especially in the fields of facial emotion recognition, online/offline musical symbol recognition, handwritten or printed script recognition, etc. Furthermore, this algorithm can be extensively used in the field of speaker recognition using acoustic features like MFCC, LPC, LSF, DWT, i-vector, x-vector, d-vector, etc.

REFERENCES

[1] K. Lee, H. Li, L. Deng, V. Hautamäki, W. Rao, X. Xiao, and A. Larcher, "The 2015 NIST language recognition evaluation: The shared view of I2R, Fantastic4 and SingaMS," 2016.

[2] R. Zazo, A. Lozano-Diez, J. Gonzalez-Dominguez, D. T. Toledano, and J. Gonzalez-Rodriguez, "Language identification in short utterances using long short-term memory (LSTM) recurrent neural networks," *PLoS ONE*, vol. 11, no. 1, Jan. 2016, Art. no. e0146917.

[3] A. Waibel, P. Geutner, L. M. Tomokiyo, T. Schultz, and M. Woszczyna, "Multilinguality in speech and spoken language systems," *Proc. IEEE*, vol. 88, no. 8, pp. 1297–1313, Aug. 2000.

[4] T. Schultz and A. Waibel, "Language-independent and language-adaptive acoustic modeling for speech recognition," *Speech Commun.*, vol. 35, nos. 1–2, pp. 31–51, Aug. 2001.

[5] C. Chelba, T. J. Hazen, and M. Saraclar, "Retrieval and browsing of spoken content," *IEEE Signal Process. Mag.*, vol. 25, no. 3, pp. 39–49, May 2008.

[6] 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.

[7] R. P. Hafen and M. J. Henry, "Speech information retrieval: A review," *Multimedia Syst.*, vol. 18, no. 6, pp. 499–518, Nov. 2012.

[8] E. Alpaydin, *Introduction to Machine Learning*. London, U.K.: MIT Press, 2010, p. 110.

[9] G. Leonard, "Language recognition test and evaluation," Texas Instrum., Central Res. Labs, Dallas, TX, USA, Tech. Rep. TI-08-79-35, 1980.

[10] J. Foil, "Language identification using noisy speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 11, Apr. 1986, pp. 861–864.

[11] F. J. Goodman, A. F. Martin, and R. E. Wohlford, "Improved automatic language identification in noisy speech," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, May 1989, pp. 528–531.

[12] D. Cimarusti and R. Ives, "Development of an automatic identification system of spoken languages: Phase I," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 7, May 1982, pp. 1661–1663.

[13] M. A. Zissman, "Automatic language identification using Gaussian mixture and hidden Markov models," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 7, Apr. 1993, pp. 399–402.

[14] M. Sugiyama, "Automatic language recognition using acoustic features," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1991, pp. 813–816.

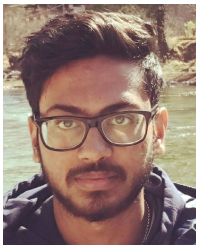
[15] Gazeau, Valentin, and Cihan Varol, "Automatic spoken language recognition with neural networks," *Int. J. Inf. Technol. Comput. Sci. (IJITCS)*, vol. 10, no. 8, pp. 11–17, 2018.

