Big Data and Information Analytics

BDIA, 10: 29–53.
DOI:10.3934/bdia.2026002
Received: 01 December 2025
Accepted:07 December 2025
Published: 12 January 2026

https://www.aimspress.com/journal/BDIA

Research article

A deep learning approach for Parkinson's disease diagnosis based on acoustic signals: LSTM with Bayesian and chaotic optimization

Huanshuo Zhang and Lijun Pei*

School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, China

* **Correspondence:** Email: peilijun@zzu.edu.cn.

Abstract: Current diagnostic methods for Parkinson's disease primarily rely on medical observations and the assessment of clinical signs. However, due to the mild nature of early symptoms, their overlap with those of other diseases, and the lack of distinct clinical markers, these early signs are often overlooked or misdiagnosed, thus making the early diagnosis of Parkinson's disease particularly challenging. This study aims to achieve a non-invasive and efficient early diagnosis of Parkinson's disease using deep learning methods based on acoustic signals, while enhancing the model's performance through hyperparameter optimization. All datasets in this study are sourced from publicly available online resources. After extracting multiple acoustic features, the data are merged into a unified dataset for model training. Among the four mainstream neural networks evaluated, long short-term memory (LSTM) demonstrates the best performance, thereby achieving an accuracy of 92.87%, a precision of 95.90%, a recall of 89.57%, and an F1-score of 92.57%. Following hyperparameter optimization using a combined Bayesian and chaotic optimization (BOCO) approach, the LSTM model's accuracy is improved to 94.22%, precision to 97.79%, recall to 90.52%, and F1-score to 93.97%. Consequently, the proposed method in this paper provides an innovative solution for a non-invasive, low-cost, and efficient diagnosis of Parkinson's disease, with broad application prospects.

Keywords: Parkinson's disease; acoustic signals; MFCC features; deep learning; LSTM; hyperparameter optimization

1. Introduction

Parkinson's disease (PD), also known as "paralysis agitans", is a common neurodegenerative

disorder of the nervous system. The primary cause of PD is the degeneration and death of dopaminergic neurons in the substantia nigra. The pathogenesis may involve genetic factors, environmental influences, and the age-related degeneration of the nervous system [1]. PD is widely recognized in the medical community as being strongly associated with aging, which is considered as the most significant risk factor for its development. The disease exhibits a marked age-related prevalence, predominantly affecting older adults. In addition, the incidence of PD is slightly higher in males than in females. It ranks as the fourth most common neurodegenerative disorder among the elderly, with approximately 1% of individuals aged 65 and older affected. In those over 40 years old, the prevalence drops to 0.4%. Despite this, PD can also occur in juvenile or adolescent cases. According to statistics, the annual incidence of PD ranges from 10 to 21 cases per 100,000 individuals [2].

The harm of PD primarily manifests as a severe decline in the patient's quality of life, with ultimately many losing the ability to care for themselves and experiencing complications such as injuries, psychological disorders, and cognitive decline. Research indicates that the 10-year mortality rate for PD patients is 47.9%, which is significantly higher than the 20.3% observed in non-PD controls [3]. According to predictions by Chinese scholars, the global number of PD patients is projected to reach 25.2 million by 2050 [4]. Although there is currently no curative treatment for PD, active therapy can alleviate symptoms. Therefore, the timely diagnosis of PD is critically important.

As shown in Figure 1, the clinical diagnosis of PD mainly relies on four core motor symptoms: resting tremor, rigidity, bradykinesia, and postural and gait abnormalities. About 70% of patients will present with resting tremor in the early stage of the disease; however, as the condition worsens, the tremor may gradually weaken. Early specific signs such as a reduced blink reflex, loss of facial expression (mask-like face), and decreased motor coordination are usually warning signals of the disease. It is worth noting that if a patient only shows tremors without other major symptoms, diseases such as essential tremors should be considered for a differential diagnosis, or dynamic follow-ups should be conducted to observe the appearance of other new signs. Currently, a clinical diagnosis mainly depends on symptom assessment, and auxiliary examinations are used to either rule out other neurological diseases (such as brain MRI, blood tests) or support the diagnosis (such as DaT-SPECT imaging). However, the most challenging task is to distinguish PD from essential tremors, as the symptoms significantly overlap, often requiring a comprehensive judgment based on the disease course and auxiliary examination results [5]. The main reasons for the difficulty in diagnosis are as follows. On the one hand, the disease insidiously progresses, and symptoms such as hand tremors and slow movements often gradually worsen. In the early stage, they have a relatively small impact on daily life, thus leading to delayed medical visits. Studies show that the average diagnostic delay is 6 to 18 months. On the other hand, the non-motor symptoms in the prodromal stage, such as hyposmia, constipation, rapid eye movement (REM) sleep behavior disorder, anxiety, and depression, are not specific, so they are easily misdiagnosed as other systemic diseases. For instance, 90% of patients develop hyposmia before the appearance of motor symptoms; however, this symptom is often attributed to aging or nasal problems. Therefore, clinicians need to adopt a multi-dimensional assessment approach, thereby combining the dynamic changes of both motor and non-motor symptoms, to accurately make a differential diagnosis [6].

In the diagnosis of PD, despite some progress made by existing methods, there is still a lack of an efficient, non-invasive, and convenient means of disease prediction. Therefore, this article will focus on exploring how to develop a prediction method suitable for PD.

With the continuous development of technology, machine learning has played an important role in the diagnosis of numerous diseases. Banerjee's team focused on exploring the application of machine learning to analyze high-dimensional datasets, especially rare diseases, for small sample datasets [7]. Arumugam's team finely tuned the decision tree model, thereby improving the ability to predict whether diabetes patients have heart disease risk [8]. Bhatt's team found that multi-layer perceptrons with cross validation outperformed all other algorithms in terms of the accuracy, thereby achieving the highest accuracy of 87.28% in predicting heart disease [9]. Rashid conducted a COVID-19 investigation, prediction, and discrimination using supervised learning methods, with an accuracy rate of 92.9% tested [10]. Islam predicted chronic kidney disease and found the most suitable classifier-XGBoost classifier. The maximum performance indicators of this classifier was an accuracy of 0.983 [11]. Ogunpola's team used the XGBoost model for the early detection of heart disease, especially myocardial infarction, and achieved an accuracy of 98.50% [12]. Govindu's team used a random forest classifier model in remote healthcare to detect early-stage PD, with an accuracy of 91% and a sensitivity of 95% [13]. The above examples fully demonstrate the application of machine learning in disease diagnoses.

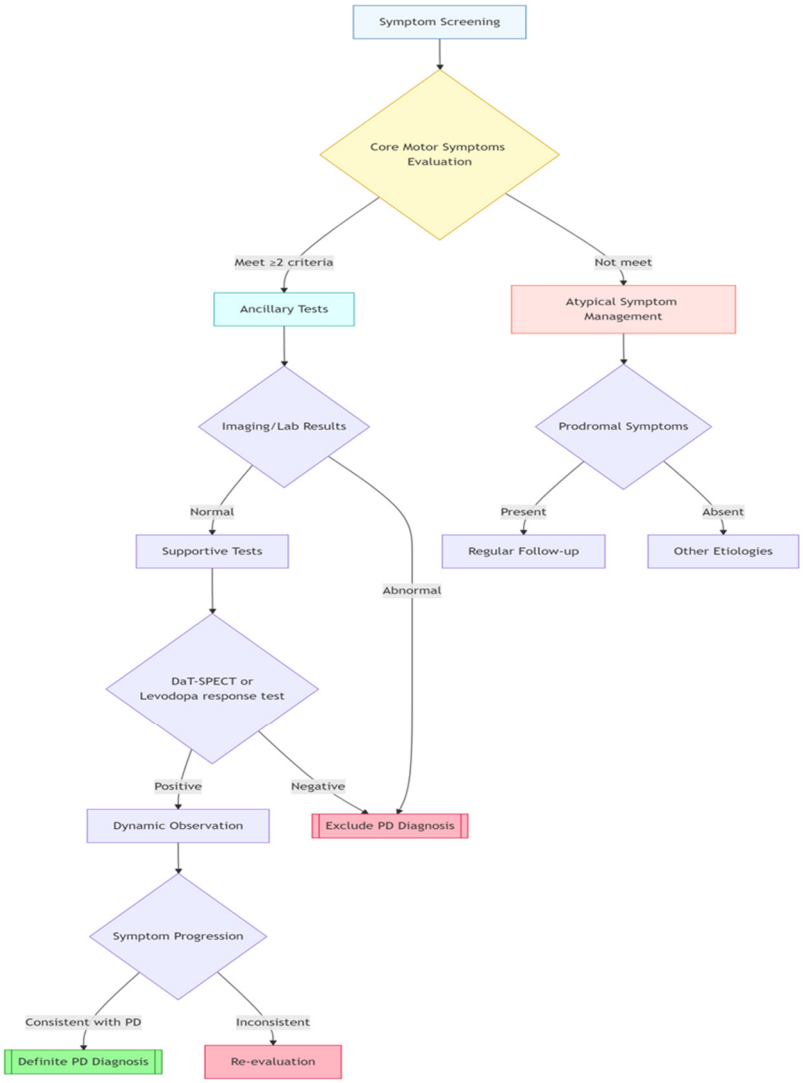


Figure 1. The diagnostic process for PD.

Currently, the application of machine learning in disease diagnoses has entered a stage of in-depth development, and deep learning technology, with its powerful feature abstraction capability, is driving the evolution of medical prediction models towards a higher precision and intelligence.

