



International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Parkinson 'S Disease Diagnosis Using An Ensemble Deep Learning Model By Utilizing Voice Features

Ramya N¹, Dr. S.Devi Suganya²

¹Ph. D (Research Scholar), Department of Computer Science, Vellalar College for Women (Autonomous), Erode Tamil Nadu, India.
E-mail: nramyainfotech@gmail.com, ORCID: 0009-0001-0831-3453

²Assistant Professor Department of Computer Science, Vellalar College for Women (Autonomous), Erode, Tamil Nadu, India.
E-mail: sdevisuganya@gmail.com

Abstract

A chronic and progressive neurodegenerative disease, Parkinson's disease (PD) affects speech patterns, motor control, and cognitive abilities, making early and precise diagnosis vital for effective intervention. However, many existing diagnostic models face significant limitations, such as sensitivity to outliers in normalization processes and the inability to effectively capture temporal patterns in voice data critical for early-stage PD detection. Additionally, conventional models often suffer from poor pre-process, inadequate feature selection, which can lead to overfitting and decreased performance in categorization. To move through these difficulties, this paper proposes a novel ensemble-based diagnostic model that integrates advanced pre-processing, optimization, and classification techniques. The approach begins by applying the Synthetic Minority Oversampling Technique (SMOTE) to address class imbalance. A hybrid normalization strategy combining Z-Score and Min-Max normalization is employed to improve feature scaling while reducing the influence of outliers. For feature selection, the Improved Whale Optimization Algorithm (IWOA) is used to identify the most discriminative and relevant features. The classification process leverages a powerful ensemble of models: An Enhanced Adaptive Neuro-Fuzzy Inference System (EANFIS), a Deep Belief Network (DBN) with hyperparameters optimized using the Pelican Optimization Algorithm (POA), and a Modified Long Short-Term Memory (M-LSTM) network. This ensemble exploits the combined strengths of fuzzy logic, deep feature learning, and temporal sequence modelling. The suggested model performs noticeably better than existing techniques on a number of criteria, including accuracy, precision, recall, and F1-score, according to experimental data. Combining efficient optimization with deep hybrid learning, and temporal modelling makes this framework a promising and effective tool for the early and reliable diagnosis of Parkinson's Disease.

Keywords: Parkinson's Disease (PD), Deep Learning, Ensemble Classification, Feature Selection; SMOTE, Z-Score Normalization, Improved Whale Optimization Algorithm (IWOA), Adaptive Neuro-Fuzzy Inference System (EANFIS), Deep Belief Network (DBN), Modified Long Short-Term Memory (M-LSTM), Pelican Optimization Algorithm (POA).

1. Introduction

Parkinson's disease significantly impacts individuals and society at large. Dopaminergic neurons in the brain are disrupted in PD, the neurodegenerative condition that is the most common. This disorder causes speech impairment, cognitive decline, bradykinesia, weakness, and tremors. In countries like India, awareness and early diagnosis of PD remain limited, contributing to delayed interventions and poorer outcomes. The Parkinson's Foundation predicts that over 10 million individuals worldwide have Parkinson's disease, and the global burden is expected to increase significantly due to aging populations [1].

Parkinson's disease may not be a single condition, according to a number of findings. The term "parkinsonism" refers to a clinical illness that might seem to have several distinct causes. The accepted reasons include the less than 10 documented genes that, when mutated, may undoubtedly cause parkinsonism. Secondly, the disease often presents with extremely varying symptoms and progression patterns, even when a definite cause is identified [2,3]. For instance, even when parkinsonian symptoms have the same toxic cause, to being exposed to the heroin analog neurotoxin 1-methyl-4-phenyl-1,2,3,6-tetrahydropyridine (MPTP), the presentation might differ significantly across individuals. Third, there are significant differences in the priorities, requirements, and desires of every individual with Parkinson's disease.

However, traditional diagnosis, based largely on clinical observation and neurological evaluation, often fails to detect early-stage PD or distinguish it from other movement disorders [4]. This underlines the need for efficient computer-aided diagnostic (CAD) tools that can assist in the early and accurate identification of

PD symptoms, especially those derived from subtle voice and speech changes that manifest early in the disease progression.

In the last decade, data mining techniques and machine learning models have gained traction for PD diagnosis because they are able to managing high-dimensional, non-linear biomedical data. These models use voice datasets (such as those from the UCI Parkinson's dataset) to extract and analyze vocal features like jitter, shimmer, and harmonic-to-noise ratios, which are impacted by vocal cord stiffness in PD patients. Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Decision Trees are examples of traditional machine learning techniques have demonstrated acceptable classification performance [3]. These models' ability to capture temporal relationships and sequential speech data is, however, limited by their frequent reliance on manually constructed features and static pattern recognition. Furthermore, they typically lack mechanisms for robust feature selection and are sensitive to issues like class imbalance and noisy inputs. As a result, deep learning models and hybrid data mining frameworks have emerged to resolve these issues and improve the precision of diagnosis [5].

To enhance the predictive capability and clinical relevance of Parkinson's Disease (PD) detection systems, this study proposes a robust computer-aided diagnosis (CAD) framework that integrates advanced pre-processing, optimization, and ensemble classification. The model begins with the SMOTE to address class imbalance, followed by a hybrid normalization strategy combining Z-Score and Min-Max normalization to improve feature scaling while mitigating the influence of outliers. Feature selection is performed using the Improved Whale Optimization Algorithm (IWOA), making certain that just the most discriminative and relevant characteristics are included. The classification phase leverages an ensemble comprising an EANFIS, a DBN with hyperparameters fine-tuned via the POA, and a M-LSTM network. This architecture enables effective extraction of temporal patterns, robust decision-making, and strong generalization across diverse voice datasets. Furthermore, the proposed framework introduces a temporal slicing mechanism and ensemble consensus strategy, improving performance on small and imbalanced datasets, and enhancing suitability for real-time clinical deployment and telehealth integration.

The article's remaining content is arranged as follows: A overview of pertinent studies on data mining and deep learning techniques for PD diagnosis is provided in Section II. Section III describes the proposed diagnostic framework in detail, including the pre-processing steps, feature selection mechanism, and the ensemble-based classification strategy. The experimental findings are examined in Section IV, which also compares the suggested model's performance to that of other PD detection methods. The paper's main conclusions are presented in Section V, which also suggests future research options for improving early and precise Parkinson's disease diagnosis.

2. Literature Review

Almasoud et al [2022] [6] adaptive moment estimation (ADAM) optimization on the network hidden layer and recurrent neural networks (RNNs) in the batch normalization layer of GLSTM have significantly enhanced the suggested classification model. This proposed system reduces dimensionality to understand the importance of feature engineering by using Linear Discriminant Analysis (LDA) to decrease dimensionality and Sparse Auto-Encoder (SAE) to extract dynamic speech features. Dynamic elements are examined by energy content from voice to voiceless (offset) and unvoiced to voice. PD datasets without sample overlap are assessed using 10-fold cross-validation. The accuracy of persistent phonations is evaluated quantitatively using the suggested smart PD detection technique, RNN-GLSTM-ADAM.

Reshma et al [2024] [7] introduced a new method for assessing Parkinson's disease by recognizing local and global gait features. An improved convolutional neural network (ICNN) is used to retrieve global features, while a stacked autoencoder is used to extract local features. To increase the CNN model's efficacy, hyperparameters such as batch size, learning rate, and number of convolutional layers may be changed using the Enhanced Sunflower Optimisation (ESO) approach. The modified bidirectional LSTM (MBi-LSTM) classifier uses a combination of local and global data as input to identify the severity of Parkinson's disease. The performance of the proposed model is extensively assessed using the GAIT-IT and GAIT-IST datasets, which include important parameters including accuracy, precision, recall, and the F-measure.

El-Hasnony et al [2020] [8] the proposed model uses a chaotic tent map for initialization and adapts the adaptive neuro-fuzzy inference system (ANFIS) constraints by using the advantages of particle swarm optimization (PSO) and grey wolf optimization (GWO). At the edge of the gateways, fog processing is used to gather and evaluate data and provide immediate notifications to the local population. The evaluation metrics for the GWO, PSO, differential evolution (DE), genetic algorithm (GA), ant colony optimization (ACO), and grey wolf optimization (GWO) demonstrated the model's advantages in the standard ANFIS model, standard deviation (SD), accuracy, and five standard datasets from the UCI machine learning repository.

Pasha et al [2024] [9] presented a unique hybrid strategy that uses an ANFIS with two optimizers, Adam and PSO, to accurately forecast Parkinson's disease. Each training step, an Adam optimizer can change each ANFIS parameter's rate of learning to identify a better solution faster. To approximate and classify

nonlinear functions, ANFIS uses fuzzy logic. The behavior of individuals like birds served as the inspiration for the metaheuristic PSO method. When compared to current techniques, combining these two approaches may enhance PD diagnostic accuracy and resilience. To improve the built-in ANFIS model's prediction accuracy, the suggested approach incorporated the benefits of both optimization strategies.