[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] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[18] S. Shukla and G. Mittal, "Spoken language identification using ConvNets," in *Proc. Eur. Conf. Ambient Intell.* Cham, Switzerland: Springer, 2019, pp. 252–265.
[19] 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.
[20] V. Tiwari, "MFCC and its applications in speaker recognition," *Int. J. Emerg. Technol.*, vol. 1, no. 1, pp. 19–22, 2010.
[21] G. S. Kumar, K. A. P. Raju, M. R. Cpvnj, and P. Satheesh, "Speaker recognition using GMM," *Int. J. Eng. Sci. Technol.*, vol. 2, no. 6, pp. 2428–2436, 2010.
[22] L. Abualigah and M. Qasim, *Feature Selection Enhanced Krill Herd Algorithm for Text Document Clustering*. Berlin, Germany: Springer, 2019.
[23] L. Abualigah, "Multi-verse optimizer algorithm: A comprehensive survey of its results, variants, and applications," *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12381–12401, Aug. 2020.
[24] L. M. Q. Abualigah and E. S. Hanandeh, "Applying genetic algorithms to information retrieval using vector space model," *Int. J. Comput. Sci., Eng. Appl.*, vol. 5, no. 1, p. 19, 2015.
[25] L. M. Abualigah and A. T. Khader, "Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering," *J. Supercomput.*, vol. 73, no. 11, pp. 4773–4795, Nov. 2017.
[26] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "Hybrid clustering analysis using improved krill herd algorithm," *Int. J. Speech Technol.*, vol. 48, no. 11, pp. 4047–4071, Nov. 2018.
[27] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A combination of objective functions and hybrid krill herd algorithm for text document clustering analysis," *Eng. Appl. Artif. Intell.*, vol. 73, pp. 111–125, Aug. 2018.
[28] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "A new feature selection method to improve the document clustering using particle swarm optimization algorithm," *J. Comput. Sci.*, vol. 25, pp. 456–466, Mar. 2018.
[29] L. M. Abualigah, A. T. Khader, E. S. Hanandeh, and A. H. Gandomi, "A novel hybridization strategy for krill herd algorithm applied to clustering techniques," *Appl. Soft Comput.*, vol. 60, pp. 423–435, Nov. 2017.
[30] L. Abualigah and A. Diabat, "A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments," *Cluster Comput.*, pp. 1–19, Mar. 2020, doi: 10.1007/s10586-020-03075-5.
[31] L. Abualigah and A. Diabat, "A comprehensive survey of the grasshopper optimization algorithm: Results, variants, and applications," *Neural Comput. Appl.*, to be published.
[32] L. Abualigah, "Group search optimizer: A nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications," *Neural Comput. Appl.*, to be published.
[33] Yang, Xin-She, "Harmony search as a metaheuristic algorithm," in *Music-Inspired Harmony Search Algorithm*. Berlin, Germany: Springer, 2009, pp. 1–14.
[34] R. Vergin and D. O'Shaughnessy, "Pre-emphasis and speech recognition," in *Proc. Can. Conf. Electr. Comput. Eng.*, vol. 2, Sep. 1995, pp. 1062–1065.
[35] 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.
[36] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *J. Acoust. Soc. Amer.*, vol. 87, no. 4, pp. 1738–1752, Apr. 1990.
[37] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 4, pp. 578–589, Oct. 1994.
[38] M. A. Anusuya and S. K. Katti, "Front end analysis of speech recognition: A review," *Int. J. Speech Technol.*, vol. 14, no. 2, pp. 99–145, Jun. 2011.
[39] K. M. Aishwarya, R. Ramesh, P. M. Sobarad, and V. Singh, "Lossy image compression using SVD coding algorithm," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2016, pp. 1384–1389.
[40] 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.
[41] J. W. Cooley, P. A. W. Lewis, and P. D. Welch, "The fast Fourier transform algorithm: Programming considerations in the calculation of sine, cosine and laplace transforms," *J. Sound Vib.*, vol. 12, no. 3, pp. 315–337, Jul. 1970.
[42] Z. Woo Geem, J. Hoon Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *SIMULATION*, vol. 76, no. 2, pp. 60–68, Feb. 2001.
[43] J. H. Kim, "Harmony search algorithm: A unique music-inspired algorithm," *Procedia Eng.*, vol. 154, pp. 1401–1405, 2016.
[44] K. S. Lee and Z. W. Geem, "A new structural optimization method based on the harmony search algorithm," *Comput. Struct.*, vol. 82, nos. 9–10, pp. 781–798, Apr. 2004.
[45] R. Salgotra and U. Singh, "The naked mole-rat algorithm," *Neural Comput. Appl.*, vol. 31, no. 12, pp. 8837–8857, Dec. 2019.
[46] S. Saremi, S. Mirjalili, and A. Lewis, "How important is a transfer function in discrete heuristic algorithms," *Neural Comput. Appl.*, vol. 26, no. 3, pp. 625–640, Apr. 2015.
[47] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Appl. Math. Comput.*, vol. 188, no. 2, pp. 1567–1579, May 2007.
[48] K. Park and T. Mulc, "CSS10: A collection of single speaker speech datasets for 10 languages," 2019, *arXiv:1903.11269*. [Online]. Available: http://arxiv.org/abs/1903.11269
[49] Lounnas, Khaled, Mourad Abbas, Hocine Teffahi, and Mohamed Lichouri, "A language identification system based on voxforge speech corpus," in *Proc. Int. Conf. Adv. Mach. Learn. Technol. Appl.* Cham, Switzerland: Springer, 2019, pp. 529–534.
[50] A. Baby, A. L. Thomas, N. L. Nishanthi, and T. T. S. Consortium, "Resources for Indian languages," in *Proc. Text, Speech Dialogue*, 2016, pp. 1–4.
[51] L. A. Jeni, J. F. Cohn, and F. De La Torre, "Facing imbalanced data-recommendations for the use of performance metrics," in *Proc. Humaine Assoc. Conf. Affect. Comput. Intell. Interact.*, Sep. 2013, pp. 245–251.
[52] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks*. Cham, Switzerland: Springer, 2019, pp. 43–55.
[53] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.
[54] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "BGSA: Binary gravitational search algorithm," *Natural Comput.*, vol. 9, no. 3, pp. 727–745, Sep. 2010.
[55] 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.
[56] 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.
[57] 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.
[58] S. Mirjalili, S. M. Mirjalili, and X.-S. Yang, "Binary bat algorithm," *Neural Comput. Appl.*, vol. 25, nos. 3–4, pp. 663–681, 2014.
[59] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016.
[60] A. G. Hussien, A. E. Hassanien, E. H. Houssein, S. Bhattacharyya, and M. Amin, "S-shaped binary whale optimization algorithm for feature selection," in *Recent Trends in Signal and Image Processing*. Singapore: Springer, 2019, pp. 79–87.
[61] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.
[62] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015.
[63] D. Rodrigues, L. A. M. Pereira, J. P. Papa, and S. A. T. Weber, "A binary krill herd approach for feature selection," in *Proc. 22nd Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 1407–1412.
[64] L. Abualigah and M. Qasim, *Feature Selection Enhanced Krill Herd Algorithm for Text Document Clustering*. Berlin, Germany: Springer, 2019.
[65] 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.
[66] 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.
[67] P. K. Singh, R. Sarkar, and M. Nasipuri, "Significance of non-parametric statistical tests for comparison of classifiers over multiple datasets," *Int. J. Comput. Sci. Math.*, vol. 7, no. 5, pp. 410–442, 2016.
[68] P. K. Singh, R. Sarkar, and M. Nasipuri, "Statistical validation of multiple classifiers over multiple datasets in the field of pattern recognition," *Int. J. Appl. Pattern Recognit.*, vol. 2, no. 1, pp. 1–23, 2015.