By analyzing a vast amount of medical data, including a patient's genetic information, lifestyle habits, medical history records, etc., deep learning models can construct accurate disease prediction models. Placido's team tested the prediction of cancer occurrence within incremental time windows based on disease code sequences in clinical histories, and found that the optimal dynamic normal prototype refinement (DNPR) model significantly improved the ability to design realistic monitoring plans for high-risk patients based on test results at different times [14]. Inthiyaz's team proposed an automated image diagnosis system that employs advanced techniques such as convolutional neural networks (CNN) to extract features from images. The system used an algorithm with a softmax classifier to classify images, thereby demonstrating higher accuracy and faster result delivery [15]. Tanveer's team systematically reviewed key advancements in ensemble deep learning, thereby integrating multiple deep neural networks to enhance the prediction robustness, with a focus on analyzing the application rationale of its design elements (ensemble types, model heterogeneity, and multimodal data fusion strategies) in Alzheimer's disease diagnoses [16]. Groh's team improved the diagnostic accuracy of both specialists and general practitioners by over 33% through the introduction of a fairness-optimized deep learning system for decision support [17]. Chen's team proposed a deep learning-based detection method, CHDdECG, which enables the diagnosis by automatically extracting pediatric electrocardiograms and wavelet transform features, and integrating them with key human-interpretable conceptual features [18]. The team led by Jiménez proposed a deep learning model to identify Alzheimer's disease from clinical records, which also demonstrated significant advantages [19]. Mandava's team proposed an innovative deep learning-based approach, which enhanced the cardiovascular disease prediction accuracy through deep feature mining. Their hybrid system achieved a prediction accuracy of 99.12% on the UCI dataset [20]. Celik established a deep learning model based on acoustic signals such as cough and breathing, which yielded outstanding performance in the detection of COVID-19 [21].

In the context of the aforementioned cases, while both machine learning and deep learning are widely applied in the medical field, no superior methods have been established to diagnose PD. Therefore, our study aims to develop a non-invasive, convenient, and efficient disease prediction approach.

The innovation of this research method is mainly reflected in the following two aspects:

First, this work adopts a non-invasive method for the PD diagnosis based on a voice signal analysis. Compared to traditional invasive diagnostic approaches, acoustic signals, as a non-invasive physiological feature, enable safer and more convenient disease detection. This method achieves an efficient and accurate diagnosis, thereby providing a feasible solution for large-scale screening and demonstrating significant potential for clinical application.

Second, by comparing the performance of four mainstream neural network models on the dataset, we select the optimal long short-term memory (LSTM) neural network and introduce two hyperparameter optimization methods—Bayesian optimization and chaotic optimization. By integrating these two approaches, we significantly improve the accuracy of the LSTM model, thereby enhancing its predictive power and diagnostic precision in PD detection tasks.

This paper is divided into five main sections; in Section 2, we outline the dataset and the quantitative processing applied to these data; Section 3 describes the models and methods used in our work; Section 4 presents the results of our proposed method alongside comparisons with other

approaches; Section 5 discusses the implications and limitations of our methodology and concludes this paper.

2. Dataset and processing

2.1. Description of the dataset

In the fields of machine learning and biomedical signal analysis, datasets as structured sample collections form the foundational core of this paper to achieve a non-invasive diagnosis of PD through acoustic signals. Our study integrates two open-source acoustic datasets to enhance the model's generalizability. Dataset 1 originates from the clinical voice repository of King's College London Hospital (KCL), containing voice samples from 21 healthy controls and 15 PD patients. The recording protocol simulates real-world telephone interaction scenarios: the participants placed a mobile device near their habitual ear and sequentially completed standardized vocal tasks. First, they read the standardized text "The North Wind and the Sun" aloud. Next, depending on the individual's language proficiency, the participants were instructed to repeat the technical phrase "Tech. Engin. Computer applications in geography". Then, they engaged in a semi-structured dialogue. Finally, the session concluded with a hang-up signal. Acoustic capture in this environment ensures that the acoustic signals remain within the near-field reverberation radius, thus meeting the clinical-grade voice analysis requirements. Dataset 2 is derived from the Neurology Department of Cerrahpaşa Faculty of Medicine, Istanbul University, and includes sustained phonation data of the vowel a from 188 PD patients, including 107 male and 81 female patients aged 33–87 years. Additionally, it contains the vowel a data from 64 healthy controls, including 23 males and 41 female healthy individuals aged 41–82 years. Professional recording equipment with a sampling rate of 44.1 kHz was used. Each participant performed three sustained vowel phonations in a soundproof booth under clinician supervision to ensure vocal stability.

Since Dataset 1 is an audio dataset, our team first performed a feature extraction on it. However, Dataset 2 had already undergone acoustic feature extraction, so we removed attributes irrelevant to the research objectives from Dataset 2. Finally, multi-source dataset fusion is conducted, thus resulting in a final dataset comprised of 85 healthy individuals and 203 patients. This study constructs an enhanced dataset with demographic diversity and acoustic scenario robustness, thus effectively mitigating sample bias caused by single-source data.

This study establishes a multidimensional feature parameter system through a quantitative analysis of acoustic signals, specifically encompassing temporal-domain stability metrics and spectral-domain characteristics. To address the periodic fluctuation properties of the signals, a dual-dimensional quantification model based on temporal-domain perturbations is developed. Aligned with the objectives of this paper, our team determines to adopt Jitter and Shimmer as key features:

(1) Jitter refers to the rate of change in the frequency of sound waves between adjacent cycles, thus reflecting differences in the vocal fold vibration frequency across cycles and indicating the irregularity in the vibration. Its feature set includes local percentage jitter (locPctJitter, quantifying the relative variability of fundamental frequency cycles), local absolute jitter (locAbsJitter, characterizing the absolute fluctuation amplitude of cycles), relative average perturbation (rapJitter), five-period phase jitter (ppq5Jitter), and differential period perturbation (ddpJitter). These parameters can effectively evaluate temporal periodic perturbations in acoustic signals.

(2) Shimmer refers to the rate of amplitude variation between adjacent cycles, thus reflecting differences in amplitude across vocal fold vibration cycles and indicating instability in vocal fold vibrations. Its feature set includes local amplitude fluctuation (locShimmer), decibel-scaled amplitude oscillation (locDbshimmer), three-cycle averaged amplitude variation (apq3Shimmer), five-cycle amplitude fluctuation (apq5Shimmer), extended-cycle amplitude variability (apq11Shimmer), and amplitude differential tremor (ddaShimmer). These parameters collectively quantify amplitude fluctuation characteristics in acoustic signals.

In the domain of frequency-domain feature engineering, this study employs Mel-frequency cepstral coefficients (MFCCs) to construct a 26-dimensional feature space, which is comprised of 13 coefficients (mean MFCC 0th–12th coefficients) and their standard deviations (std MFCC 0th–12th coefficients). The zeroth-order coefficient reflects the logarithmic energy distribution of the signal, while higher-order coefficients capture spectral envelope characteristics through nonlinear Mel-scale transformations. The standard deviation parameters characterize the dynamic variability of energy distribution across the frequency bands. This feature system deconstructs the spectral characteristics of acoustic signals via a time-frequency joint analysis. The jitter/shimmer metrics focus on evaluating the mechanical stability of vocal fold vibrations, whereas the MFCC feature matrix effectively characterizes vocal tract resonance properties. Together, they provide a multidimensional quantitative basis for the voice quality assessment, pathological diagnosis, and pattern recognition.

In the dataset used in this study, in addition to the aforementioned relevant feature values, there is also an identification document (ID)which indicates the serial number and a class which indicates whether the patient is sick. The above constitutes a specific description of the dataset. Following this description, we will outline the methods applied in the quantitative processing of the dataset.

2.2. Quantization processing of acoustic signals

In practical applications, raw voice signal data often suffer from contamination and information loss, thus requiring quantitative processing to enhance the analytical efficiency of machine learning models. To address environmental noise interference in PD acoustic signals, we employ the Wiener filtering algorithm for denoising [22–24]. Based on the minimum mean square error criterion, this algorithm achieves an optimal linear estimation of degraded observed signals and targets signals under stationary noise conditions. Subsequently, MFCC which incorporate dynamic characteristics are extracted as acoustic features, a method proven robust in pathological voice analyses. The specific workflow includes the following steps. First, a first-order high pass filter is used to implement pre-emphasis compensation to enhance the high-frequency components of the acoustic signals. This operation compensates for the high-frequency energy loss in sound wave propagation by increasing the high-frequency response by 6 dB/octave. Second, the signal is divided into frames with a frame length of 25 ms and overlap of 10 ms (corresponding to 400 sampling points at 16 kHz sampling); then, the windowed signal is subjected to a 1024 point fast Fourier transform (FFT) to obtain the power spectrum. The power spectrum calculation formula is as follows:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \tag{1}$$

which retains the first 257 frequency points (symmetry simplification) to obtain spectral representation. Then, using a Mel scale triangular bandpassfilter bank (Q filters, coverage range of 20–4000 Hz) for spectral smoothing, the linear frequency is converted to a Mel scale; then, 26 filters are distributed

at equal intervals in the Mel scale range of 20-4000Hz. The function of output logarithmic energy is as follows:

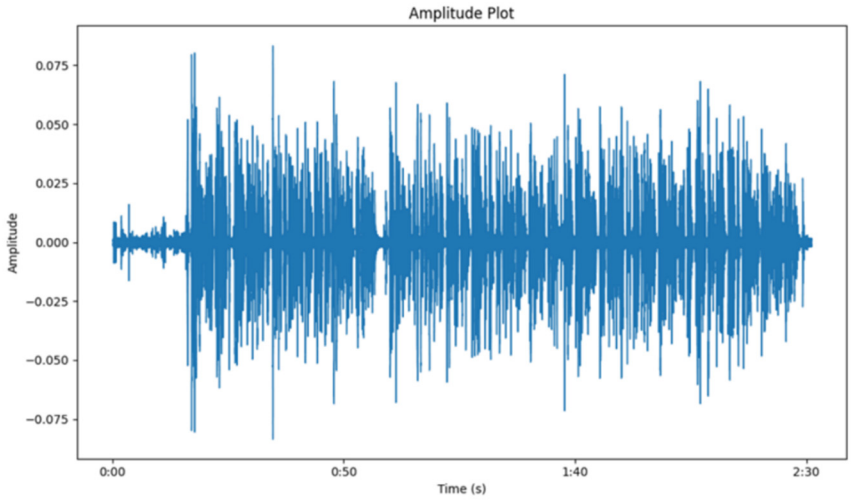
$$E_m = log(\sum_k |X[k]|^2 H_m(k)), \tag{2}$$

which is used to simulate the auditory characteristics of the human ear. Finally, the discrete cosine transform (DCT) is applied to the logarithmic filter energies, the output of the Mel filter banks is compressed via DCT, static MFCC coefficients are extracted, the first 13 dimensions (Q = 26) are retained to represent the vocal tract characteristics where higher-order coefficients reflect spectral details, and Q-dimensional static MFCC parameters per frame are generated. To capture dynamic speech patterns, the first-order (Δ) and second-order (ΔΔ) differential coefficients are further computed through a linear regression of adjacent frames, thus capturing time-varying features as follows:

$$\Delta t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2}. \tag{3}$$

A final N × 3Q-dimensional feature matrix is synthesized. Through column-wise mean compression, a 1 × 3Q-dimensional dynamic MFCC feature is obtained. This representation method effectively encodes time-frequency spectral evolution patterns while maintaining compatibility with machine learning models. The standardized dataset process through these steps provides a reliable foundation to develop supervised learning-based PD detection models.

As shown in Figure 2, the acoustic signals are quantized to obtain relevant feature parameters.



signal feature extraction

Ordinal number	locPctJitter	locAbsJitter	rapJitter	···	std_MFCC_10th_coef	std_MFCC_11th_coef	std_MFCC_12th_coef	class
1	–0.21492	-0.08887	-0.03571	···	-0.60708	-0.6344	-0.50967	1

Figure 2. Characteristic values of acoustic signals.

3. Methods and models

3.1. A Brief Introduction to 10 × 10-fold cross-validation

The division method and ratio of the training and test sets play a crucial role in model training, thereby directly impacting the model's training effectiveness, generalization performance, and even potentially leading to the occurrence of overfitting and underfitting situations.

In previous scientific experiments, the Holdout method was commonly used, which directly divides the dataset into two mutually exclusive sets at a predetermined ratio—one serving as the training set and the other as the test set. This division approach has a notable limitation: once the training-test ratio is set, there can be multiple ways to split the dataset. The model's training outcomes may significantly vary with different data splits, and in extreme cases, the model might never encounter samples of a certain category.

The evaluation metrics obtained through cross-validation training are generally more reliable than those from the Holdout method [25,26]. As shown in Figure 3, the 10-fold cross-validation method first divides the dataset into 10 mutually exclusive subsets of an approximately equal size. Each iteration uses the union of 9 subsets as the training set and the remaining subset as the test set. This process is repeated 10 times with different test subsets selected each time, thus ultimately returning the average of these 10 training results. This approach ensures that the model encounters all elements in the original dataset during training.

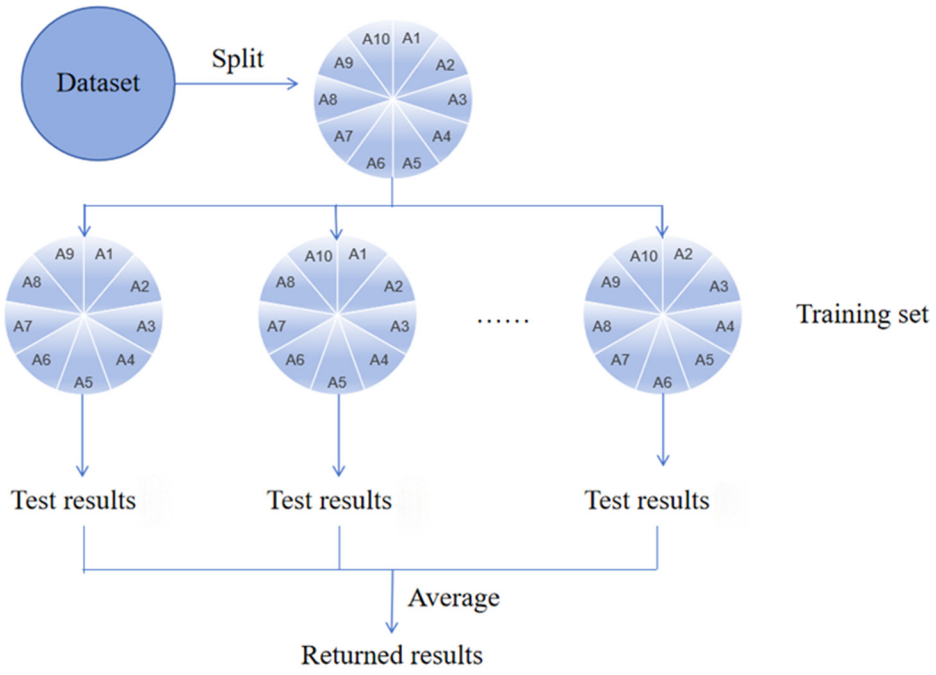


Figure 3. 10-fold cross-validation flowchart.

The 10 × 10-fold cross-validation extends 10-fold cross-validation by performing 10 distinct data splits, each followed by a 10-fold cross-validation process. Compared to the direct holdout method, this approach is better suited for small datasets in this experiment, as it maximizes data

utilization for training and testing while minimizing the impact of random partitioning effects.

3.2. Neural network model

3.2.1. A brief introduction to multi-layer perceptron (MLP) neural network

MLP, extended from the perceptron concept, is characterized primarily by its multiple neuron layers, and is also referred to as an artificial neural network. An MLP typically consists of an input layer, several hidden layers, and an output layer [27–29]. In contrast, the perceptron contains only input and output layers, which together form a simple neuron model–essentially a single neuron structure considered is the predecessor of the MLP. By stacking multiple hidden layers, the MLP can learn complex nonlinear relationships from the data, making it applicable to various types of tasks such as classification, regression, and time series prediction.

The MLP refers to a neural network with at least an input layer (receiving data), intermediate layers (one or more hidden layers for computation), and an output layer (producing results). Every node in the network is fully connected to all nodes in adjacent layers. Therefore, the most critical component here is the dense layer. In a dense layer, the fully connected layer contains learnable parameters: a weight matrix w and a bias vector b of length m. The layer computes the output y from the input x. Common activation functions in hidden layers include the sigmoid and tanh functions, whose expressions are as follows:

$$sigmoid\ \ function\colon\ \sigma(x) = \frac{1}{1+\exp(-x)}, \tag{4}$$

$$tanh\ \ function\colon\ tanhx = \frac{sinhx}{coshx} = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}. \tag{5}$$

The training process of the MLP typically employs the Backpropagation algorithm and gradient descent method to optimize the network parameters. During training, forward propagation is first used to calculate the output of each layer, followed by the computation of the loss function (such as cross-entropy loss or mean squared error). Finally, through backward propagation, the error is propagated back from the output layer to the input layer, gradients are computed layer by layer, and the network's weights and biases are updated. In practical applications, the MLP remains one of the indispensable models in deep learning.

3.2.2. A brief introduction to recurrent neural network (RNN)

A recurrent neural network (RNN) is a class of artificial neural networks with internal cyclic connections designed to process sequential data [30–32]. Its most distinctive feature is the presence of loops within the network, thus enabling information to circulate and facilitate the storage and processing of sequential information. This is manifested through the network's ability to retain previous information and apply it to the current output calculations: hidden layer nodes become interconnected (rather than isolated), with each hidden layer inputs encompassing both the output from the input layer and the hidden layer output from the previous timestep. An RNN typically exhibits various architectures including one-to-one, one-to-many, many-to-one, and many-to-many structures. During computation, the hidden layer's state continuously updates with a timestep

progression, thus forming temporal dependencies. The output at each timestep depends not only on current inputs but also on historical input sequences, thus making the RNN particularly suitable for tasks that require temporal dependency modeling.

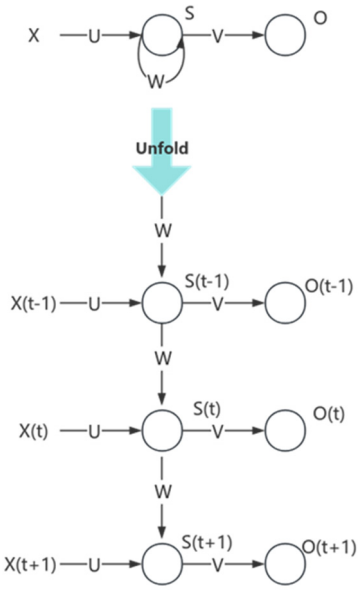


Figure 4. RNN structure.

As shown in Figure 4, the mathematical principle of an RNN can be described as follows. A unidirectional information flow travels from the input units to hidden units, while another unidirectional flow moves from the hidden units to output units. The hidden layer's input includes the previous hidden layer's state, meaning nodes within the hidden layer can have self-connections or interconnections. The structure on the right represents an unfolded version for computational clarity. Specifically, x denotes the state of the input layer, o denotes the state of the output layer, s denotes the state of the hidden layer, and the state of t indicates the computation timestep. With weight matrices V, W, and U, the hidden layer state at timestep t is computed as follows:

$$s_t = f(U * x_t + W * s_{t-1}). \tag{6}$$

Thus, it links the current input to previous computations.

By leveraging its powerful temporal modeling capabilities, the RNN has become a core tool to process sequence data and time-dependent problems. Despite advancements in deep learning technologies, the RNN remains an essential tool for sequence data modeling.