Anisha and Arulanand [2020] [10] using ensemble classifiers on the speech characteristics dataset obtained from the UCI repository, the suggested approach makes an early prediction of PD. With three speech feature occurrences per participant, the speech features dataset has several feature subsets of 252 individuals. The highly related traits are integrated without surrendering any information by using Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). Ensemble classifiers are used to predict PD because integrating many models, either the same or different, improves prediction results. The research of PD prediction uses a variety of ensemble classifiers, such as Gradient Boosting Machine (GBM) and Extreme Gradient Boosting (XgBoost), Adaptive Boosting (AdaBoost), and Bagging. Grid search and random search are examples of hyperparameter tuning approaches that are to determine the optimal conditions for inclusion into the ensemble classifiers. To evaluate ensemble classifier models, many metrics are utilized, like support, F1 score, recall, accuracy, and precision.

Berdjough et al [2023] [11] suggested a new deep feed-forward neural network for detecting abnormalities in health indicators that is assisted by pelican gorilla troop optimization. The bootstrapping technique is used to enhance anomaly detection during the data augmentation phase because deep learning needs a substantial amount of data to provide accurate results. To finish data preparation and improve detection results, Z-score normalization is also used. According to the findings of the experimental assessment, the suggested method achieves detection performance with regard achieving 0.879, 0.902, and 0.929 in terms of accuracy, precision, and recall, respectively.

Vats and Mehta [2024] [12] deployed a deep belief network (DBN) method, considered a highly advanced machine learning algorithm, which is more of a memory structure capable of learning deeply and hierarchically. Their study implied the use of a DBM model for a diverse data set of 500 PD subjects suspected to have disease in its early stage. The dataset contains medical records, speaking analysis, audio recordings of subjects, and biometric monitoring. The model was trained using a two-phase training approach. The first phase is an unsupervised pre-training process to learn general characteristics. The results were very encouraging: 93% accuracy, 90% sensitivity, 93% specificity, and an AUC of 0.797 were all obtained using the DBN model.

Singh et al [2023] [13] to address clustering issues, the expanded whale optimization algorithm (EWOA) is presented. Two further operational processes are added to and modified from the whale optimization algorithm (WOA). The technique increases the rate of convergence and expands the search space by using the water wave optimization algorithm's position modifying equations. To address the problem of local optima, the tabu and neighbourhood search approaches were introduced. The suggested EWOA is tested using eight benchmark datasets in a simulation. The findings are compared to seven clustering techniques and methods. Each algorithm's performance is assessed using average intra-cluster distance and f-measure metrics.

3. Proposed Methodology

Feature selection, data pre-processing, and multi-model classification are some of the sequential steps which make up the suggested ensemble-based diagnostic model. Initially, the Data Pre-processing phase addresses class imbalance using the SMOTE. For the minority class, SMOTE creates new data manually, enhancing dataset balance and improving classifier generalization. Subsequently, a hybrid normalization strategy is applied, combining Z-Score Normalization and Min-Max Normalization. This hybrid approach ensures that features are not only standardized (by scaling to unit variance after removing the mean) but also scaled within a defined range, reducing the influence of outliers and enhancing model convergence. In the Feature Selection phase, the IWOA is utilized to identify the most significant and discriminative features. IWOA efficiently navigates the search space to select optimal features that reduce dimensionality, improve learning speed, and enhance classification accuracy. Finally, the Classification stage leverages a robust ensemble framework that integrates three powerful models:

- Enhanced Adaptive Neuro-Fuzzy Inference System (EANFIS): Incorporates adaptive learning capabilities with fuzzy logic for dealing with uncertainty and imprecision in data.
- Deep Belief Network (DBN): Employs deep hierarchical feature extraction. Its hyperparameters are optimized using the POA, ensuring optimal learning performance.
- Modified Long Short-Term Memory (M-LSTM): Long-term dependencies and temporal patterns in sequential data may be captured using this improved LSTM model.

This ensemble strategy capitalizes on the complementary strengths of fuzzy systems, deep learning, and recurrent neural networks, ultimately leading to improved diagnostic accuracy. Figure 1 shows the general process of the suggested paradigm.

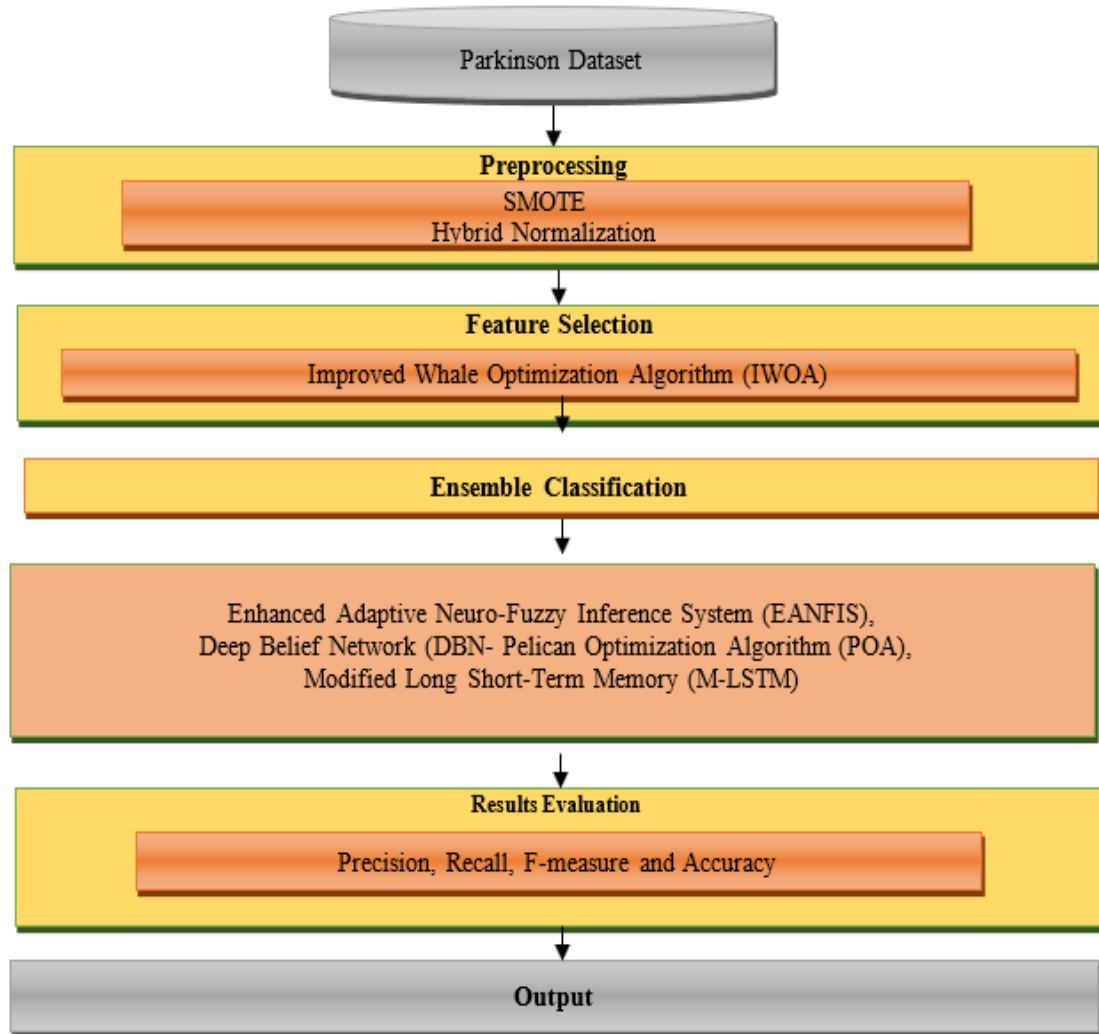


FIGURE 1. PROPOSED ARCHITECTURE DIAGRAM

3.1. Dataset Description

The public may access the voice data used in this investigation at archive.ics.uci.edu/ml. Provided by Oxford University, the dataset contains a variety of acoustic speech features gathered from 195 individuals, including 147 patients diagnosed with Parkinson's disease. It includes multiple voice recordings per subject, with each feature representing a specific vocal measurement. The dataset's "status" column indicates the health condition of the individual; negative values correspond to healthy subjects, while positive values indicate Parkinson's disease cases, enabling clear distinction between the two groups. The dataset is extensively examined in Table 1.

TABLE 1. DETAIL OF PD DATASET

Dataset Characteristic	Multivariate
Counts of Instances	197
Attribute Characteristics	Real
Counts of Attributes	23
Missing Values	N/A
Made by	The University of Oxford's Max Little
Associated Tasks	Classification
Types of Classification	Binary {1 for PD patients and 0 for healthy}

3.2. Data Pre-processing

SMOTE, that generates synthetic minority class samples to improve data distribution, is an essential part of the data pre-processing step that addresses class imbalance and improves model performance. Additionally, a hybrid normalization strategy combining Z-Score and Min-Max normalization is employed to standardize feature scales and reduce outlier impact, ensuring robust and efficient model training.

3.2.1. Data Balancing by Synthetic Minority Over-Sampling Technique (SMOTE)

Oversampling was applied during the pre-processing stage due to the imbalanced nature of the PD dataset analysed in this study. This data analysis technique modifies the class distribution to increase the minority class's sample size in comparison to the majority class. To increase the minority group's presence in the dataset, this is usually accomplished by randomly replicating instances from the minority group. One widely used method for handling class imbalance is SMOTE, the minority class's manufactured samples are created by linear interpolation. The SMOTE algorithm functions in two main steps [14,15].