[69] A. H. Jog, O. A. Jugade, A. S. Kadegaonkar, and G. K. Birajdar, "Indian language identification using cochleagram based texture descriptors and ANN classifier," in *Proc. 15th IEEE India Council Int. Conf. (INDICON)*, Dec. 2018, pp. 1–6.
[70] A. A. Chowdhury, V. S. Borkar, and G. K. Birajdar, "Indian language identification using time-frequency image textural descriptors and GWO-based feature selection," *J. Exp. Theor. Artif. Intell.*, vol. 32, no. 1, pp. 111–132, Jan. 2020.
[71] G. Montavon, "Deep learning for spoken language identification," in *Proc. NIPS Workshop Deep Learn. Speech Recognit. Rel. Appl.*, 2009, pp. 1–4.
[72] D. Justus, J. Brennan, S. Bonner, and A. S. McGough, "Predicting the computational cost of deep learning models," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 3873–3882.
[73] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.



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.

AANKIT DAS 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. (engineering) degree from Jadavpur University (J.U.), in 2013 and 2018, respectively. He also received the RUSA 2.0 Fellowship for pursuing his Postdoctoral Research in J.U. in 2019. He is currently working as an Assistant Professor with the Department of Information Technology, J.U. He has published more than 50 research articles in peer-reviewed journals and international conferences. His current 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 also a member of The Institution of Engineers, India, and 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).

ALI AHMADIAN (Member, IEEE) received the Ph.D. degree (Hons.) from Universiti Putra Malaysia (UPM), in 2014. He is currently a Fellow Researcher with the Institute of Industry Revolution 4.0, UKM. As a Young Researcher, he is 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); in this context, I have worked on projects related to drug delivery systems, 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 is also a member of editorial board in *Progress in Fractional Differentiation and Applications* (Natural Sciences Publishing) and a Guest Editor in *Advances in Mechanical Engineering* (SAGE), *Symmetry* (MDPI), *Frontier in Physics* (Frontiers), and *International Journal of Hybrid Intelligence* (Inderscience Publishers). He is an author of 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 Simulation*, *Computational Physics*, and so on. He also presented his research works in 38 international conferences held in Canada, Serbia, China, Turkey, Malaysia, and UAE. He was a member of programme committee in 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.

NORAZAK SENU is currently an Associate Professor with the Institute for Mathematical Research, Universiti Putra Malaysia. As his main interests are working on different types of differential equations and modeling real-world systems using such equations, he published more than 100 articles in the peer-reviewed international journals. He received several prizes for his research works from 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. (engineering) degree from Jadavpur University, in 2005 and 2012, respectively. He joined at 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 Fulbright-Nehru Fellowship (USIEF) for Postdoctoral Research in the University of Maryland, College Park, USA, from 2014 to 2015. His current research interests include image processing, pattern recognition, machine learning, and bioinformatics.