3.2.3. A brief introduction to long short-term memory (LSTM) neural network

The LSTM network is a special type of RNN architecture designed to learn long-term dependencies. As illustrated in Figure 5, its components primarily include the forget gate, input gate, output gate, and cell state [33–35]. This mechanism enables LSTM networks to effectively capture long-range dependencies within sequences. Unlike conventional RNNs, LSTM addresses the

vanishing gradient problem encountered during the training of long sequences. The core of LSTM lies in its cell state, which is regarded as an information conveyor belt that flows through the entire temporal sequence of the network. While the cell state maintains long-term memory, the gate mechanisms regulate the inflow, outflow, and degree of information retention. The forget gate determines which information in the cell state should be retained or discarded. The mathematical formulation of the forget gate is as follows:

$$f_t = \sigma(W_f[h_{t-1}\ , x_t] + b_f). \tag{7}$$

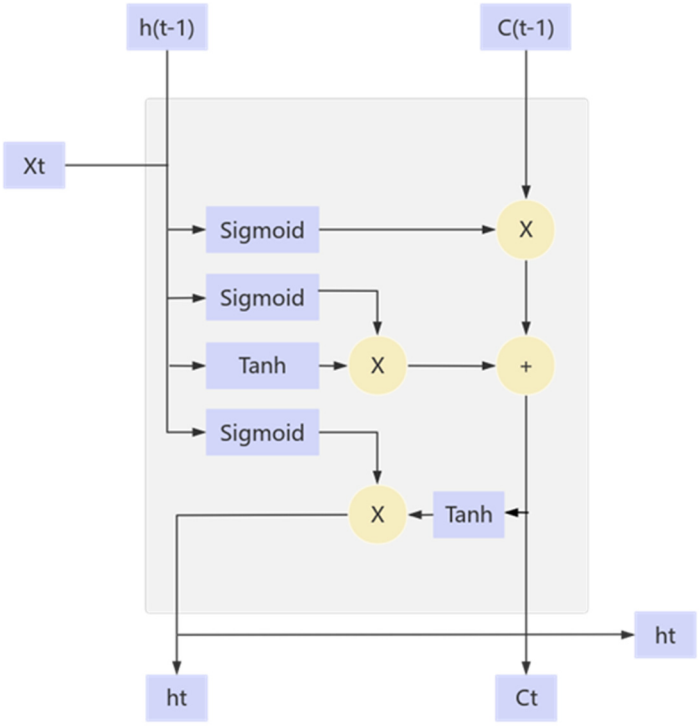


Figure 5. LSTM neural network structure.

In this context, W_f and b_f represent the weight matrix and bias vector of the forget gate, respectively. The forget gate receives the current and the hidden states from the previous timestep as inputs, thus generating a vector with values between 0 and 1. This vector indicates the "degree of forgetting" for each piece of information: values close to 0 signify discarding the current information, while values close to 1 signify retaining it. The input gate controls how much of the current input will be updated into the cell state. It consists of two components: one part calculates candidate values for new information, and the other part regulates the proportion of current information to be incorporated into the cell state. Its computational formula is as follows:

$$i_t = \sigma(W_i[h_{t-1}\ , x_t] + b_i), \tag{8}$$

$$\tilde{C}_t = tanh(W_c[h_{t-1}\ , x_t] + b_c), \tag{9}$$

where, W_i, b_i, W_c, and b_c are the weight matrices and bias vectors of the input gate, respectively. The output gate determines which information from the cell state will be output as the hidden state at the

current timestep. Its computational formula isas follows:

$$o_t = \sigma(W_o[h_{t-1}\ , x_t] + b_o), \tag{10}$$

$$h_t = o_t \odot tanh(C_t), \tag{11}$$

where, W_o and b_o are the weight matrix and bias vector of the output gate, respectively.

The uniqueness of the LSTM lies in its ability to meticulously control the transmission, thereby forgetting and updating of information through its cell state and three gated mechanisms (forget gate, input gate, output gate), thus effectively mitigating the vanishing gradient problem encountered by traditional RNNs when training long sequences. Due to its exceptional memory capacity and ability to learn long-term dependencies, the LSTM has become a standard model for processing sequential data and temporal problems, and is widely applied in fields such as natural language processing, speech recognition, and time series prediction. Despite advancements in deep learning technologies, the LSTM remains an indispensable core model to handle sequential data.

3.2.4. A brief introduction to gated recurrent unit (GRU) neural network

The GRU is a type of RNN designed to address issues in traditional RNNs, such as the inability to retain long-term memory and gradient problems during backpropagation. Similar to the LSTM, the GRU simplifies the architecture of the LSTM by introducing fewer parameters and a reduced structural complexity [36–38]. As a result, the GRU is simpler than the LSTM and is easier to train. The GRU has been widely applied in various fields, particularly in handling time series and sequential data.

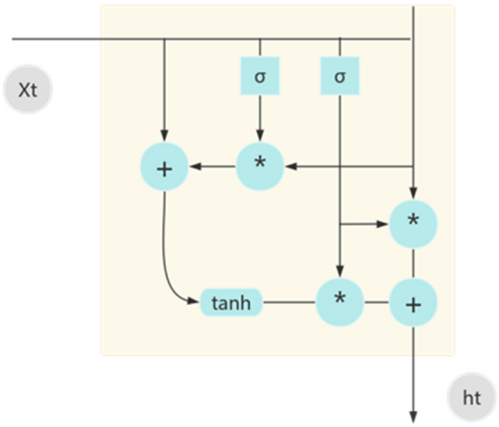


Figure 6. GRU neural network structure.

As shown in Figure 6, the structure of the GRU neural network is relatively simple, containing only two gates (the update and the reset gates). This simplified architecture allows the GRU to maintain its performance while improving its computational efficiency. The output of a GRU unit is jointly determined by these two gating mechanisms, thereby enabling the retention or discarding the input information as needed. In GRU networks, the hidden state not only preserves memory from previous parts of the sequence but also updates or resets information through gating mechanisms.

This design allows the network to retain long-term contextual information in sequence tasks while adapting to changes in the data. In the GRU model, at each timestep (t), the update gate (z_t) and reset gate (r_t) are computed through the following formulas:

$$z_t = \sigma(W_z \cdot x_t + U_z \cdot h_{t-1} + b_z). \tag{12}$$

The update gate controls the blending ratio of information between the previous hidden state and the current timestep's state. The reset gate determines the degree of "forgetting" applied to the previous hidden state, which is used to compute the candidate hidden state. The above are the main mathematical formulas for calculating the update and reset gates in the GRU model.

3.3. Optimization algorithm

3.3.1. Bayesian optimization algorithm

The Bayesian optimization algorithm is a gradient-free optimization method designed to find the global optimum of a function with minimal evaluations [39,40]. Its core idea involves constructing a surrogate model of the objective function and designing an intelligent acquisition function to approach the global optimal solution through limited iterations. The surrogate model in Bayesian optimization is typically a Gaussian process. Commonly used acquisition functions include the following: expected improvement (EI), which maximizes the expected improvement of new points compared to the current best value; upper confidence bound (UCB), which prefers regions of high mean or high uncertainty; and probability of improvement (PI), which calculates the probability that a new point is better than the current optimum. For example,

$$El(x) = (\mu(x) - f^+ - \xi)\phi(Z) + \sigma(x)\varphi(Z), \tag{13}$$

$$Z = \frac{\mu(x) - f^+ - \xi}{\sigma(x)}, \tag{14}$$

where φ and ϕ represent the cumulative distribution function and probability density function of the standard normal distribution, respectively.

3.3.2. Chaotic optimization algorithm

The chaotic optimization algorithm is a search method based on chaos theory, thereby leveraging characteristics of chaotic systems—such as randomness, regularity, ergodicity and sensitivity—to perform global optimization [41–43]. It is a novel global optimization algorithm that mimics the random motion of air molecules in its search process, thus ensuring at least a chance to reach the global optimum. The chaotic optimization algorithm is generally divided into two phases: the first phase is the global search phase, where the extensive exploration of the solution space is conducted using the traversability of chaotic variables to identify potential optimal regions; and the second phase is the local fine search phase, which involves applying perturbations (such as chaotic perturbations or Gaussian distributions) near the optimal region to further approach the optimal solution. In chaos theory, chaotic mapping often employs the Logistic map, which generates non-periodic, non-convergent sequences under specific parameters. Its mathematical expression is as follows:

$$x_{n+1} = \mu x_n(1 - x_n), \tag{15}$$

where, μ is the control parameter. When $\mu \in [3.5699456, 4]$, the system exhibits chaotic behavior.

In this paper, we plan to employ Bayesian optimization, chaotic optimization, and a hybrid strategy which integrates both approaches to respectively optimize the hyperparameters of the best-performing neural network, with the aim of further enhancing its performance.

4. Results and comparisons

4.1. Training results of the basic model

4.1.1. Comparison of results from distinct models

After extracting the required features from the audio datasets of PD patients and healthy individuals, we merge them with another feature text dataset to create a novel integrated dataset. In this experiment, we individually train four neural network architectures (GRU, RNN, MLP, and LSTM) to select the best-performing architecture as the base model. Subsequently, we apply Bayesian and chaotic optimization for hyperparameter tuning to compare the optimal combination of neural network architecture and hyperparameter optimization for the integrated dataset. The accuracy, precision, recall, and F1-score of these four neural networks on the test set are summarized in Table 1, ranked in ascending order based on the average values of accuracy and F1-score.

Table 1. Comparison of training results of LSTM, MLP, RNN and GRU models.

