
CHAPTER 5

OPTIMIZATION BASED ENSEMBLE FEATURE SELECTION ALGORITHM AND FUZZY CONVOLUTION BIDIRECTIONAL-LSTM CLASSIFIER

5.1 LEVY FLIGHT CUCKOO SEARCH ALGORITHM (LFCSA)

CSA Cuckoo Search Algorithm is advanced metaheuristic algorithms that draws its inspiration from a small number of cuckoo species that intentionally brood parasitize other host birds by depositing their eggs in their nests. Female parasitic cuckoos can mimic the hues and patterns of the eggs produced by certain host species because of the diversification of many cuckoo species. This decreases the likelihood that the eggs will be abandoned, increasing their reproductive potential. It must be noted that possible host birds actively oppose the obtrusive cuckoos. Cuckoo parasites frequently select a nest site where their host bird has already laid eggs. An egg in a nest signifies a result, while an egg from a cuckoo signifies a novel strategy. The goal is to replace the best solutions in the nests with new, maybe superior solutions (cuckoos). Simply put, there is only one egg in every nest. The technique may be applied to exceedingly complicated situations where each nest includes a number of eggs that stand for a number of potential answers (Shehab et al. 2017, Guerrero-Luis et al. 2021). CSA can be summarized as three basic rules:

- Cuckoos lay eggs at specific moments and place them in randomly selected nests.
- Best nests producing highest caliber eggs (solutions) are passed down to next generations.
- The potential host nests counts are predetermined with a probability of $p_a \in [0, 1]$ for hosts to recognize alien eggs. If this happens, the host bird has two options: it can either leave the nest or lay another egg somewhere else, or it can throw the egg away (Shehab et al. 2017).

This kind of breeding nature in Cuckoo Search Algorithm models makes them suitable for a variety of optimization tasks. It is found that the performance achieved with the Cuckoo Search Algorithm can be enhanced applying Lévy flights rather than a plain random walk

(Shehab et al. 2017, Guerrero-Luis et al. 2021). Numerous recent research works published in the literature have successfully applied the cuckoo searches (especially its modified version that utilizes Lévy flights) to difficult optimization issues from a variety of domains. However, to the greatest of our abilities, the method has not yet been applied to mathematical modeling and data fitting. Regarding this, the main superiority of the cuckoo searches using Lévy flights are its simplicity: compared to other metaheuristic strategies, fewer parameters are needed, which means that parameter adjustment is typically a simple operation.

The Levy flight, the type of random walks where the steps have a defined probability distribution, are described in regards to step lengths, and traverse arbitrarily isotropic pathways, generating unique solutions every iteration. A random cuckoo egg is chosen. The true creators of the technique assert that the Lévy flight trick was selected over alternative basic random walks due to its notable enhancement of the CS's overall performance. The following is the process for generating random numbers using Lévy flights: 1. First, a uniform distribution is used to determine a random direction; 2. The Lévy distribution is then used to construct the steps as shown in figure 5.1