The initial step consists of finding the k nearest neighbours by computing the Euclidean distances between instances of the minority class and arranging them in increasing order. Based on these calculated distances, the k -nearest neighbours (KNN) are selected. This process uses Equation (1) to measure the Euclidean distance between two minority class samples, x and y , across n features.

$$d(x,y)=\sqrt{\sum_{a=1}^n(x_a - y_a)^2} \quad (1)$$

In the second step, artificial data is generated using interpolating between the two minority samples. A candidate from the k NN is randomly selected, and new data points are generated by combining the selected neighbour (y) with the original sample (x). The interpolation process is defined by Equation (2) for the a -th attribute between x and y :

$$\text{SyntheticData}_a(x,y) = x_a + r \cdot (y_a - x_a) \text{ for } 0 \leq r \leq 1 \quad (2)$$

Where,

r - indicates a random number in the range of 0 to 1.

To produce the required number of synthetic samples using the provided formula, this process is repeated for each attribute across all n dimensions.

3.3. Hybrid Normalization Strategy

To enhance the effectiveness of data scaling, a hybrid normalization strategy is implemented, integrating both Z-Score Normalization and Min-Max Normalization. This combined approach leverages the strengths of each technique standardizing data distribution while also constraining feature values within a fixed range resulting in improved model stability and faster convergence. A hybrid normalization strategy is employed by first applying Z-Score normalization to standardize feature distributions, the standardized data are then scaled within a predetermined range using Min-Max normalization [16]. This dual-step process enhances consistency across features and supports faster, more stable model training.

Z-Score normalization, also known as standardization, is a widely used data pre-processing technique for scaling feature values in a dataset. Each characteristic is altered using this procedure to possess a standard deviation of one and a mean of zero [17]. It is particularly beneficial when features have varied scales or units, as it ensures that all attributes contribute equally during model training.

The following is the formula for Z-Score normalization:

$$z = \frac{x - \mu}{\sigma} \quad (3)$$

Where:

- X is the data's initial element,
- μ represents the dataset mean,
- σ is the dataset's standard deviation.

This transformation standardizes each data point by taking the standard deviation and dividing it by the mean. With this approach, the data becomes suitable for machine learning algorithms that rely on normalized input for better training speed and performance [18]. Once normalization is complete, the data proceeds to the feature selection stage, where the most informative features related to speech signals are extracted from the pre-processed dataset.

To further improve feature scaling and restrict values within a specific bound, Normalized Z-Score data is subjected to Min-Max normalization. Through the use of a scaling method termed min-max normalization, input values are converted to either $[0, 1]$ or $[-1, 1]$. A scaling method termed min-max normalization converts input numbers to a predetermined range, usually $[0, 1]$ or $[-1, 1]$. Using the formula in equation 4, in the range $[low, high]$, this method transforms the input value x of the attribute X to x_{norm} [19].

$$x_{norm} = \frac{(high-low) \cdot (x - minX)}{maxX - minX} \quad (4)$$

The input data set's attribute X has a minimum value of $minX$ and a maximum value of $maxX$. The data scaled to the $[0, 1]$ range using the aforementioned method. This combined normalization approach Z-Score followed by Min-Max ensures that the data is both standardized and scaled within a bounded range, thereby enhancing consistency across features. It is particularly effective for algorithms sensitive to input scale and improves model convergence and learning efficiency [20]. Once this hybrid normalization is completed, the data is passed to the feature selection stage, where the most significant features related to speech signals are extracted from the processed dataset

3.4. Feature selection Using Improved Whale Optimization Algorithm (IWOA)

Once normalization is done, it is necessary to choose the optimal features since, for this study, a limited portion of the input data shows a variety of profiles for various groups or samples. Hence, IWOA is used in this research to help in feature selection. The WOA is usually an innovative approach for stochastic optimization, which was formed some time before, in accordance with nature-influenced population dynamics. The solution for an optimization problem can be found with the help of the search agents in the WOA [21]. The World Ocean to identify its prey, association use a technique termed "bubble-net hunting" that imitates the actions of humpback whales. The three basic phases of the WOA include surrounding the prey, searching for the ideal prey and attacking with a bubble net.

Equations (5) and (6) provide the WOA's fundamental mathematical component.

$$X(t+1) = X^*(t) - A \cdot |C \cdot X^*(t) - X(t)| \text{ if } p < 0.5 \quad (5)$$

$$X(t+1) = |C \cdot X^*(t) - X(t)| \cdot e^{bl} \cos(2\pi t) + X^*(t) \text{ if } p \geq 0.5 \quad (6)$$

X^* , where the most effective solution to date is $A=2a \cdot (r-a)$; $C=2 \cdot r$ for all iterations from 2 to 0; to have a linear reduction in a coefficient vector a , b is set to 1; t is time; and X is a vector that represents the positions of all the whales. The logarithmic spiral's form is determined by the constant variable t . Between 0 and 1, the values supplied to the random vector, r , occur inside this range. When the positions of the whales are updated, equations (5) and (6) are updated using any random value between -1 and 1. The probabilities in equations (5) and (6) are 50% and 50%, respectively, suggesting that whales select one of the paths at random and with the same probability during the optimization step. Vector A has a random value in the bubble-net stage, that ranges from [-1, 1], but during the search phase, it may fluctuate between 1 and -1. The search technique utilized is given by Eq (7).

$$X(t+1) = X_{rand} - A \cdot |C \cdot X_{rand} - X(t)| \quad (7)$$

The WOA algorithm uses this stochastic search strategy to do a global search. When the WOA search initially starts, results are generated at random. Through a series of phases, the approach is later used to update these results. The analysis will proceed after the maximum number of phases has been reached.

Improved whale optimization algorithm

An optimization algorithm must be strongly balanced between exploration and exploitation to achieve better outcomes. Thus, as the number of repetitions in WOA increases, the step size decreases. It is determined by parameter A . It has been shown, nonetheless, that modest divergence during the WOA iteration restricts the slips to the local best [22].

This study uses an enhanced whale optimization technique to address these issues. The value of A may be altered using the levy fly function. This enhances the exploitation and exploration strengths of WOA concurrently.

The Levy distribution may be expressed mathematically as follows.

$$L(s, \gamma, \mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} \exp\left[-\frac{\gamma}{2(s-\mu)}\right] \frac{1}{(s-\mu)^{3/2}} & \text{if } 0 < \mu < \infty \\ 0 & \text{if } s \leq 0 \end{cases} \quad (8)$$

In this case, μ , γ , and s stand for the scale, which is controlled by the set of samples and the position and scale variables of distribution, respectively (equation 8).

An enhanced algorithm for optimizing whales

1. START
2. Import data
3. Determine where the whale population X is located
4. Compute each whale's fitness
5. Compute A and C after initializing a and r
6. Update the current best whale's position to X^*
7. Set iteration counter $t=1$
8. while $t \leq \text{max iterations}$ do
9. for each whale do
10. Generate random number $p \in [0,1]$
11. if $p < 0.5$ then
12. if $|A| < 1$ then
13. Update whale position using Eq. (5):
14. Else
15. Randomly select a whale X_{rand}
16. Update position using Eq. (8) with Levy flight:
(Levy function introduces stochastic step to avoid local optima)
17. end if
18. else ($p \geq 0.5$)
19. Update whale position using Eq. (6) with Levy perturbation:

20. end if
 21. end for
 22. Update X^* if a better solution is found
 23. Increment $t=t+1$
 24. end while
 25. Output X^* as the best solution
 26. END

3.5. Classification using Ensemble Model

Ensemble methods leverage the principle of combining multiple models to develop a categorization system that is more reliable and accurate. Among these, the average-based ensemble approach is widely recognized for its simplicity and effectiveness. This method aggregates the predictions of several classifiers by averaging their outputs to produce a final decision. By doing so, it balances out the individual weaknesses of each model while capitalizing on their collective strengths, leading to improved overall performance.

In this study, an average-based ensemble classification approach is proposed, which integrates three distinct and complementary classifiers. The first is an Enhanced Adaptive Neuro-Fuzzy Inference System (EANFIS), capable of handling uncertainty and imprecision in data through fuzzy logic and learning capabilities. The second is an Enhanced Deep Belief Network (EDBN), a deep generative model whose hyperparameters are fine-tuned using the Pelican Optimization Algorithm (POA), thereby enhancing its learning efficiency and accuracy. In sequential data, the third is a Modified Long Short-Term Memory (M-LSTM) network, it efficiently records long-term dependencies. Together, these classifiers contribute to a comprehensive ensemble framework that enhances classification accuracy and robustness.

3.5.1. Enhanced Adaptive Neuro-Fuzzy Inference System (ANFIS)

Fuzzy logic-based neural networks (NNs) analyze a given dataset in neuro-fuzzy systems to determine properties (sets and rules). ANFIS offers experimental examples for attaining notable results while simulating non-linear processes [23]. It evaluates the dataset's features and modifies the system's settings in accordance with the given error criteria. To anticipate the parameters of FIS, NN algorithms are used for learning.