Number	Name	Accuracy	Precision	Recall	F1-Score
1	GRU	0.8729	0.9036	0.8381	0.8682
2	RNN	0.9037	0.9311	0.8729	0.9002
3	MLP	0.9196	0.9513	0.8851	0.9164
4	LSTM	0.9287	0.9590	0.8957	0.9257

The accuracy of a model on the test set is the proportion of correctly identified samples to the total number of samples, which most intuitively measures the model's ability to diagnose PD based on vocal features [44]. As shown in Table 1, the neural network with the highest accuracy on the test set is the LSTM neural network, thereby achieving an accuracy of 0.9287, which is approximately 0.01 higher than the second-best performing MLP neural network. The precision of a model on the test set is the proportion of correctly identified positive cases to all cases predicted as positive, thus reflecting the model's ability to accurately predict positive classes. The neural network with the highest precision on the test set is the LSTM neural network, thus achieving a precision of 0.9590, which is approximately 0.0077 higher than the MLP neural network. The recall of a model on the test set is the proportion of correctly identified positive cases to all true positive cases, thus reflecting the model's ability to comprehensively detect positive classes. The neural network with the highest recall on the test set is the LSTM neural network, thereby achieving a recall of 0.8957, which is approximately 0.01 higher than the MLP neural network. The F1-score is a composite metric that balances precision and recall. The neural network with the highest F1-score on the test set is the

LSTM neural network, thereby achieving an F1-score of 0.9257, which is approximately 0.01 higher than the MLP neural network.

Therefore, the LSTM neural network achieves the highest accuracy, precision, recall, and F1-score on the test set. The other neural networks showed significant gaps in these four evaluation metrics compared to the LSTM. The data demonstrates that the LSTM neural network holds a clear advantage in diagnosing PD through vocal features. Thus, this study adopts the LSTM as the foundational neural network architecture and performs hyperparameter optimization on the LSTM neural network in subsection 4.2.

4.1.2. Visual analysis of results—violin plot

A violin plot is a visualization chart that combines a box plot and kerneldensityestimation (KDE) to display data distribution, density, central tendency, and variability [45,46]. Given that our work employs a 10-times repeated, 10-fold cross-validation strategy, each trained neural network generates 100 sets of evaluation metrics from the test sets, including accuracy, precision, recall, and F1-score. Visualizing these metrics through violin plots effectively displays their distribution patterns, symmetry, central tendency, and outliers across all 100 evaluations.

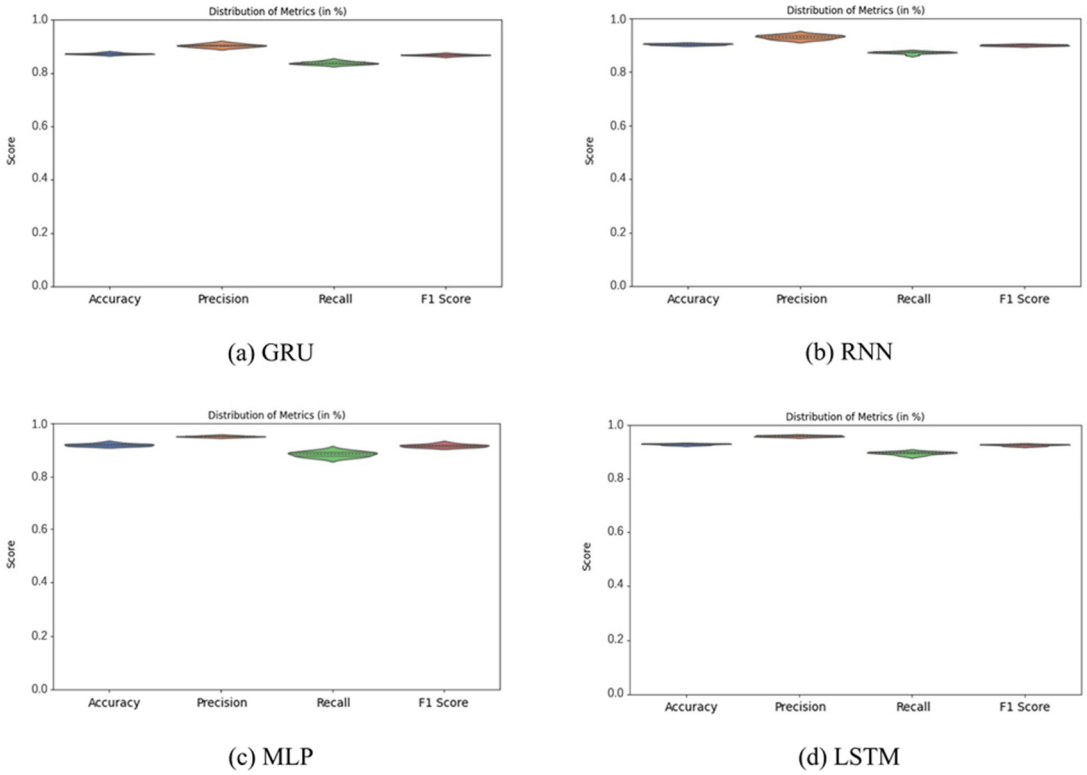


Figure 7. Comparison of violin plots of LSTM, MLP, RNN and GRU models.

In a violin plot, the length of a "violin bar" (the projection length of any metric bar mapped onto the y-axis) reflects the concentration of the data. The distribution shape and range of each metric can illustrate the variability of the model's performance on that metric and whether outliers exist. If the length of a "violin bar" for a specific metric is small, then it indicates that the scores for that metric

are mostly clustered within a certain range, thus demonstrating the consistent and reliable performance of the model on that metric. As shown in Figure 7, the violin plots generated by each trained neural network consist of four "violin bars" with colors (from left to right) of blue, yellow, green, and red, which correspond to the comprehensive performance of accuracy, precision, recall, and F1-score, respectively, across different folds (10-fold cross-validation) in the test sets. Compared with the other three neural networks, the LSTM neural network demonstrates significantly reduced violin lengths across all four evaluation metrics, thus indicating a higher concentration of its evaluation scores across different test folds, and consequently, a superior stability in our work. Simultaneously, the LSTM neural network exhibits noticeably greater violin heights for all four metrics compared to other architectures, thus reflecting its enhanced predictive performance. These visual characteristics collectively establish the LSTM neural network as the most stable and effective model among the four candidates for vocal signal-based PD diagnoses.

In a violin plot, the height of the "violin shape" reflects the magnitude of each evaluation metric. A taller "violin shape" for a specific evaluation metric of a neural network indicates a higher score for that metric on the test set, thus directly demonstrating the superior performance of the network on the test set. As clearly shown in Figure 7, all evaluation metrics of the LSTM neural network approach the peak values in the plots and are significantly higher than those of the other three neural networks for the same metrics. Thus, it can be concluded that the LSTM neural network outperforms the other three networks in diagnosing PD through vocal features.

By comparing two metrics, namely length and height in the violin plots of four neural networks, we conclude that the LSTM neural network demonstrates the most stable and superior performance among the four in to diagnose PD through vocal features. We further verify this conclusion using other visualization methods.

4.1.3. Visual analysis of results—receiver operating characteristic (ROC)curve

ROC curve was initially applied in a radar signal analysis and later introduced into the field of machine learning. In the ROC plot, the x-axis represents the false positive rate (FPR), which indicates the proportion of negative samples incorrectly predicted as positive. The y-axis represents the true positive rate (TPR), which is equivalent to the recall defined in subsection 4.1.1, thus reflecting the proportion of correctly identified positive samples. The closer the curve approaches the top-left corner of the graph, the lower the FPR and the higher the TPR, thus signifying a better model performance in distinguishing positive and negative samples [47,48]. The area under the curve (AUC) quantifies the area under the ROC curve. The AUC value ranges from 0 to 1, with values closer to 1 indicating that the ROC curve is near the top-left corner, thus demonstrating a superior model performance.

As clearly illustrated in Figure 8, the ROC curves of all four neural networks exhibit a concave shape, starting from the lower-left corner, rising rapidly toward the upper-left corner, and nearly leveling off at a certain point. Notably, compared to other neural network architectures, the ROC curve of the LSTM neural network is closer to the upper-left corner and achieves a near-horizontal state more quickly. It indicates that the LSTM model performs well across most decision thresholds, efficiently distinguishes positive and negative samples, and outperforms the other three neural network architectures.

Second, as shown in Figure 8, the AUC values for the four neural network architectures—GRU, RNN, MLP, and LSTM—are 0.93, 0.96, 0.97, and 0.97, respectively. Based on the magnitude of AUC values, the MLP and LSTM exhibit the best predictive performance in diagnosing PD through

vocal features. A further analysis of the ROC curve positions confirms that the LSTM neural network outperforms the other three architectures.

Finally, based on the numerical analysis of the accuracy, precision, recall, and F1-score across neural network architectures on the test set, along with visual data interpretation methods such as violin plots and ROC curves, we conclude that the LSTM neural network demonstrates the best performance among the four architectures for diagnosing PD through vocal features. Consequently, we adopt the LSTM as the foundational architecture and implement three hyperparameter optimization strategies: Bayesian optimization, chaotic optimization, and an integrated Bayesian-chaotic optimization approach.