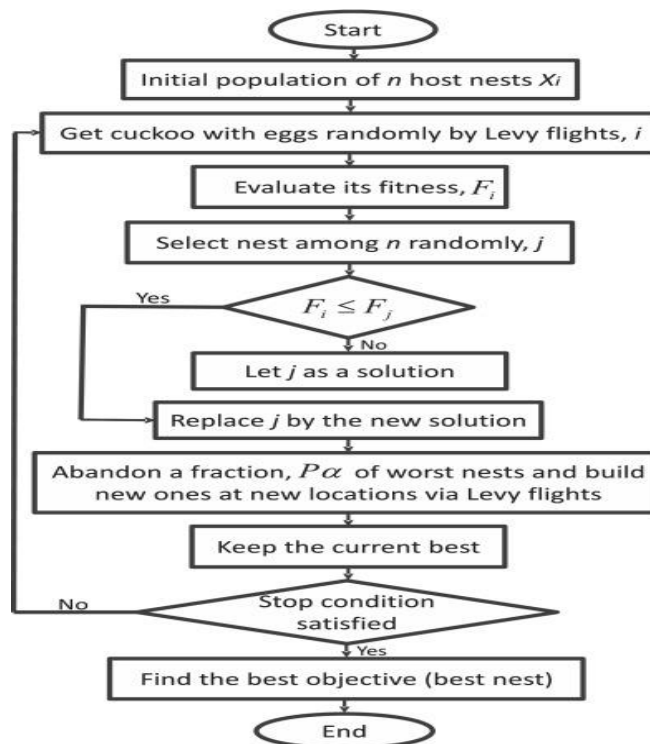


Figure 5.1. Flow diagram of Levy Flight Cuckoo Search Algorithm

5.2 FIREFLY ALGORITHM (FA)

Firefly technique is a metaheuristic method and it is helpful in finding a solution for the optimization problems. The architecture of Firefly Algorithm was designed on the basis of the three following idealized rules: (i) The fireflies are attracted to one another due to their illumination. (ii) The fireflies that are much brighter show better attractiveness level for the rest. (iii) The fireflies having reduced level of brightness move towards others exhibiting greater level of brightness. The three behaviors of fireflies in the nature have been the inspiration behind developing an optimization algorithm known as firefly algorithm. There exists a mutual relationship between the behaviors of fireflies and the development of Firefly Algorithm. Perhaps, every firefly associated with every optimal solution will have its brightness related to the fitness function of the optimal solutions. The behavior that the fireflies having lesser brightness will search for and move towards other fireflies having better brightness levels is identical to the solutions that are newly generated on the basis of old solutions having an increased fitness function as shown in figure 5.2. As a result, in FA, every old solution can be generated new many times based on the comparison between its brightness and those of the others. Due to this, just one new solution of every old solution is retained in accordance with the fitness function comparison.

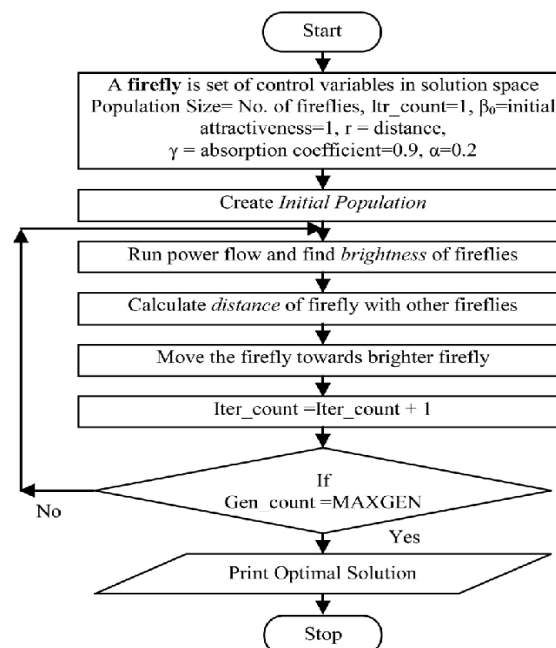


Figure 5.2. Flow diagram of Firefly Algorithm

Based on the hypothesis of intensity light 'I', Firefly Algorithm is presented for which the incidence of light reduces as ' r^2 ' (square of the distance) increases. Due to the fast rise in the origin of the light, the light moves towards the obscure condition on the basis of the light observance (retention) (Wang et al. 2017). For optimization, the current condition is associated with objective functions. In addition to Firefly Algorithms fascination, the difference in light intensity has drawn particular attention from researchers and other experts. However, after Firefly Algorithm is started, a lot of adjustments and improvements are made, and Firefly Algorithm is categorized according to its behavior and performance. It is organized as follows: knowledge of fireflies, fitness function computation, population process, creation of a novel best approach, and movable fireflies.

Firefly Algorithm is one of the efficient and often employed metaheuristic algorithms. Firefly Algorithm has become more common in Feature Selection and categorization. Firefly Algorithm has also been used in biological and healthcare industries in a similar manner. The standard Firefly Algorithm has shown to be effective and successful in a variety of fields over the past few decades. Additionally, the performance and convergence of the Firefly Algorithm have been meticulously improved by optimization community experts. If the rapidly expanding literatures are kept an eye on, several Firefly Algorithm variants are being created daily. The significant techniques have certain shortcomings, such as time & computational complexity, and lack precision. A fresh variant called as adaptable Firefly Algorithm has been created to address all of these drawbacks in the latest work (Wang et al. 2017).

5.3 PROPOSED METHODOLOGY

A new Feature Selection and classification method for the categorization of Parkinson's disease is described in this chapter. Based on the type of speech, features are extracted, dimensionality is decreased utilizing KPCA, OBEFS is carried out using LFCSA, AFA, FMBOA, and LFCSA, FCBi-LSTM is applied to the chosen feature set, and lastly, measures like MCC, f-measures, and accuracy are utilized to evaluate the performance of the classifiers. These are the five key steps of the proposed technique. Figure 5.1 shows the proposed system's overall flowchart.