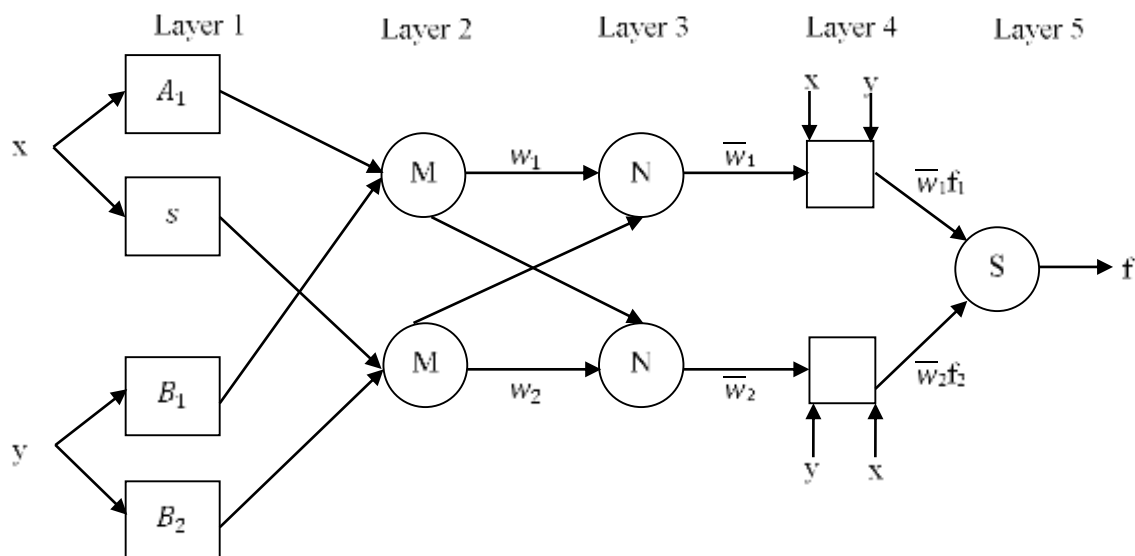


FIGURE 2. EANFIS ARCHITECTURE

The EANFIS architecture is seen in Figure 2. The adaptive node is shown as a square, while the fixed node is shown as a circle. The output is represented by z . Let's assume that the inputs are "S" and "MS," which stand for missing and non-missing samples, respectively. A typical ruleset with two fuzzy if-then rules is shown below. The proposed ANFIS in this work aims at developing a first-order Sugeno fuzzy model.

Rule 1: if s is A_1 and ms is B_1 then $z_1 = p_1s + q_1ms + r_1$

Rule 2: if s is A_2 and ms is B_2 then $z_2 = p_2s + q_2ms + r_2$

here, p_i, q_i, r_i ($i = 1, 2$) define the parameters used in the learning process, while A_i, B_i ($i = 1, 2$) indicate fuzzy sets of predecessors. Below is a full explanation of the five EANFIS layers seen in Figure 2,

Layer 1: Each node i has a defined purpose and is adaptable.

$$O_i^1 = \mu_{A_i}(s), i = 1, 2 \quad (9)$$

$$O_i^1 = \mu_{B_i}(ms), i = 3, 4 \quad (10)$$

In which s and ms denotes input of node i , and $\mu_{A_i}(s)$, and $\mu_{B_i}(ms)$ adopts any Membership Function (MF). Triangular membership function is defined as:

$$\mu(x; a, b, c) = \begin{cases} 0 & x \leq a \text{ or } x \geq c \\ \frac{(x-a)}{(b-a)} & a < x \leq b \\ \frac{(c-x)}{(c-b)} & b < x < c \end{cases} \quad (11)$$

Such that c describes the centre of triangular membership function (equation 11), and a denotes lower bound (left foot of the triangle), b denotes center (peak where membership is 1), c denotes upper bound (right foot of the triangle).

Layer 2: Each node, which is provided as follows, multiplies the incoming samples to show the stability of the rule's ring,

$$O_i^2 = \omega_i = \mu_{A_i}(s) \cdot \mu_{B_i}(ms), i = 1, 2, 3, 4 \quad (12)$$

Layer 3: The node determines the proportion between the general stability of rules and the stability of the rules' ring [24].

$$O_i^3 = \bar{\omega}_i = \frac{\omega_i}{\omega_1 + \omega_2}, i = 1, 2, 3, 4 \quad (13)$$

In which $\bar{\omega}_i$ Shows normalized ring stability in equation 13.

Layer 4: The node function is described by

$$O_i^4 = \bar{\omega}_i z_i = \bar{\omega}_i(p_i s + q_i ms + r_i), i = 1, 2, 3, 4 \quad (14)$$

Here equation 14 $\bar{\omega}_i$ denotes parameter set, and $\{p_i, q_i, r_i\}$ is the layer 3 output. The subsequent parameters are the names given to the parameters in this layer.

Layer 5: This layer's node adds together all of the incoming sample datasets to determine the total result.

$$O_i^5 = \bar{\omega}_i z_i = \frac{\omega_1 z_1 + \omega_2 z_2}{\omega_1 + \omega_2}. \quad (15)$$

It is clear from EANFIS that if there are fixed values of premise parameters, the total output is represented as integrated consequent parameters

$$z = (\bar{\omega}_1 s)p_1 + (\bar{\omega}_1 ms)q_1 + (\bar{\omega}_1)r_1 + (\bar{\omega}_2 s)p_2 + (\bar{\omega}_2 ms)q_2 + (\bar{\omega}_2)r_2. \quad (16)$$

3.5.2. Deep Belief Network (DBN) with Pelican Optimization Algorithm

There are many levels of hidden units in a DBN, with each layer made up of binary latent variables. The design of these layers is similar to that of a Restricted Boltzmann Machine (RBM) in that each hidden layer functions as the visible layer for the one above it. After training on a set of data samples, a DBN can probabilistically reconstruct its inputs. It can then be further fine-tuned under supervised learning for classification tasks.

Working Procedure of DBN:

The first hidden layer is trained by the DBN to directly extract features, such pixel values, from the input data. The subsequent hidden layer uses the preceding hidden layer's output as its input, which is designed to capture more abstract representations. The lower limit on the training data's log-likelihood becomes better with each more layer, signifying better model performance [25].

The training process of a DBN follows the Greedy Learning Algorithm (GLA), a top-down approach that trains each layer individually in a generative manner. The generative weights define the connections between layers. An RBM is generated by the top two hidden layers, where multiple iterations of Gibbs sampling are carried out. Once the RBM is trained, a single pass of ancestral sampling is conducted through the remaining layers. The final hidden states are estimated using a bottom-up pass that computes the latent variable values layer by layer. Greedy pretraining begins from the bottom layer and successively fine-tunes the generative weights in the reverse direction using the observed input data. Figure 3. Shows DBN with POA

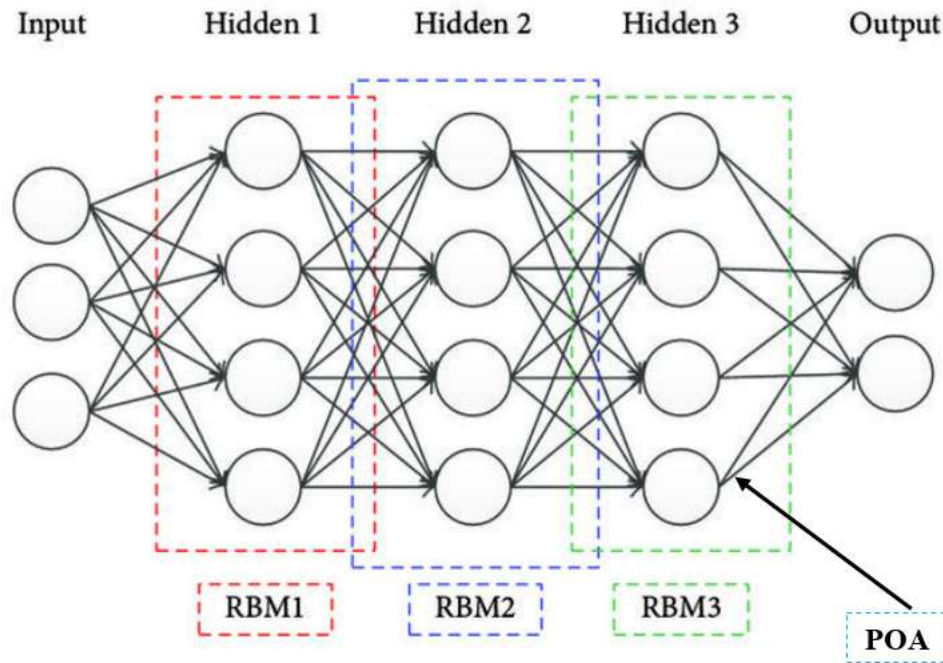


FIGURE 3. DEEP BELIEF NETWORK WITH POA

Each RBM is trained individually through the Contrastive Divergence algorithm, it has the positive phase and the negative phase as its two primary phases. The hidden binary states in the positive phase units are determined based on the visible units and weights, increasing the model's probability of reproducing the training data [26]. The negative phase aims to reduce the model's probability of generating samples not present in the dataset. Each RBM is trained iteratively until all layers of the DBN are optimized.