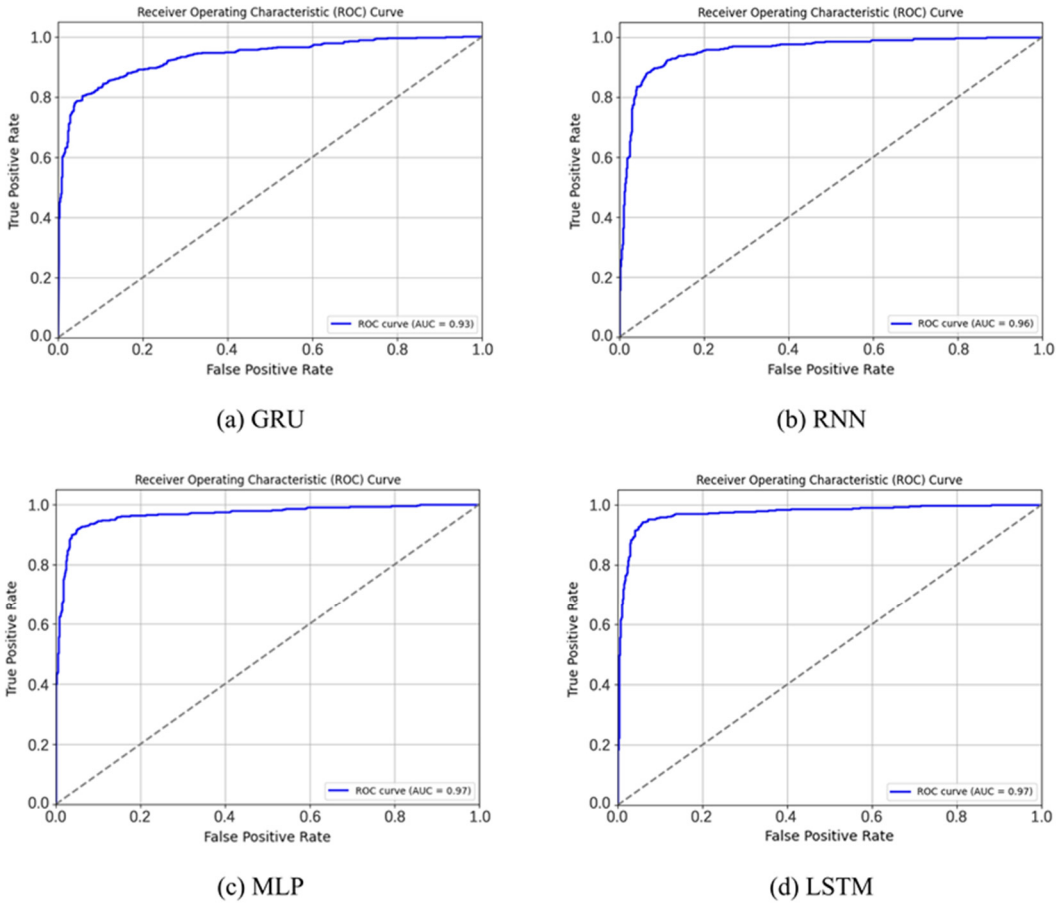


Figure 8. ROC curve comparison of LSTM, MLP, RNN and GRU models.

4.2. Hyperparameter optimization results

4.2.1. Comparison of optimization results

In this experiment, we apply three hyperparameter optimization methods to the LSTM neural network. First, we incorporate Bayesian optimization, and the resulting model is denoted as LSTM-BO. Second, chaotic optimization is applied, and the optimized model is named as LSTM-CO. Third, we integrate both Bayesian optimization and chaotic optimization for hyperparameter tuning within the LSTM architecture, with the combined model labeled as LSTM-BOCO. The accuracy,

precision, recall, and F1-score of these three optimized models and the unoptimized control group on the test set are summarized in Table 2, sorted in ascending order based on the average values of test-set accuracy and F1-score.

Table 2. Comparison of results for LSTM model with distinct optimization methods.

Number	Name	Accuracy	Precision	Recall	F1-Score
1	LSTM	0.9287	0.9590	0.8957	0.9257
2	LSTM-BO	0.9286	0.9636	0.8908	0.9258
3	LSTM-CO	0.9294	0.9605	0.8955	0.9264
6	LSTM-BOCO	0.9422	0.9779	0.9052	0.9397

As shown in Table 2, the LSTM neural network and LSTM-BO neural network exhibit nearly identical evaluation metrics on the test set, with no discernible improvement from the Bayesian optimization method applied to the LSTM. While the LSTM-CO neural network displays minor enhancements over the baseline LSTM, these results do not substantiate a significant benefit from chaotic optimization. Consequently, we integrate both Bayesian optimization and chaotic optimization methods for hyperparameter tuning of the LSTM. The experimental results demonstrate that the LSTM-BOCO neural network achieves an accuracy of 0.9422, a precision of 0.9779, a recall of 0.9052, and an F1-score of 0.9397 on the test set, thus outperforming the baseline LSTM by 0.0135, 0.0189, 0.0095, and 0.0140 in these metrics, respectively. It quantitatively validates the superiority of the LSTM-BOCO model in diagnosing PD through acoustic signals. A subsequent analysis will leverage three data visualization methods to further corroborate these findings.

4.2.2. Visual analysis of optimization results—violin plot

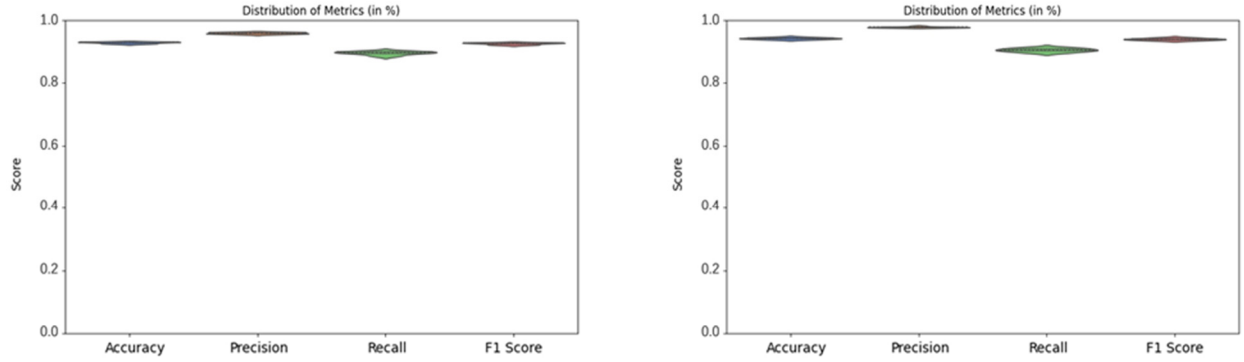
First we conduct a visual analysis by comparing the violin plots of the LSTM and the LSTM-BOCO neural networks. As shown in Figure 9, the violin plots for both neural networks consist of four "violin components," colored as blue, yellow, green, and red from left to right, which correspond to the comprehensive performance of accuracy, precision, recall, and F1-score on different test folds. Since the baseline LSTM already performs well in this task, the width and length of the "violin components" in LSTM-BOCO display minimal changes compared to those of the original LSTM. However, the height of the "violin components" in the violin plots exhibits notable differences. In Figure 9, the height of each evaluation metric's "violin component" for LSTM-BOCO is significantly greater than that of the unoptimized LSTM, particularly for the yellow component that represents precision. It indicates that LSTM-BOCO achieves higher scores across all evaluation metrics on the test set, thus demonstrating a superior performance.

4.2.3. Visual analysis of optimization results—ROC curves

Figure 10 clearly demonstrates that the ROC curves of both the LSTM and LSTM-BOCO neural networks exhibit a near-right-angled convex upward shape. Both curves rise sharply from the lower-left corner toward the upper-left corner and eventually plateau. However, the LSTM-BOCO neural network's curve nearly levels off at an FPR of 0.2, whereas the LSTM network's curve continues to rise slowly until reaching an FPR of 0.6. It indicates that the LSTM-BOCO neural

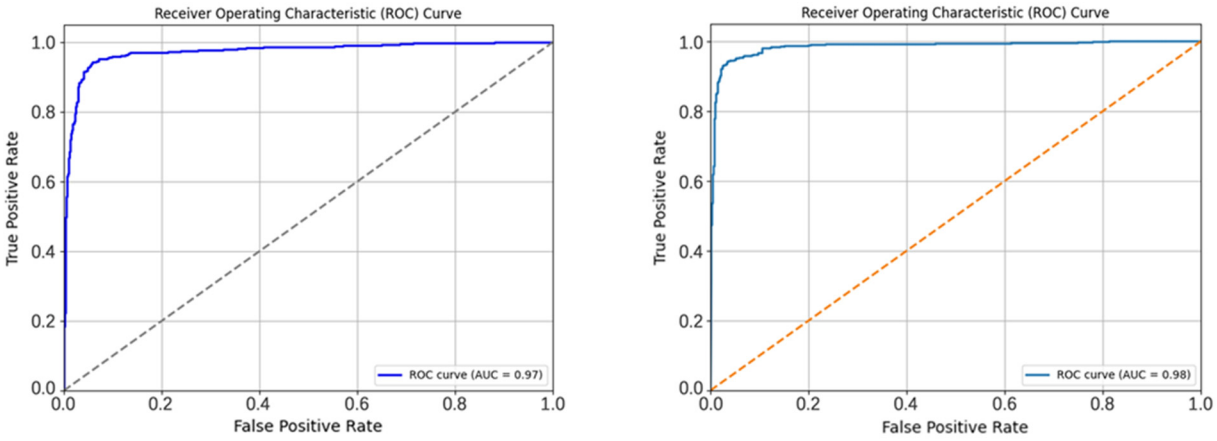
network achieves a stable performance faster across most decision thresholds, thus demonstrating a higher efficiency in distinguishing positive and negative samples.

Second, Figure 10 also reveals the AUC values of these two neural network architectures: the LSTM and LSTM-BOCO models achieve AUC values of 0.97 and 0.98, respectively. Based solely on the magnitude of the AUC values, it can be concluded that the LSTM-BOCO neural network demonstrates a superior predictive performance compared to the standard LSTM in our work.



(a) LSTM (b) LSTM-BOCO

Figure 9. Violin plot comparison of two LSTM models.