For the purpose of diagnosing Parkinson's disease, the Parkinson dataset, which contains the characteristics extracted from speech samples, was used. The UCI ML repository provided the data for this research, and its usage was just recently uncovered through analysis (Sakar et al. 2019). The dataset has been used to obtain time-frequency characteristics, baseline characteristics, MFCC, TQWT, and vocal fold features. KPCA is a method for dimensionality reduction that is used quite frequently. In comparison to the entire sound recording field of features, KPCA receives a smaller dimensionality linear subspace, wherein the novel sounds recording attributes of Parkinson's disease receive the biggest variance. Through FMBOA, a dimension-reduced feature vector is used for Feature Selection.

These are the five key steps of the proposed technique. Figure 5.3 shows the proposed system's overall flowchart.

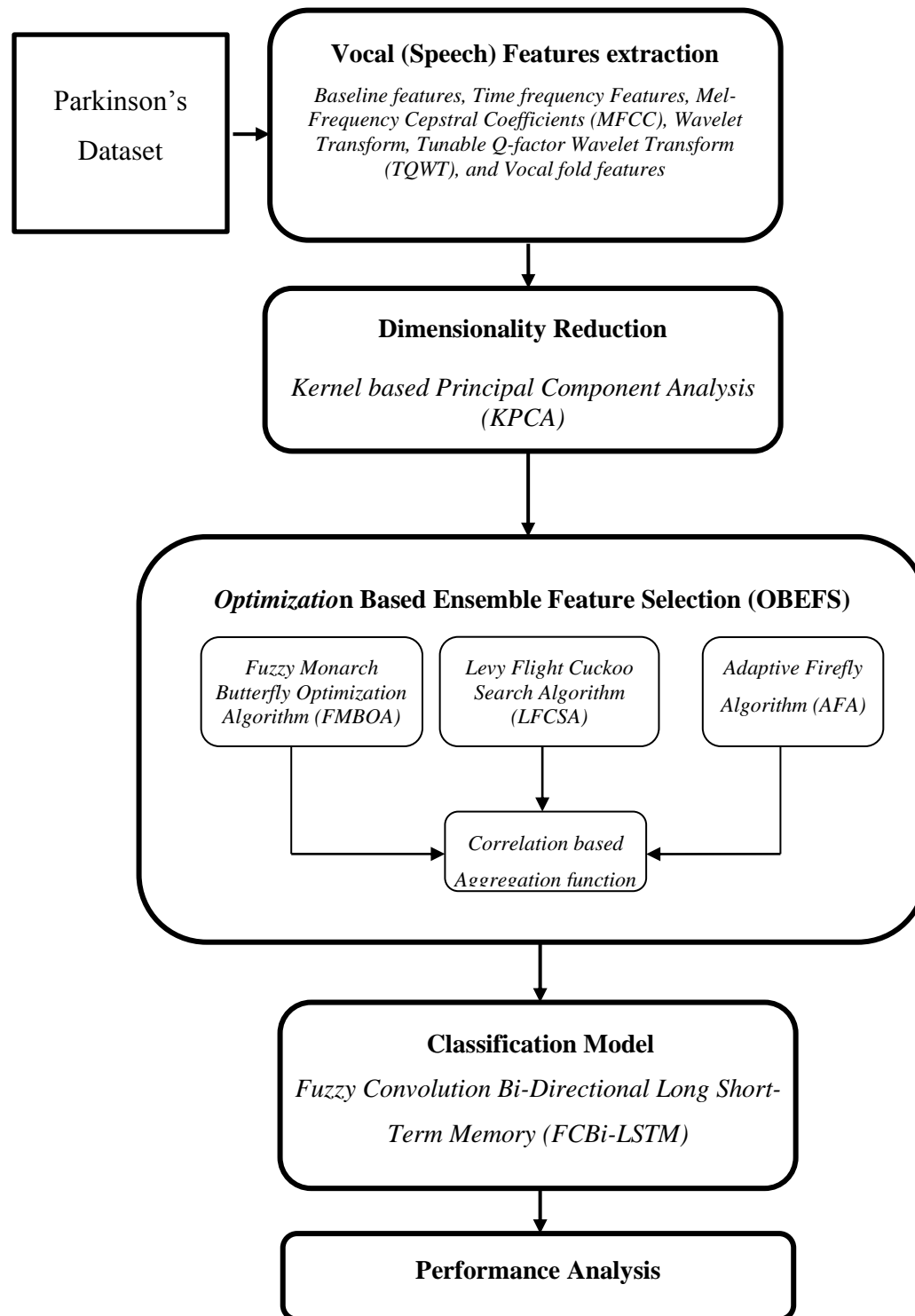


Figure 5.3. Overall Flow of the Proposed System

5.3.1 Optimization Based Ensemble Feature Selection

OBEFS combines the normalized outputs of several Parkinson's disease techniques to experimental ensemble significance. The first stage establishes a collection of several feature selectors, each with a distinct output, and the second stage aggregates the outputs of the various models. Correlation can assist in the aggregate of numerous feature selections, for example, in the case of achieving a balanced feature ranking or by tallying the most often picked features in the case of having to choose a balanced feature subset. In this work, the OBEFS technique unifies the feature rankings that each of the single feature selectors, including FMBOA, LFCSA, and AFA, provides and uses correlation to arrive at a final, balanced rating.

5.3.1.1. Fuzzy Monarch Butterfly Optimization Algorithm

FMBOA is introduced to select the feature sets. Feature Selection of every sample is achieved through FMBOA, when assessing the amount of influence of the features over the existence of Parkinson's disease. The selected attributes of each sample (m represents the number of voice samples) are then given to a classifier. The FMBOA method has been proven to produce improved results in terms of classification accuracy when used consistently, demonstrating that a balance exists between global and local search in this algorithm. The MBOA technique global search update was adjusted in this research so that extremely accurate results may be achieved and its efficiency in acquiring the correct features from the start and prior to switching to local search is increased. Every single butterfly comprehends the features and has a local interaction with one another, sharing information within the swarm, leading to the developing potential of the system (Alweshah2020, Wang et al. 2015, Feng et al. 2018). It is achieved by using two operations, namely the Migration operator and Butterfly operator adjustment.

5.3.1.2. Levy Flight Cuckoo Search Algorithm

Cuckoo Search Algorithm takes its inspiration from the intriguing behavior of few cuckoo species known as obligate inter-specific brood-parasitism (Mesa et al. 2018). This pattern of behavior is dependent on the fact that only a small number of species choose the best host based on the attributes in the dataset to raise their offspring. The Cuckoo Search Algorithm mechanism is used to avoid parental responsibility for raising their children and to