Creating the DBN Model:

The DBN's final top layer is trained to generate class labels for input vectors, enabling it to classify unseen data. The energy-based function used in an RBM to compute the joint probability distribution is defined as in equation 17:

$$En(v, h, \theta) = v^T W h - a^T v - b^T h = \sum_{i=1}^{D_v} \sum_{j=1}^{D_k} w_{ij} v_i h_j - \sum_{i=1}^{D_v} a_i v_i - \sum_{j=1}^{D_k} b_j h_j \quad (17)$$

where $\theta = \{b_i, a_j, w_{ij}\}$ has units i and j biases, and a_i and b_j are the biases of visible cell i to hidden cell j . The joint probability distribution for the RBM is provided by (equation 18):

$$P(v, h; \theta) = \frac{1}{Z(\theta)} \exp(-E(v, h; \theta)) \quad (18)$$

where $Z(\theta)$ is a normalizing constant's value by i & j . It is computed by equation (19),

$$Z(\theta) = \sum_v \sum_h \exp(-E(v, h; \theta)) \quad (19)$$

RBM uses the energy equation to estimate input dataset probability. Cells i and j have conditional probability functions given by the joint probability distribution function. It is computed by equations (20-22),

$$P(h_j = 1|v) = \delta(b_j + \sum_i v_i w_{ij}) \quad (20)$$

$$P(v_i = 1|h) = \delta(a_i + \sum_j h_j w_{ij}) \quad (21)$$

$$\delta(x) = \frac{1}{1 + \exp(-x)} \quad (22)$$

Using Eq. (21), each v_i is set to 1 to reconstruct the input. The hidden units are updated accordingly to represent features of the input. Training's objective is to increase considering the characteristics in the model, the probability of the training data that represents in equation 23:

$$\text{maximize} \{b_j, a_i, w_{ij}\} \frac{1}{m} \sum_{l=1}^m \log(P(v^l)) \quad (23)$$

The quantity of training samples is denoted by m . Since the computation of $Z(\theta)$ is complex, gradient-based optimization is performed using sampling techniques like Contrastive Divergence or Persistent Contrastive Divergence. Once the DBN has been fully trained using this greedy layer-wise method, the topmost layer serves as the classifier. This layer uses the high-level features extracted by the RBMs to perform

classification. A softmax activation function is applied for multi-class classification, and a sigmoid function is used for binary classification tasks.

Hyperparameter Optimization using Pelican Optimization Algorithm (POA):

Hyperparameters like learning rate, number of hidden units, and batch size are tuned using the POA to improve the DBN's classification performance. The cooperative hunting techniques of pelicans served as the inspiration for POA, a metaheuristic that successfully finds exploration and exploitation in harmony. It works especially well in high-dimensional search regions like neural network optimization due to its adaptive behavior and global convergence properties.

By integrating POA, the DBN model avoids suboptimal configurations that could hinder learning efficiency. POA systematically searches the hyperparameter space and fine-tunes the settings to achieve minimal classification error, resulting in faster convergence, improved generalization, and enhanced robustness in predictive tasks.

Pelican Optimization Algorithm (POA)

In the pursuit of the global optimum, POA, a novel optimization approach inspired by stochastic nature, is thought to provide superior exploration and exploitation [27]. The most popular algorithms lately are those that are based on swarms. POA is influenced by pelican behavior and hunting tactics. Pelican hunting groups are common. The pelicans determine their food source and then descend, which is prayer, and then extend their wings to let the prayers reach the shallow water and the water's surface, allowing them to attack them more readily. The following sets the POA's primary steps:

Initialization: The population-based algorithm POA considers each pelican a solution. At the start of optimization, Equation 24 randomly initializes each population member.

$$x_i = LB + rand * (UB - LB) \quad i = 1, 2, \dots, N \quad (24)$$

N is the number of individuals in the population, x_i is the value of the potential solution, and $rand$ is a random vector in the interval $[0, 1]$. The responses of those initial candidates are then used to assess the objective function for the given problem. The objective function's vector is then determined. By simulating the behavior of pelicans as they attack their food source, the suggested solution is updated. There are two stages to this process simulation:

Phase 1: Moving towards food source (Exploration phase):

This stage replicates the pelican's approach to searching the region for a food source. The Pelicans begin their approach to the prey position after recognizing their prey. The ability to generate the prey location at random, which increases exploring power, is a key component of POA. The following is a mathematical simulation of the present phase of The following is the i th pelican candidate solution:

$$x_i^{new-1} = \begin{cases} x_i(t) + rand \cdot (x_p - Ix_i), & \text{if } F(x_p) \leq F(x_i) \\ x_i(t) - rand \cdot (x_p - Ix_i), & \text{else} \end{cases} \quad (25)$$

Equation 25 states that the prey's randomly produced position is denoted by x_p , its objective function is represented by $F(x_p)$, and I is a vector with a value of 1 or 2 that is generated at random. Equation 26 then performs an update to the answer depending on the changed position:

$$x_i(t+1) = \begin{cases} x_i^{new-1}, & \text{if } F(x_i^{new-1}) \leq F(x_i) \\ x_i(t), & \text{else} \end{cases} \quad (26)$$

Phase 2: Winging on the water surface (exploitation phase):

Pelicans ascend at this phase by extending their wings on the water. The following is a mathematical description of the new state of the i th pelican candidate solution on phase 2:

$$x_i^{new-2} = x_i(t) + R \cdot \left(1 - \frac{t}{T}\right) \cdot (2rand - 1)x_i(t) \quad (27)$$

In equation 27, t is the maximum number of iterations, R is a constant equal to 0.2, and t is the current iteration. The updated position is then reflected in the response as

$$x_i(t+1) = \begin{cases} x_i^{new-2}, & \text{if } F(x_i^{new-2}) \leq F(x_i) \\ x_i(t), & \text{else} \end{cases} \quad (28)$$

Where $F(x_i^{new-2})$ is the candidate solution's x_i^{new-2} objective function value. (equation 28) .

Algorithm 2. Pseudo-code of POA.

1. Start POA.
2. Enter the details of the optimization issue.
3. Calculate the number of iterations (T) and the size of the POA population (N).
4. Calculating the objective function and initializing the pelican positions.
5. For $t = 1:T$
6. Create a random position for the prey.

7. For $I = 1:N$
8. Phase 1: Approaching the prey (period of exploration).
9. For $j = 1:m$
10. Use Equation (25) to determine the new state of the j th dimension.
11. End.
12. Use Equation (26) to update the i th population member.
13. Phase 2 (exploitation phase): Winging on the water's surface.
14. For $j = 1:m$.
15. Use Equation (27) to determine the new state of the j th dimension.
16. End
17. Use Equation (28) to update the i th population member.
18. End
19. Revise the best possible solution.
20. End.

3.5.3. Modified LONG SHORT-TERM MEMORY (MLSTM)

To enhance the identification of Parkinson's disease using LSTM networks, a Gaussian Weight-based Genetic Algorithm (GWGA) is integrated into the model optimization process. The vanishing gradient issue was addressed by a specific type of RNN known as LSTM, which may be useful for detecting long-term relationships in sequential data. Memory cells and input, forget, and output gates are examples of gating systems, are included into LSTM in contrast to conventional RNNs to regulate the information flow through the network [28]. The following equations 29 and 30 are used by a simple Recurrent Neural Network (RNN) to analyze an input sequence $x=(x_1,x_2,...,x_T)$ and iteratively calculate the Hidden State (HS) sequence $h=(h_1,h_2,...,h_T)$ and the Output Sequence $y=(y_1,y_2,...,y_T)$ from $t=1$:

$$h_t = \mathcal{H}(W_{xh}h_t + W_{hh}h_{t-1} + b_h) \quad (29)$$

$$y_t = W_{hy}h_t + b_y \quad (30)$$

Here, $\mathcal{H}$ represents the activation function of the Hidden Layer (commonly a sigmoid function applied element-wise), b terms are bias vectors (e.g., hidden bias vector is b_h), and W terms represent weight matrices (e.g., the input-hidden weight matrix is represented as W_{xh}). While standard RNNs are effective for sequence modelling, they often struggle with long-term dependencies. Through the use of memory cells that preserve information over lengthy periods, LSTM networks get around this restriction. The LSTM cell updates are defined by the following set of equations (31 to 35).

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (31)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (32)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (33)$$

$$O_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \quad (34)$$

$$h_t = O_t \tanh(c_t) \quad (35)$$

In this formulation:

- i_t is the input gate,
- f_t is the forget gate,
- O_t is the output gate,
- c_t is the cell state,
- σ represents the sigmoid function.

The diagonal weight matrices (e.g., W_{ct}) connect the cell state to gates element-wise, ensuring that each gate receives input only from its corresponding cell unit. A significant limitation of conventional RNNs is their reliance solely on past context; they cannot capture future context. This is particularly restrictive in tasks like speech recognition, where the entire sequence is known beforehand.

To address this, Bidirectional RNNs (BRNNs) utilize two separate hidden layers:

- A **Forward Layer (FL)** processes the input from $t=1$ to T ,
- A **Backward Layer (BL)** processes it in reverse from $t=1$ to T .