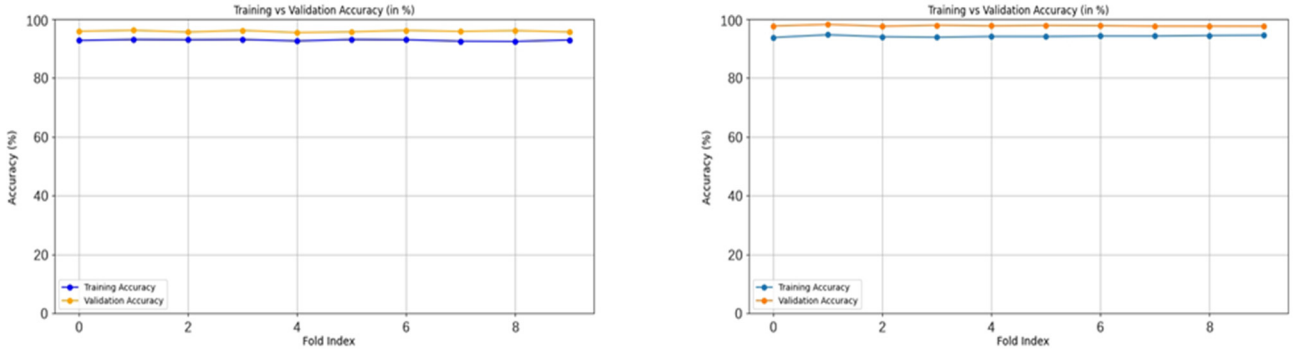
(a) LSTM (b) LSTM-BOCO

Figure 10. ROC curves comparison of two LSTM models.

4.2.4. Visual analysis of optimization results—comparison of accuracy between training and validation sets

The comparison of accuracy between the training and validation sets can reveal various model performance and help identify potential problems such as overfitting, underfitting, or others. First, an analysis in Figure 11 discloses that both the LSTM and LSTM-BOCO neural networks exhibit small accuracy differences between the training and validation sets, both below 5%, thus indicating that neither network suffers from overfitting or underfitting. Furthermore, LSTM-BOCO achieves a

higher accuracy than the LSTM on both the training and validation sets, with values exceeding 0.95. It demonstrates that the model fits well on both the training and the validation data, exhibits strong generalization capabilities, and can adapt effectively to new, unseen data.



(a) LSTM (b) LSTM-BOCO

Figure 11. Comparison of accuracy between training and validation sets.

Based on the numerical analysis of accuracy, precision, recall, and F1-score for hyperparameter-optimized neural network architectures on the test set, along with three visualization methods including violin plots, ROC curves, and a training-validation accuracy comparison, we conclude that the LSTM neural network demonstrates the best performance among the four architectures. Integrated Bayesian and chaotic optimization are the hyperparameter tuning methods that most significantly enhance the LSTM network's performance among the three optimization approaches.

5. Discussion and conclusions

In this work, we proposed an LSTM neural network enhanced by integrated Bayesian and chaotic optimization. This model accurately diagnosed PD by analyzing a patient's acoustic signals, thereby achieving high prediction accuracy. Traditional diagnostic methods for PD often suffer from invasiveness, inefficiency, and subjectivity, which not only increase patient discomfort but also raise healthcare costs. In contrast, our vocal signal-based approach is non-invasive, highly efficient, and accurate, thus offering a novel methodology for an early diagnosis and even prevention of PD. Particularly in settings that lack traditional diagnostic tools, this method holds significant potential for large-scale population screening.

In recent years, the rise of machine learning has enabled researchers and medical professionals to explore opportunities at the intersection of healthcare and machine learning. Significant progress has been made in disease diagnoses using machine learning techniques. For example, Govindu et al. trained four machine learning methods—SVM, random forest, KNN, and logistic regression—on MDVP voice data from 30 PD patients and healthy individuals. Their comparative analysis revealed that the Random Forest classifier is the ideal machine learning technique for PD detection, thereby achieving a detection accuracy of 91.83% [13]. In contrast, our work integrates a larger-scale dataset with hyperparameter optimization methods, which collectively enhanced the model's performance and ultimately achieved a diagnostic accuracy of 94.22%.

Hyperparameter optimization is critical to enhance the predictive performance. During the final hyperparameter tuning of the LSTM neural network in this paper, we observed that Bayesian or chaotic optimization alone yields limited improvements. However, when both optimization methods are combined, the LSTM model exhibits significant performance gains. We hypothesize that Bayesian optimization gradually approaches optimal solutions through a refined local search but struggles with exploration, thus often converging to the local optima. Conversely, chaotic optimization emphasizes the global search, where its nonlinear jumps and exploratory paths may overlook local optima. When individually used, both of them face limitations in addressing specific hyperparameters and fail to fully leverage their strengths. However, by combining them, Bayesian optimization effectively guides the search process, while chaotic optimization explores broader potential regions, thus enhancing the model's exploration-convergence balance. This synergy prevents entrapment in the local optima, thereby improving the model's efficacy.

However, current mainstream neural networks typically solely focus on the performance of one-time training while neglecting the model's ability for long-term and repeated continual learning. When a trained model learns new knowledge, changes in parameters can lead to forgetting previously learned knowledge—a phenomenon known as catastrophic forgetting [49,50]. In future work, we will further explore the application of continual learning methods such as elastic weight consolidation (EWC) and learning without forgetting (LWF) in our experiments. Additionally, we plan to incorporate more deep learning techniques and advanced feature extraction methods to further enhance the model's prediction accuracy.

In this work, we extracted various acoustic features, including MFCC, from the collected audio dataset and merged them with other acoustic feature datasets. Subsequently, we compared the performance differences among four mainstream neural networks—GRU, RNN, MLP, and LSTM—in diagnosing PD using a patient's acoustic signals. This comparison was based on metric discrepancies on the test set and two visualization methods. We found that the LSTM neural network outperformed the others, thereby achieving an accuracy of 92.87%, a precision of 95.90%, a recall of 89.57%, and an F1-score of 92.57%. Next, we applied three hyperparameter optimization strategies—Bayesian optimization, chaotic optimization, and an integrated approach combining BOCO—to the LSTM neural network. The results showed that solely using Bayesian or chaotic optimization provided limited improvement. However, combining both of them significantly enhanced the LSTM's performance, thereby achieving an accuracy of 94.22%, a precision of 97.79%, a recall of 90.52%, and an F1-score of 93.97%. This performance boost was further validated through three visualization methods.

Overall, the innovation and practicality of this paper bring new perspectives to the field of medical diagnosis. By integrating voice signal analyses with deep learning and hyperparameter optimization techniques, we developed a more efficient diagnostic method for PD. This approach not only enhances the accuracy of early-stage PD diagnoses and reduces healthcare costs, but also holds significant potential for large-scale health screenings. We believe that, with continuous technological advancements and data accumulation, machine learning-based disease diagnostic methods which leverage acoustic signals will establish a prominent role in the future of intelligent healthcare.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by College Students' Innovative Entrepreneurial Training Plan Program under Grant No. 202410459130.

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. Brakedal B, Toker L, Haugarvoll K, Tzoulis C, (2022) A nationwide study of the incidence, prevalence and mortality of Parkinson's disease in the Norwegian population. *NPJ Parkinson's Dis* 8: 19. https://doi.org/10.1038/s41531-022-00280-4
2. Corti O, Lesage S, Brice A, (2011) What genetics tells us about the causes and mechanisms of Parkinson's disease. *Physiol Rev* 91: 1161–1218. https://doi.org/10.1152/physrev.00022.2010
3. Ryu DW, Han K, Cho AH, (2023) Mortality and causes of death in patients with Parkinson's disease: A nationwide population-based cohort study. *Front Neurol* 14: 1236296. https://doi.org/10.3389/fneur.2023.1236296
4. Su D, Cui Y, He C, Yin P, Bai R, Zhu J, et al. (2025) Projections for prevalence of Parkinson's disease and its driving factors in 195 countries and territories to 2050: Modelling study of Global Burden of Disease Study 2021. *BMJ* 2025: 388. https://doi.org/10.1136/bmj-2024-080952
5. Brooks DJ, (2012) Parkinson's disease: Diagnosis. *Parkinsonism Relat Disord* 18: S31–S33. https://doi.org/10.1016/S1353-8020(11)70012-8
6. Lippa CF, Duda JE, Grossman M, Hurtig HI, Aarsland D, Boeve BF, et al. (2007) DLB and PDD boundary issues: Diagnosis, treatment, molecular pathology, and biomarkers. *Neurology* 68: 812–819. https://doi.org/10.1212/01.wnl.0000256715.13907.d3
7. Banerjee J, Taroni JN, Allaway RJ, Prasad DV, Guinney J, Greene C, (2023) Machine learning in rare disease. *Nat Methods* 20: 803–814. https://doi.org/10.1038/s41592-023-01886-z
8. Arumugam K, Naved M, Shinde PP, Leiva-Chauca O, Huaman-Osorio A, Gonzales-Yanac T, (2023) Multiple disease prediction using Machine learning algorithms. *Mater Today Proc* 80: 3682–3685.https://doi.org/10.1016/j.matpr.2021.07.361
9. Bhatt CM, Patel P, Ghetia T, Mazzeo PL, (2023) Effective heart disease prediction using machine learning techniques. *Algorithms* 16: 88. https://doi.org/10.3390/a16020088
10. Kwekha-Rashid AS, Abduljabbar HN, Alhayani B, (2023) Coronavirus disease (COVID-19) cases analysis using machine-learning applications. *Appl Nanosci* 13: 2013–2025. https://doi.org/10.1007/s13204-021-01868-7
11. Islam MA, Majumder MZH, Hussein MA, (2023) Chronic kidney disease prediction based on machine learning algorithms. *J Pathol Inf* 14: 100189. https://doi.org/10.1016/j.jpi.2023.100189
12. Ogunpola A, Saeed F, Basurra S, Albarrak AM, Qasem SN, (2024) Machine learning-based predictive models for detection of cardiovascular diseases. *Diagnostics* 14: 144. https://doi.org/10.3390/diagnostics14020144
13. Govindu A, Palwe S, (2023) Early detection of Parkinson's disease using machine learning. *Procedia Comput Sci* 218: 249–261. https://doi.org/10.1016/j.procs.2023.01.007