decrease the possibility of egg loss (unnecessary traits) to other species. Placing the eggs (features) in different nests reveals which ones have the finest qualities. The strategy's goal is to use these new cuckoo egg-related solutions (features) to replace the present approach with eggs (unnecessary characteristics) that were previously placed in the nest. This substitution, performed in an iteration, might possibly increase the solution quality as the iterations increase, slowly resulting in an excellent solution of the Feature Selection problem. Especially, the CSA depends on three well-defined rules (Guerrero et al. 2015, Mareli et al.2018),

1. One egg (characteristics) at a time, each cuckoo places its eggs into a randomly selected nest (accuracy).

2. The following generation will inherit the nests with the finest eggs, or characteristics of better quality, guaranteeing that good answers, or attributes, are maintained over time.

3. As things stand, there is a limited quantity of host nests available. The likelihood that a host will recognize an alien egg (feature) is $prb_a \in [0, 1]$. The notion that new nests will replace a ratio prb_a of the n persisting host nests allows for the estimation of this rule. N host nests comprise the initial population in which the LFCSA algorithm starts. To represent the initial numbers of the k^{th} component in the j^{th} nest, use Equation (5.1).

$$f_j^k(0) = \mu \cdot (up_j^k - low_j^k) + low_j^k \quad (5.1)$$

where up_j^k and low_j^k stands for the lower and upper values of that k^{th} elements (feature), correspondingly, and μ indicates the consistent random variables on the open period (0, 1). These boundary criteria require controlling at every step of iteration to make sure that these feature values are inside the FS search spaces. For every iterations' t , the cuckoo eggs (feature), say i , is chosen in random and new feature outcome f_i^{t+1} are produced. Utilizing Lévy flights instead of random walks can increase the efficiency of this random search. A Lévy flight is a kind of random walk where the steps are characterized by step lengths that obey a given probability distribution, and where the orientations of the steps must be randomized and isotropic. Here, the common expression for the Lévy flight is the Equation (5.2),

$$f_i^{t+1} = f_i^t + \alpha \oplus \text{levy}(\lambda) \quad (5.2)$$

where $\alpha > 0$ denotes the step size, the symbol \oplus indicates the entry-wise multiplication and the superscript t denotes the current generation. Random walks can be less efficient than Lévy flights in this random search. The Lévy flight is a type of random walk in which the positions of every step need to be arbitrary and isotropic, and the steps themselves have step lengths that follow a specified probability distribution.

$$\text{levy}(\lambda) \sim g^{-\lambda}, (1 < \lambda \leq 3) \quad (5.3)$$

Two crucial processes are involved in creating the random numbers using Lévy flights from a computational position:

First, choosing a random direction from a uniform distribution comes first.