The two hidden layers' outputs are then combined to get the final output:

$$\vec{h}_t = \mathcal{H}(W_{x\vec{h}}x_t + W_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}}) \quad (36)$$

$$\overleftarrow{h}_t = \mathcal{H}(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}) \quad (37)$$

$$y_t = W_{\vec{h}y}\vec{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y \quad (38)$$

By combining RNNs with LSTM cells, the Bidirectional LSTM (Bi-LSTM) is able to leverage long-range dependencies from both past and future contexts within the input sequence (equation 36 to 38). To enhance model capacity, deep architectures are commonly employed. This stacking approach uses the output of one RNN hidden layer as the input for the subsequent layer. The hidden state for layer n of a network with N layers at time t is determined using equation 39:

$$h_t^n = \mathcal{H}(W_{h^{n-1}h^n}h_t^{n-1} + W_{h^n h^n}h_{t-1}^n + b_h^n) \quad (39)$$

Here $h^0 = x$ is defined. y_t is the network output denoting in equation 40. Then,

$$y_t = W_{h^{N_y}} h_t^N + b_y \quad (40)$$

By substituting the forward and backward sequences, $\vec{h}^n$ and $\overleftarrow{h}^n$, for each hidden sequence h^n , and making sure that each neuron accepts input from the FL and BL at the lower level, LSTM can be created. LSTM model optimization in the context of PD diagnosis is greatly assisted by genetic algorithms [29]. To ensure that the model performs as well as possible, the number of hidden units, learning rate, batch size, and dropout rates are among the hyperparameters of the LSTM network that may be changed using GA. By encoding these hyperparameters as genes in chromosomes, the GA searches for the optimal combination that maximizes diagnostic accuracy and minimizes prediction error. This optimization approach significantly enhances the LSTM's ability to learn temporal patterns from biomedical signals, such as speech recordings, gait data, or tremor measurements, leading to more accurate and reliable Parkinson's disease diagnosis.

The Genetic Algorithm (GA), a successful stochastic optimization technique, is based on the concepts of natural selection and genetics. It is often used to resolve challenging optimization issues that are nonlinear, multi-modal, and difficult to address using traditional methods. GA operates by mimicking the evolutionary process, where generations of chromosomes a population of possible answers evolve toward superior solutions. Depending on the type of issue, each chromosome represents a potential solution that is represented in different forms, such as binary characters, real values, or permutations.

A random population is initialized at the beginning of the GA procedure. A fitness function, which measures how effectively the solution addresses the issue, is used to assess each member of the population. Better fitness scores increase the probability of selecting for reproduction in a specified population. To increase the likelihood that superior solutions would pass on their genetic material to the next generation, the selection process is often carried out utilizing strategies includes selection based on rank, selection based on the roulette wheel, and selection based on tournaments.

Once selected, additional offspring are produced by using genetic operators like crossover and mutation. By fusing the genetic material of two parent chromosomes, the crossover operator generates one or more offspring and encourages the solution space to be explored. Two-point, uniform, and single-point crossover are among the several crossover strategies that are used. To preserve population variety and prevent an early convergence to local optima, mutations cause tiny, random alterations to individual chromosomes. For real-valued parameters, Gaussian weight is often employed, where the data is refined by the addition of a small value taken from a Gaussian distribution. The Gaussian weight is mathematically expressed as in equation 41,

$$W_i = \exp\left(-\frac{(f_i - \mu)^2}{2\sigma^2}\right) \quad (41)$$

where f_i is the fitness of an individual, the standard deviation that regulates the distribution of selection pressure is denoted by σ , whereas μ is the mean fitness. Using replacement tactics like generational replacement or steady-state replacement, the older population is replaced by a new generation of people. Until a condition of termination is satisfied, such reaching a certain level of fitness or the maximum number of generations, this repeated process of selection, crossover, mutation, and replacement keeps on.

To be successful, GA has to be able to balance the exploration and exploitation of the search space. It can handle optimization problems that are highly nonlinear, non-differentiable, and complex with multiple constraints. However, GA also has some limitations, including the need for careful parameter tuning (such as the size of the population, the rate of crossover, and the rate of mutation) and higher computational costs when dealing with large populations or complex fitness evaluations. Despite these limitations, GA is highly flexible and can be hybridized with other algorithms for improved performance. Its application to Parkinson's disease detection exemplifies its potential to improve model performance, particularly when combined with LSTM.

The proposed average-based ensemble classification integrates EANFIS, DBN-POA, and M-LSTM to improve diagnostic accuracy. EANFIS effectively handles data uncertainty through fuzzy reasoning, while DBN, optimized by POA, captures deep hierarchical patterns. M-LSTM, enhanced with Gaussian-based Genetic Algorithm, efficiently models temporal dependencies in biomedical data. This ensemble framework achieves superior accuracy, robustness, and generalization compared to individual classifiers in Parkinson's disease detection.

4. Results And Discussion

The suggested ENSMOD model's results from experiments are shown in this section. Precision, Recall, F-Measure, and Efficiency are important evaluation measures used to assess the way the model detects Parkinson's disease. The model's ability to classify data is thoroughly evaluated using these measures. The performance of ENSMOD is compared against several existing methods, including LSVM, ANN, WRNN, HYMOD. As shown in Table 2, the proposed ENSMOD model consistently outperforms these methods, demonstrating superior accuracy and robustness in classifying speech signals for early detection of PD.

Performance Metrics

1) Precision

The percentage of expected positive instances that turn out to be positive is referred to as precision. It answers the question: “Among all instances classified as positive, how many are truly positive?” High precision reflects a low false positive rate, which is critically important in medical diagnostics to avoid incorrectly identifying healthy individuals as having Parkinson’s Disease (PD). In essence, precision measures (equation 42) positive prediction precision by showing the proportion of relevant instances among the retrieved instances. Precisions, which are defined as the proportion of results that are relevant,

$$\text{Precision} = \frac{\text{Truepositive}}{\text{truepositive} + \text{falsepositive}} \quad (42)$$

F) Recall

Recall, sensitivity, or true positive rate is the proportion of true positives the model identifies. It responds to the question “Out of all the actual positive instances, how many were accurately detected by the model?” Achieving high recall is crucial in medical diagnostics, particularly for Parkinson’s Disease, as it ensures that the majority of affected individuals are correctly diagnosed, thereby minimizing the occurrence of false negatives. Recall denotes in equation 43.

$$\text{Recall} = \frac{\text{Truepositive}}{\text{truepositive} + \text{FalseNegative}} \quad (43)$$

In disease detection scenarios, **recall plays a critical role**, as failing to identify a positive case (**false negatives**) can lead to serious or even life-threatening consequences

F) Accuracy

Accuracy is a measure of the number of accurately predicted instances including true positives (TP) and true negatives (TN) there were out of all the predictions made, which indicates how successful the model is overall. Equation 44 offers a broad evaluation of both positive and negative classes’ performance of the model. Accuracy alone, however, might be deceptive in situations involving unbalanced datasets where one class far dominates the other, because it was unable to accurately depict recognizing instances of the minority class using the model. When the dataset is balanced, accuracy serves as a helpful indicator of the model’s overall correctness.

$$\text{Accuracy} = \frac{\text{Truepositive} + \text{TrueNegative}}{\text{Total}} \quad (44)$$

4) F measure

An accurate assessment of a model’s accuracy is provided by the F1-Score, a performance statistic that integrates precision and recall into a single score, especially when dealing with unbalanced datasets. It is extremely beneficial in cases when both false positives and false negatives have significant consequences. To guarantee that accuracy and recall contribute equally to the final result, the F1-result (equation 45) is the harmonic mean of these two measures. The model does well in terms of both when its F1-Score is high capturing true positive instances (recall) and minimizing false positives (precision). This metric is highly valuable in medical diagnostics like Parkinson’s disease detection, where achieving a balance between correctly identifying patients and avoiding misclassification is crucial. A higher *F-measure* indicates that the model has avoiding false positives while yet capturing true positives.

$$\text{F-Measure} = 2 \times [(\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})] \quad (45)$$

TABLE 2. RESULTS OF PERFORMANCE COMPARISON

Metrics	Methods				
	LSVM	ANN	WRNN	HYMOD	ENSMOD
Accuracy (%)	90	96.7	98	99	99.45
Precision (%)	74	82	90	95	97
Recall (%)	88	92.42	93	95	98
F-measure (%)	76	87.01	89	93	97

TABLE 3. CONFUSION MATRIX EVALUATION OF ENSMOD

Class	Predicted 0	Predicted 1
Actual 0	515	4
Actual 1	5	516

Table 3. shows the confusion matrix evaluation of ENSMOD. Figure 4 illustrates the confusion matrix comparison of classes for the ENSMOD method. This matrix clearly demonstrates the model's strong classification capability, accurately identifying 515 instances of Class 0 and 516 instances of Class 1. The minimal number of misclassifications—only 4 false positives and 5 false negatives—indicates a highly reliable prediction outcome. With these values, the ENSMOD model achieves an impressive overall accuracy of 99.13%, confirming its effectiveness in correctly distinguishing between the two classes.