14. Placido D, Yuan B, Hjaltelin JX, Zheng C, Haue AD, Chmura PJ, et al. (2023) A deep learning algorithm to predict risk of pancreatic cancer from disease trajectories. *Nat Med* 29: 1113–1122. https://doi.org/10.1038/s41591-023-02332-5
15. Inthiyaz S, Altahan BR, Ahammad SH, Rajesh V, Kalangi RR, Smirani LK, et al. (2023) Skin disease detection using deep learning. *Adv Eng Software* 175: 103361. https://doi.org/10.1016/j.advengsoft.2023.103361
16. Tanveer M, Goel T, Sharma R, Malik AK, Beheshti I, Del Ser J, et al. (2024) Ensemble deep learning for Alzheimer's disease characterization and estimation. *Nat Mental Health* 2: 655–667. https://doi.org/10.1038/s44220-024-00237-x
17. Groh M, Badri O, Daneshjou R, Koochek A, Harris C, Soenksen LR, et al. (2024) Deep learning-aided decision support for diagnosis of skin disease across skin tones. *Nat Med* 30: 573–583. https://doi.org/10.1038/s41591-023-02728-3
18. Chen J, Huang S, Zhang Y, Chang Q, Zhang Y, Li D, et al. (2024) Congenital heart disease detection by pediatric electrocardiogram based deep learning integrated with human concepts. *Nat Commun* 15: 976. https://doi.org/10.1038/s41467-024-44930-y
19. Ávila-Jiménez JL, Cantón-Habas V, del Pilar Carrera-González M, Rich-Ruiz M, Ventura S, (2024) A deep learning model for Alzheimer's disease diagnosis based on patient clinical records. *Comput Biol Med* 169: 107814. https://doi.org/10.1016/j.compbiomed.2024.107814
20. Mandava M, (2024) MDensNet201-IDRSRNet: Efficient cardiovascular disease prediction system using hybrid deep learning. *Biomed Signal Process Control* 93: 106147. https://doi.org/10.1016/j.bspc.2024.106147
21. Celik G, (2023) CovidCoughNet: A new method based on convolutional neural networks and deep feature extraction using pitch-shifting data augmentation for covid-19 detection from cough, breath, and voice signals. *Comput Biol Med* 163: 107153. https://doi.org/10.1016/j.compbiomed.2023.107153
22. Göreke V, (2023) A novel method based on Wiener filter for denoising Poisson noise from medical X-Ray images. *Biomed Signal Process Control* 79: 104031. https://doi.org/10.1016/j.bspc.2023.104031
23. Liu X, Fu S, Lin B, Nie X, (2023) Windowed variation kernel Wiener filter model for image denoising with edge preservation. *Opt Laser Technology* 167: 109688. https://doi.org/10.1016/j.optlastec.2023.109688
24. Gorev V, Gusev A, Korniienko V, Shedlovska Y, (2023) On the use of the Kolmogorov-Wiener filter for heavy-tail process prediction. *J Cyber Secur Mobility* 12: 315–338. https://doi.org/10.13052/jcsm2245-1439.123.4
25. Elkari B, Chaibi Y, Kousksou T, (2024) Random forest with feature selection and K-fold cross validation for predicting the electrical and thermal efficiencies of air based photovoltaic-thermal systems. *Energy Rep* 12: 988–999. https://doi.org/10.1016/j.egyr.2024.07.002
26. Gorriz JM, Segovia F, Ramirez J, Ortiz A, Suckling J, (2024) Is K-fold cross validation the best model selection method for machine learning?, preprint, arXiv:2401.16407. https://arxiv.org/abs/2401.16407
27. Tolstikhin IO, Houlsby N, Kolesnikov A, Beyer L, Zhai X, Unterthiner T, et al. (2021) MLP-mixer: An all-MLP architecture for vision, preprint, arXiv:2105.01601. https://doi.org/10.48550/arXiv.2105.01601
28. Zhu Q, Chen BY, Morgan N, Stolcke A, (2004) On using MLP features in LVCSR. *Interspeech* 2004: 921–924. https://doi.org/10.21437/Interspeech.2004-337

29. Safar AA, Salih DM, Murshid AM, (2023) Pattern recognition using the multi-layer perceptron (MLP) for medical disease: A survey. *Int J Nonlinear Anal Appl* 14: 1989–1998. https://doi.org/10.22075/ijnaa.2022.7114
30. Shiri FM, Perumal T, Mustapha N, Mohamed R, (2023) A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU, preprint, arXiv:2305.17473. https://arxiv.org/abs/2305.17473
31. Abumohsen M, Owda AY, Owda M, (2023) Electrical load forecasting using LSTM, GRU, and RNN algorithms. *Energies* 16: 2283. https://doi.org/10.3390/en16052283
32. Luo Y, Yu J, (2023) Music source separation with band-split RNN. *IEEE/ACM Trans Audio Speech Lang Process* 31: 1893–1901. https://doi.org/10.1109/TASLP.2023.3271145
33. Yadav H, Thakkar A, (2024) NOA-LSTM: An efficient LSTM cell architecture for time series forecasting. *Expert Syst Appl* 238: 122333. https://doi.org/10.1016/j.eswa.2024.122333
34. Alshingiti Z, Alaqel R, Al-Muhtadi J, Haq QEU, Saleem K, Faheem MH, (2023) A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN. *Electronics* 12: 232. https://doi.org/10.3390/electronics12010232
35. Pan S, Yang B, Wang S, Guo Z, Wang L, Liu J, et al. (2023) Oil well production prediction based on CNN-LSTM model with self-attention mechanism. *Energy* 284: 128701. https://doi.org/10.1016/j.energy.2023.128701
36. 46. Mim TR, Amatullah M, Afreen S, Yousuf MA, Uddin S, Alyami SA, et al. (2023) GRU-INC: An inception-attention based approach using GRU for human activity recognition. *Expert Syst Appl* 216: 119419. https://doi.org/10.1016/j.eswa.2023.119419
37. Mohsen S, (2023) Recognition of human activity using GRU deep learning algorithm. *Multimedia Tools Appl* 82: 47733–47749. https://doi.org/10.1007/s11042-023-15571-y
38. Tan KL, Lee CP, Lim KM, (2023) RoBERTa-GRU: A hybrid deep learning model for enhanced sentiment analysis. *Appl Sci* 13: 3915. https://doi.org/10.3390/app13063915
39. Aghaabbasi M, Ali M, Jasiński M, Leonowicz Z, Novák T, (2023) On hyperparameter optimization of machine learning methods using a Bayesian optimization algorithm to predict work travel mode choice. *IEEE Access* 11: 19762–19774. https://doi.org/10.1109/ACCESS.2023.3247448
40. Guo J, Ranković B, Schwaller P, (2023) Bayesian optimization for chemical reactions. *Chimia* 77: 31–38. https://doi.org/10.2533/chimia.2023.31
41. Luo T, Xie J, Zhang B, Zhang Y, Li C, Zhou J, (2024) An improved levy chaotic particle swarm optimization algorithm for energy-efficient cluster routing scheme in industrial wireless sensor networks. *Expert Syst Appl* 241: 122780. https://doi.org/10.1016/j.eswa.2024.122780
42. Peng T, Fu Y, Wang Y, Xiong J, Suo L, Nazir MS, et al. (2023) An intelligent hybrid approach for photovoltaic power forecasting using enhanced chaos game optimization algorithm and Locality sensitive hashing based Informer model. *J Build Eng* 78: 107635. https://doi.org/10.1016/j.jobe.2023.107635
43. Mohamed AA, Kamel S, Hassan MH, Zeinoddini-Meymand H, (2024) CAVOA: A chaotic optimization algorithm for optimal power flow with facts devices and stochastic wind power generation. *IET Gener Transm Distrib* 18: 121–144. https://doi.org/10.1049/gtd2.13076
44. Diallo R, Edalo C, Awe OO, (2024) Machine learning evaluation of imbalanced health data: a comparative analysis of balanced accuracy, MCC, and F1 score, In: *Practical Statistical Learning and Data Science Methods: Case Studies from LISA 2020 Global Network*, *USA*, Cham: Springer Nature Switzerland, 283–312. https://doi.org/10.1007/978-3-031-72215-8_12

45. Demir V, Citakoglu H, (2023) Forecasting of solar radiation using different machine learning approaches. *Neural Comput Appl* 35: 887–906. https://doi.org/10.1007/s00521-022-07841-x
46. Maniaci A, Riela PM, Iannella G, Lechien JR, La Mantia I, De Vincentiis M, et al. (2023) Machine learning identification of obstructive sleep apnea severity through the patient clinical features: A retrospective study. *Life* 13: 702. https://doi.org/10.3390/life13030702
47. Li J, (2024) Area under the ROC Curve has the most consistent evaluation for binary classification. *PLoS One* 19: e0316019. https://doi.org/10.1371/journal.pone.0316019
48. Hogan J, Adams NM, (2023) On averaging ROC curves. *Trans Mach Learn Res* 2023.
49. Dalal S, Seth B, Radulescu M, Cilan TF, Serbanescu L, (2023) Optimized deep learning with learning without forgetting (LwF) for weather classification for sustainable transportation and traffic safety. *Sustainability* 15: 6070. https://doi.org/10.3390/su15076070
50. Ranjith J, Baskaran S, Adithya B, (2024) Mitigating catastrophic forgetting in deep learning models for sentiment analysis, In: *2024 Second International Conference on Advances in Information Technology (ICAIT)* 1: 1–7. https://doi.org/10.1109/ICAIT61638.2024.10690454