The next step is to generate the set of steps that correspond to the chosen Lévy distribution.

Mantegna algorithm is utilized for symmetric distributions (Feng et al. 2018). This technique helps computing the factor by Equation (5.4),

$$\hat{\phi} = \left(\frac{\Gamma(1 + \hat{\beta}) \cdot \text{Sin}\left(\frac{\pi \cdot \hat{\beta}}{2}\right)}{\Gamma\left(\frac{1 + \hat{\beta}}{2}\right) \cdot \hat{\beta} \cdot 2^{\frac{\hat{\beta}-1}{2}}}\right)^{\frac{1}{\hat{\beta}}} \quad (5.4)$$

Where, Γ represents the Gamma functions. $\hat{\beta} = 3/2$ is utilized in the current work (Feng et al. 2018), therefore, the same range is used. The Mantegna algorithm takes this factor into account while calculating the step length ζ using Equation (5.5).

$$\zeta = \frac{u}{|v|^{\frac{1}{\hat{\beta}}}} \quad (5.5)$$

Where, u and v adopt the normal distributions of 0 mean and deviations σ_u^2 and σ_v^2 , correspondingly. Here, σ_u conforms to the Lévy distributions expressed by Equation (5.4) and $\sigma_v = 1$. So, the steps size ζ is computed by Equation (5.6),

$$\zeta = 0.01\zeta(f - f_{best}) \quad (5.6)$$

where ζ is computed as per Equation (5.6). At last, x is changed as: $f \leftarrow f + \zeta \cdot \Psi$, wherein Ψ refers to the arbitrary vectors having the dimensions of the result x and adopts the normal distributions $N(0, 1)$. After evaluating the fresh solution's (FS) fitness (accuracy), the LFCSA compares it to the existing solution. If an improvement is seen, the existing solution is replaced with this new one (FS). Subsequently, a portion of the worst nests are replaced and disregarded with fresh arbitrary (FS) results in order to expand the FS search spaces. The replacements rate is decided in random using a parameter known as the probability value prb_a , whose tuning has to be done right for improve the performance. So, each one of the current solutions (FS solutions) are graded during every iteration step as per their fitness (accuracy) and the best solution (feature) attained till now is stored to be the feature vector f_{best} in Equation (5.6). The algorithm is used in an iterative manner till a predefined termination condition is fulfilled. A Pseudo code for the LFCSA is outlined in algorithm 5.1.

ALGORITHM 5.1. LEVY FLIGHTS CUCKOO SEARCH ALGORITHM (LFCSA)

Begin

Objective functions $f(f), f = (f_1, \dots, f_d)^T$ with $d = \dim(\Omega)$

Generate Initial Populations of N host nests $f_i (i = 1, \dots, N)$

While ($t < \text{MaxGeneration}$)

Set initial values for cuckoos (Say i) randomly by Levy flights

Judge Fitness F_i using accuracy of the classifiers

Select a nest Among N (say j) randomly

If ($F_i > F_j$)

 Replace j by the new solution

End

A fraction (prb_a) of worse nest (features in the dataset) are ignored and new features are constructed using Levy Flights ; Retain the best solutions in terms of accuracy of the feature

End while

 Post Process findings and Visualization ; End

5.3.1.3. Adaptive Firefly Algorithm

Firefly algorithm depends on the well-defined behavior of the flashing characteristic of fireflies (Wang et al. 2017). The three rules below are defined for clearly specifying FA,

- All fireflies are unisex and the objective of fireflies will be attracting one another irrespective of their sex so that the features are optimally selected from the dataset,
- Attractiveness is in proportion to their brightness (accuracy), and it reduces with the increase in distance between two fireflies,
- The firefly brightness is decided in regard to the objective function (accuracy).