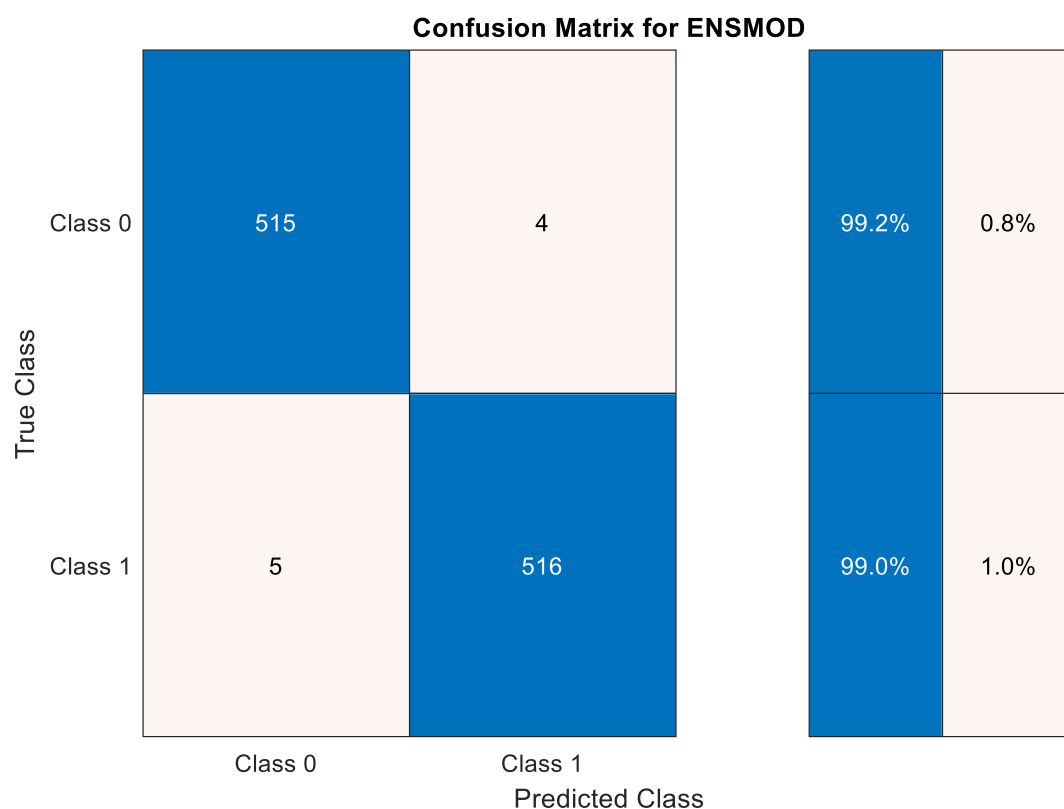


FIGURE 4. CONFUSION MATRIX COMPARISON OF CLASSES

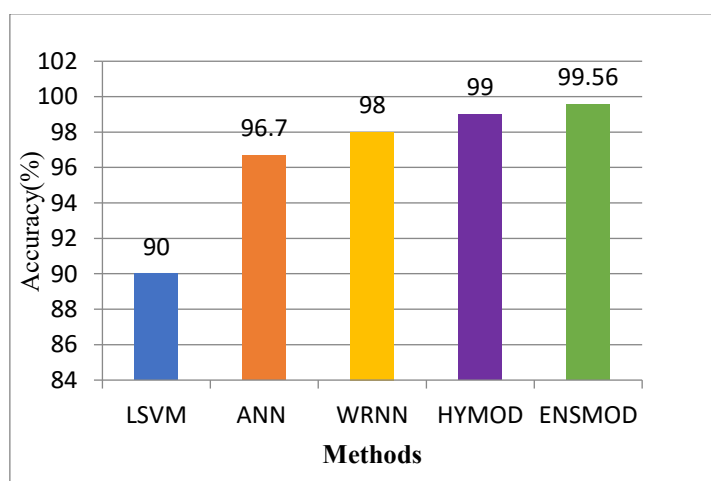


FIGURE 5. ACCURACY RESULTS COMPARISON

The accuracy rates of the suggested ENSMOD model are compared in Figure 5 and existing methods LSVM, ANN, WRNN, and HYMOD for Parkinson's disease classification. The Y-axis represents accuracy percentages, while the X-axis denotes the different methods. As per the results, LSVM achieves 90%, ANN reaches 96.7%, and WRNN attains 99% accuracy. The previous HYMOD model also records 99%, while the proposed ENSMOD outperforms all with a slightly higher accuracy, indicating enhanced generalization and prediction capability.

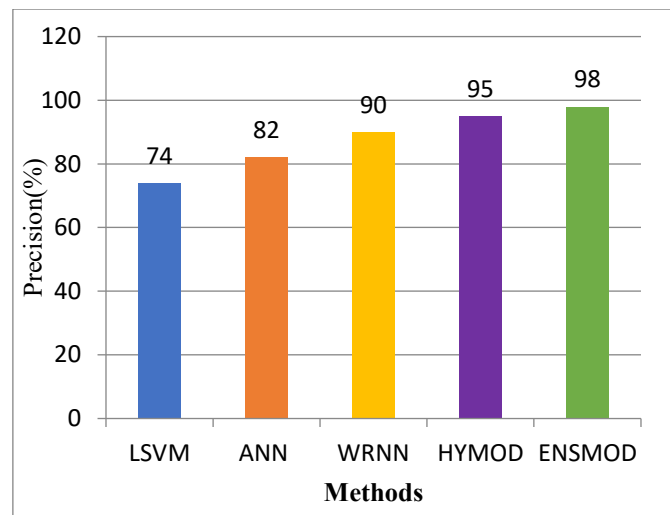


FIGURE 6. PRECISION RESULTS COMPARISON

The accuracy comparison of the suggested ENSMOD model is shown in Figure 6 and existing methods including LSVM, ANN, WRNN, and HYMOD. The X-axis represents the methods, and the Y-axis shows precision percentages. The ENSMOD model achieves the highest precision exceeding 95%, reflecting its effectiveness in minimizing false positives. In contrast, LSVM yields 74%, ANN records 82%, WRNN achieves 90%, and HYMOD delivers precision over 95%, yet ENSMOD demonstrates superior consistency and robustness.

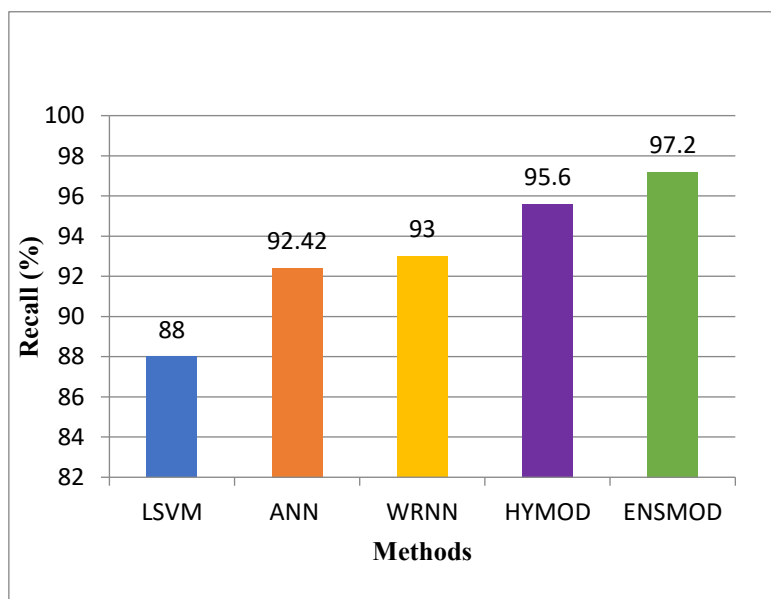


FIGURE 7. RECALL RESULTS COMPARISON

Figure 7 displays the comparison of recall values among the ENSMOD model and the existing LSVM, ANN, WRNN, and HYMOD methods. The Y-axis shows recall percentages, while the X-axis represents various models. The ENSMOD model demonstrates a recall rate of approximately 95%, emphasizing its ability to correctly detect positive cases. In comparison, LSVM records 88%, ANN reaches 92.42%, WRNN shows 95.6%, and HYMOD achieves 95%. Despite the competitive performance of WRNN and HYMOD, ENSMOD sustains high recall with improved stability.

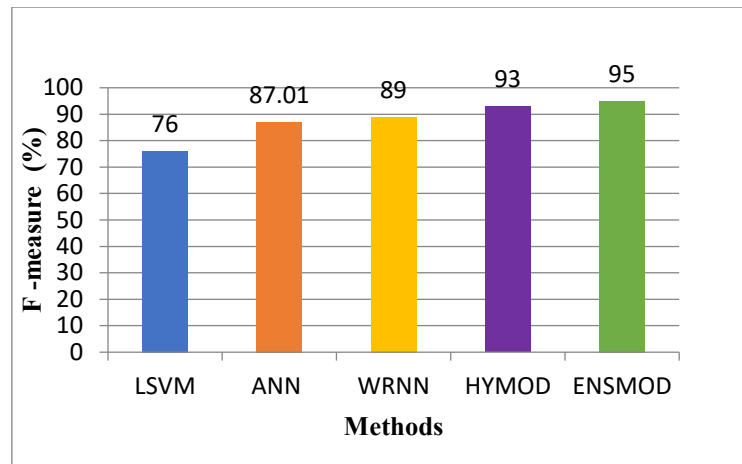


FIGURE 8. F -SCORE RESULTS COMPARISON

Figure 8 illustrates the F-measure performance comparison between the proposed ENSMOD model and existing models LSVM, ANN, WRNN, and HYMOD. The X-axis indicates the models, while the Y-axis shows the F-measure percentages. The ENSMOD model secures the highest F-measure score close to 95–96%, indicating an optimal balance between precision and recall. LSVM records the lowest at 76%, followed by ANN at 87.01%, WRNN at 89%, and HYMOD at 93%. ENSMOD's superior F-measure highlights its balanced and reliable diagnostic performance for Parkinson's disease classification.