The brightness I_n changes according to the distance in exponential and monotonical way. It is defined using Equation (5.7),

$$I_n = I_{n_0} e^{-\gamma r} \quad (5.7)$$

where I_{n_0} refers the initial brightness and γ stands for the light absorbance coefficients. Equation (5.8) is used to describe a firefly's attraction β , which is proportional to its brightness as viewed by nearby fireflies (features).

$$\beta = \beta_0 e^{-\gamma r^2} \quad (5.8)$$

Where the attraction at $r=0$ is indicated by $\beta_0 = 1$. When one firefly (feature) (i) becomes attracted to another (feature) (j), which is determined by Equation (5.9), the firefly (feature) "i" moves.

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \varepsilon \quad (5.9)$$

The third term, which comes from a Gaussian distribution, describes the randomization with step α . In the conventional FA, the third term of Equation (5.9) refers to the randomization with steps α . It is generated with the similar distributions ranging between [0,1]. The steps are defined using a static or linear technique most of the time; it only uses the highest generation, which has been shown to be addicting to a variety of fireflies. At first, every firefly exhibits the same step configurations, and its value reduces as the iterations progress. As a result, it may slip into the local optimum, leading to early convergence.

Second, at the first stage of the search, a large step can cause the firefly to overlook the better possible solution if it is nearby. Hence, the search performance may decrease.

At last, Equation (5.9) is useful for firefly in exploring the new FS space using a bigger step, however it does not come to use during the convergence to global optimum. When the step size is less, the result is vice versa. Due to this, the phase α has a significant impact on the algorithm explorations and convergences. The value progressively falls with more iterations used, but drops quickly with fewer iterations used. An innovative method using self-adaptive steps is presented that takes into account the aforementioned difficulties and uses each firefly unique experience to select the dataset features most effectively.

To find a solution to the above stated issues, the step of firefly is at a distance from the optimal solution has to be as big as possible. Fireflies between the above two are utilized for having a tradeoff between the local search and global search to optimally select the features from the dataset. Hence, the step of firefly should also be considered with its past data and present condition. The past data of the Firefly is used in this research and it comprises the optimum value of its earlier two iteration. Equations 5.10–5.11, which are based on multiple experiments and the remarks made above, are used to calculate each firefly's step α . It is given as below,

$$h_i(t) = \frac{1}{\sqrt{(f_{pi}(t-1) - f_{pi}(t-2))^2 + 1}} \quad (5.10)$$

$$\alpha_i(t+1) = 1 - \frac{1}{\sqrt{(f_{best}(t) - f_i(t))^2 + h_i(t)^2 + 1}} \quad (5.11)$$

Here, $h_i(t)$ refers to the last 2-iterations past data of i^{th} fireflies. f_{pi} stands for the fitness values of the better result of i^{th} fireflies. f_{best} indicates the fitness values of the better result of populations got till now, and f_i stands for the fitness values of i^{th} fireflies, which depict the present data. Every firefly has the ability to self-adapt in the following iteration step depending on how different their present fitness values are from the maximum fitness score of the population. Because of this, every firefly step may vary with every iteration, and every repetition also alters every firefly step.

ALGORITHM 5.2. ADAPTIVE FIREFLY ALGORITHM (AFA)

Begin

Objective function $f(x), x = (x_1, \dots, x_d)^T$

Create Initial Population of n fireflies $x_i (i = 1, \dots, n)$

Derive light intensity I_n by objective functions $f(x)$

While ($t < \text{MaxGeneration}$)

Specify absorption coefficient γ

Assess Fitness F_i in terms of accuracy of the classifier

For $i=1$ to n (n fireflies)

For $j=1$ to n (n fireflies)

If ($I_j > I_i$)

Move firefly i towards j

End if ; Change attractiveness with distance r through $\exp(-\gamma r^2)$

Evaluate the recently selected feature solutions, adjust the light intensity, and modify each firefly's stride. Equations (5.10) and (5.11), when used to calculate the step.

End for j

End for i

Rank the best features and get the current best features

End *while*

Post-Process the results and visualization

End

5.3.1.4. Correlation function

Correlation-based ensemble features selector helps in calculating the correlations values of every feature chosen from these 3-techniques, and the features with high similarity are eliminated from every feature set; later, the features chosen from all the three techniques are provided to an ensemble feature selector. The ultimate optimal feature set of the ensemble feature selector is attained using majority voting on the output of their feature sets individually. Using the following Equation (5.12), the correlation coefficient matrix is computed for the features chosen in the Out-ensemble FS outcome.

$$\text{CorrelationCoefficient} = \frac{N \sum xy - (\sum x)(\sum y)}{\sqrt{[N \sum x^2 - (\sum x)^2][N \sum y^2 - (\sum y)^2]}} \quad (5.12)$$

Where, x and y refer to the features scores that are considered and N indicates the overall instance numbers. The input for the classification is the collection of features selected by the correlation-based ensemble features selector.