Conclusion

The proposed ensemble-based diagnostic model demonstrates remarkable effectiveness in accurately detecting Parkinson's Disease by addressing challenges like class imbalance, outlier sensitivity, and irrelevant features. The combination of SMOTE, hybrid normalization, and IWOA-based feature selection significantly enhances data quality and model performance. The ensemble classifier, integrating EANFIS, DBN-POA, and M-LSTM, successfully captures both deep feature representations and sequential dependencies. Experimental evaluations confirm that the model outperforms existing techniques across all key metrics. This proves the model's robustness and suitability for early PD detection. In future work, the model can be extended to detect other neurological disorders such as Alzheimer's or Multiple Sclerosis. Additionally, optimizing the model for lightweight deployment on mobile or edge devices can enable practical, real-time clinical applications.

Reference

1. Bloem, B.R., Okun, M.S. and Klein, C., 2021. Parkinson's disease. *The Lancet*, 397(10291), pp.2284-2303.
2. Blauwendraat, C., Nalls, M.A. and Singleton, A.B., 2020. The genetic architecture of Parkinson's disease. *The Lancet Neurology*, 19(2), pp.170-178.
3. Ben-Shlomo, Y., Darweesh, S., Llibre-Guerra, J., Marras, C., San Luciano, M. and Tanner, C., 2024. The epidemiology of Parkinson's disease. *The Lancet*, 403(10423), pp.283-292.
4. Jankovic, J. and Tan, E.K., 2020. Parkinson's disease: etiopathogenesis and treatment. *Journal of Neurology, Neurosurgery & Psychiatry*, 91(8), pp.795-808.
5. Höglinger, G.U., Adler, C.H., Berg, D., Klein, C., Outeiro, T.F., Poewe, W., Postuma, R., Stoessl, A.J. and Lang, A.E., 2024. A biological classification of Parkinson's disease: the SynNeurGe research diagnostic criteria. *The Lancet Neurology*, 23(2), pp.191-204.
6. Almasoud, A.S., Eisa, T.A.E., Al-Wesabi, F.N., Elsafi, A., Al Duhayyim, M., Yaseen, I., Hamza, M.A. and Motwakel, A., 2022. Parkinson's detection using RNN-graph-LSTM with optimization based on speech signals. *Comput. Mater. Contin.*, 72, pp.872-886.
7. Reshma, S., Chennakesavulu, M., Patil, S.S. and Lamani, M.R., 2024. Efficient feature fusion model with modified bidirectional LSTM for automatic Parkinson's disease classification. *International Journal of Information Technology*, 16(6), pp.3963-3971.
8. El-Hasnony, I.M., Barakat, S.I. and Mostafa, R.R., 2020. Optimized ANFIS model using hybrid metaheuristic algorithms for Parkinson's disease prediction in IoT environment. *IEEE Access*, 8, pp.119252-119270.
9. Pasha, A., Ahmed, S.T., Painam, R.K., Mathivanan, S.K., Mallik, S. and Qin, H., 2024. Leveraging ANFIS with Adam and PSO optimizers for Parkinson's disease. *Heliyon*, 10(9).

10. Anisha, C.D. and Arulanand, N., 2020, February. Early Prediction of Parkinson's Disease (PD) Using Ensemble Classifiers. In *2020 International Conference on Innovative Trends in Information Technology (ICITIIT)* (pp. 1-6). IEEE.
11. Berdjouh, C., Meftah, M.C.E., Laouid, A., Hammoudeh, M. and Kumar, A., 2023. Pelican Gorilla troop optimization based on Deep feed forward neural network for human activity abnormality detection in smart spaces. *IEEE Internet of Things Journal*, 10(21), pp.18495-18504.
12. Vats, S. and Mehta, S., 2024, August. The Impact of Deep Belief Networks on the Early Diagnosis of Parkinson's Disease. In *2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 980-984). IEEE.
13. Singh, H., Rai, V., Kumar, N., Dadheech, P., Kotecha, K., Selvachandran, G. and Abraham, A., 2023. An enhanced whale optimization algorithm for clustering. *Multimedia tools and applications*, 82(3), pp.4599-4618.
14. Camacho, L., Douzas, G. and Bacao, F., 2022. Geometric SMOTE for regression. *Expert Systems with Applications*, 193, p.116387.
15. Douzas, G. and Bacao, F., 2019. Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE. *Information sciences*, 501, pp.118-135.
16. Lanfranchi, F., Arnaldi, D., Miceli, A., Mattioli, P., D'Amico, F., Raffa, S., Donegani, M.I., Chiola, S., Massa, F., Pardini, M. and Di Raimondo, T., 2023. Different z-score cut-offs for striatal binding ratio (SBR) of DaT SPECT are needed to support the diagnosis of Parkinson's disease (PD) and dementia with Lewy bodies (DLB). *European journal of nuclear medicine and molecular imaging*, 50(4), pp.1090-1102.
17. Kanagaraj, S., Hema, M.S. and Gupta, M.N., 2020. NORMALISATION AND DIMENSIONALITY REDUCTION TECHNIQUES TO PREDICT PARKINSON DISEASE USING PPMI DATASETS. *Oxidation Communications*, 43(1).
18. Al Sayaydeha, O.N. and Mohammad, M.F., 2019, April. Diagnosis of the Parkinson disease using enhanced fuzzy min-max neural network and OneR attribute evaluation method. In *2019 International Conference on Advanced Science and Engineering (ICOASE)* (pp. 64-69). IEEE.
19. Shantal, M., Othman, Z. and Bakar, A.A., 2023. A novel approach for data feature weighting using correlation coefficients and min-max normalization. *Symmetry*, 15(12), p.2185.
20. Kanagaraj, S., Hema, M.S. and Gupta, M.N., 2020. NORMALISATION AND DIMENSIONALITY REDUCTION TECHNIQUES TO PREDICT PARKINSON DISEASE USING PPMI DATASETS. *Oxidation Communications*, 43(1).
21. Chaudhuri, A. and Mohdiwale, S., 2023, July. An Effective Feature Selection Technique for Detecting Parkinson's Disease Using Binary Whale Optimization Algorithm. In *2023 IEEE World Conference on Applied Intelligence and Computing (AIC)* (pp. 58-62). IEEE.
22. Al-Najjar, H., Al-Rousan, N. and Al-Najjar, D., 2024. Hybrid grey wolf and whale optimization for enhanced Parkinson's prediction based on machine learning models using biomedical sound. *Informatics in Medicine Unlocked*, 48, p.101524.
23. Tien Bui, D., Khosravi, K., Li, S., Shahabi, H., Panahi, M., Singh, V.P., Chapi, K., Shirzadi, A., Panahi, S., Chen, W. and Bin Ahmad, B., 2018. New hybrids of anfis with several optimization algorithms for flood susceptibility modeling. *Water*, 10(9), p.1210.
24. Daher, A., Yassin, S., Alsamra, H. and Abou Ali, H., 2021, December. Adaptive neuro-fuzzy inference system as new real-time approach for Parkinson seizures prediction. In *2021 4th International Conference on Bio-Engineering for Smart Technologies (BioSMART)* (pp. 1-4). IEEE.
25. Hannan, S.A., 2024. Advancing Parkinson's Disease Severity Prediction using Multimodal Convolutional Recursive Deep Belief Networks. *International Journal of Advanced Computer Science & Applications*, 15(2).
26. Nilashi, M., Ahmadi, H., Sheikhtaheri, A., Naemi, R., Alotaibi, R., Alarood, A.A., Munshi, A., Rashid, T.A. and Zhao, J., 2020. Remote tracking of Parkinson's disease progression using ensembles of deep belief network and self-organizing map. *Expert Systems with Applications*, 159, p.113562.
27. Jabari, M., Ekinci, S., Izci, D., Bajaj, M. and Zaitsev, I., 2024. Efficient DC motor speed control using a novel multi-stage FOPD (1+ PI) controller optimized by the Pelican optimization algorithm. *Scientific Reports*, 14(1), p.22442.
28. Balaji, E., Brindha, D. and Vikrama, R., 2021. Automatic and non-invasive Parkinson's disease diagnosis and severity rating using LSTM network. *Applied Soft Computing*, 108, p.107463.
29. Abd El Aal, H.A., Taie, S.A. and El-Bendary, N., 2021. An optimized RNN-LSTM approach for parkinson's disease early detection using speech features. *Bulletin of Electrical Engineering and Informatics*, 10(5), pp.2503-2512.