5.3.2 Parkinson's Disease Classification using FCBi-LSTM Classifier

With the FCBi-LSTM classifier, PD categorization is accomplished in this work. The FCBi-LSTM classifier adjusts the response of the Parkinson features for the extraction of the highly relevant features from the predicted Parkinson features and computes the fuzzy weight using the membership function. CNN and FCBi-LSTM classifiers are used to process the characteristics selected from the PD dataset (Chen et al. 2017). Layers for convolution and pooling make up the network. After performing a convolution, it pools the data, and the following convolution layer uses the pooled data as input. CNN has a major edge in picture feature extraction due to its properties. CNN employs convolution with partial filters, depending on how biological visual cells interpret their local vision. In particular, the interaction between the local filter and the inputs term's local sub-matrix forms the inner product. Applying convolution layers with filters to several output matrices helps improve the representations of the attributes chosen from the PD datasets. The CNN pooling layer helps reduce additional matrix dimensions while maintaining the internal connections of the chosen attributes. The input of the pooling layer, which is the mean pooling layers, comes from the convolution layer that came before it, and its output becomes the input for the layer that comes after. When evaluating Bi-LSTM data, the outcome of the last convolution layers serves as an intermediary variable (Xingjian et al. 2015). Hence, LSTM doesn't just use a non-linear component for transforming the inputs and loop cells. Fuzzy weights computation is achieved with the Gaussian Membership function. Bi-LSTM can acquire more structural information, and therefore its performance is much superior compared to individual directional LSTM. The two Convolution layers perform the processing of the ultimate output

of Bi-LSTM so that PD diagnostics becomes possible. Multi-modal Factorized Bilinear Pooling (MFB) is helpful in combining the features that the CNN processes and then Bi-LSTM performs the processing of the features.

5.4 EXPERIMENTAL RESULTS

The experimental findings obtained with the comparison between the suggested FCBi-LSTM classifiers and the methods like FCLSTM-CNN, CNN, and SVM are explained in this section. The UCI Machine Learning repository provided the data that was used in this work, and it is still being used in research today [29]. Through the use of Matrix Laboratory R2016, the PD identification and classification system tests were carried out (MATLAB R2016a). The system specs for the device on which it is installed are an Intel(R) CoreTMi3-4160T CPU@3.10 GHz 3.09 GHz processor, 4.00 GB of RAM, Windows 8.1 pro, 64-bit OS, and a 1 TB hard drive.

For the cuckoo search algorithm, the upper limit and lower limit is set as 5 and the step size is set 0.01. The fitness value is calculated using a total number of iterations 719514 and the best fitness value obtained is 8.2941e-06 For the firefly algorithm, the randomness value is set to be high with a variable $\alpha = 0.2$, the absorption coefficient i.e. $\gamma = 0.1$, randomness reduction is 0.97.

Cuckoo Search:

Probability range: [0 1]

fitness value:

Total number of iterations=719514

bestnest4 =

Columns 1 through 9

0.99975	0.99888	1.0012	0.99919	1.0002	0.99937	1.0003	1.0001
1.0013							

Columns 10 through 15

0.99994	0.99993	1.0008	0.99959	0.99906	0.99905
---------	---------	--------	---------	---------	---------

fmin4 = 8.2941e-06

Fire fly algorithm:

% Grid values are used for display only

Ngrid=100;

Best score value:

best4 =

4.8833	-1.9337	3.3224e-12
-4.8711	-2.081	3.7992e-12
-4.8514	-2.0276	4.4237e-12
-4.8473	-2.012	4.5833e-12
-4.9797	-2.686	6.0724e-12
-4.838	-2.2314	6.9283e-12
-4.9007	-1.4198	1.0059e-11

5.4.1. Performance Metrics

Analysis of the classifiers' predictive performance is done in terms of evaluation measures. The accuracy, error, MCC, and F-Measure are among the evaluation metrics used to gauge how successfully the classifiers predict results.

5.4.2. Results Comparison

In the experiment, classifiers are employed in conjunction with three distinct feature categories. Metrics like MCC, F-Measure, accuracy, and error are among those used to assess how well the classifiers perform. Combining the properties of MFCC, Wavelet, and Concat with an SVM classifiers yield the accuracy rate of 88.1294%. Even though the CNN classifier's accuracy rate using the MFCC+Wavelet+Concat ensemble is 94.1752%. TQWT + Wavelet + Concat, TQWT+MFCC+Wavelet, MFCC+Wavelet+Concat, and TQWT + MFCC + Concat yield accuracy values of 93.0470%, 93.0854%, 93.1261%, and 95.1557%, respectively, according to the FCLSTM-CNN classifier. The suggested FCBi-WLSTM classifier performs best when used with the MFCC+Wavelet+Concat ensemble (See Table 5.1), with accuracy, F-measure, MCC, and error rate while utilizing the ensemble feature sets.

With an f-measure rates of 98.5010% and 71.400% for MCC, its accuracy rate 98.7720%. The TQWT+MFCC+Wavelet ensembles of the set of features yield notable results of 98.3100%, 96.6381%, 74.300%, and 3.3619% for f-measure, accuracy, MCC when contrasted with other characteristic combinations sets. The results are displayed in figures 5.1–5.3.

Table 5.1 Accuracy Comparison Results of Classifiers with Triple Feature with (KPCA+ OBEFS)

Feature Extraction Combination	FCBi-LSTM	FCLSTM	CNN
TQWT+MFCC+Wavelet	96.6381	93.0470	87.5696
TQWT+MFCC+Concat	98.0244	93.0854	91.9315
TQWT+ Wavelet + Concat	97.3457	93.1261	88.0694
MFCC + Wavelet + Concat	98.7720	95.1557	94.1752

Table 5.1 describes the Accuracy Comparison Results of Classifiers with Triple Feature with KPCA+ OBEFS. In this result MFCC+Wavelet+Concat shows the higher accuracy compared to the other feature combinations. It gives 98.7720% of accuracy for FCBi-LSTM classifier, 95.1557% of accuracy for FCLSTM Classifier and 94.1752% of accuracy for CNN classifier

Table 5.2 F-measure Comparison Results of Classifiers with Triple Feature with KPCA+ OBEFS

Feature Extraction Combination	FCBi-LSTM	FCLSTM	CNN
TQWT+MFCC+Wavelet	98.3100	94.2258	85.8697
TQWT+MFCC+Concat	96.5900	91.5250	90.2315
TQWT+ Wavelet + Concat	97.5200	93.4200	86.3695
MFCC + Wavelet + Concat	98.5010	91.6921	92.4752

Table 5.2 describes the F-measure Comparison Results of Classifiers with Triple Feature with KPCA+ OBEFS. In this result MFCC+Wavelet+Concat shows the higher accuracy compared to the other feature combinations. It gives 98.5010% of F-measure for FCBi-LSTM classifier, 91.6921% of F-measure for FCLSTM Classifier and 92.4752% of F-measure for CNN classifier.

Table 5.3 Mathews Correlation Coefficient Comparison Results of Classifiers with Triple Feature with KPCA+ OBEFS

Feature Extraction Combination	FCBi-LSTM	FCLSTM	CNN
TQWT+MFCC+Wavelet	74.300	67.6669	57.2007
TQWT+MFCC+Concat	72.300	67.7060	61.4600
TQWT+ Wavelet + Concat	70.300	65.4457	63.3994
MFCC + Wavelet + Concat	71.400	67.2960	64.5384

Table 5.3 describes the MCC Comparison Results of Classifiers with Triple Feature with KPCA+ OBEFS. In this result TQWT+ Wavelet + Concat shows the higher accuracy compared to the other feature combinations. It gives 70.300% of MCC for FCBi-LSTM classifier, 65.4457% of MCC for FCLSTM Classifier and 63.3994% of MCC for CNN classifier.

5.5 SUMMARY

This chapter provides a multi-class classification problem for the aim of applying FS to analyze PD. It is suggested that OBEFS and FCBi-LSTM be used in the PD analysis. Several algorithms are used to implement the suggested OBEFS algorithm, such as FMBOA, LFCSA, and AFA. To select the better features from each of the 3-feature subsets, OBEFS uses the correlation function. Then, the FCBi-LSTM classifier is presented as a PD diagnostic instrument. A classification approach's efficacy is assessed using f-measure, accuracy, error, and MCC. The outcomes show that contrasted to the competing methods, the accuracy of the suggested model is much higher.