International Journal of Speech Technology
https://doi.org/10.1007/s10772-024-10152-2

XEMLPD: an explainable ensemble machine learning approach for Parkinson disease diagnosis with optimized features

Fahmida Khanom[1] · Shuvo Biswas[2] · Mohammad Shorif Uddin[3] · Rafid Mostafiz[4]

Received: 30 May 2024 / Accepted: 16 September 2024

Abstract
Parkinson's disease (PD) is a progressive neurological disorder that gradually worsens over time, making early diagnosis difficult. Traditionally, diagnosis relies on a neurologist's detailed assessment of the patient's medical history and multiple scans. Recently, artificial intelligence (AI)-based computer-aided diagnosis (CAD) systems have demonstrated superior performance by capturing complex, nonlinear patterns in clinical data. However, the opaque nature of many AI models, often referred to as "black box" systems, has raised concerns about their transparency, resulting in hesitation among clinicians to trust their outputs. To address this challenge, we propose an explainable ensemble machine learning framework, XEMLPD, designed to provide both global and local interpretability in PD diagnosis while maintaining high predictive accuracy. Our study utilized two clinical datasets, carefully curated and optimized through a two-step data preprocessing technique that handled outliers and ensured data balance, thereby reducing bias. Several ensemble machine learning (EML) models—boosting, bagging, stacking, and voting—were evaluated, with optimized features selected using techniques such as SelectedKBest, mRMR, PCA, and LDA. Among these, the stacking model combined with LDA feature optimization consistently delivered the highest accuracy. To ensure transparency, we integrated explainable AI methods—SHapley Adaptive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME)—into the stacking model. These methods were applied post-evaluation, ensuring that each prediction is accompanied by a detailed explanation. By offering both global and local interpretability, the XEMLPD framework provides clear insights into the decision-making process of the model. This transparency aids clinicians in developing better treatment strategies and enhances the overall prognosis for PD patients. Additionally, our framework serves as a valuable tool for clinical data scientists in creating more reliable and interpretable CAD systems.

Keywords Parkinson's disease · Ensemble machine learning · Feature optimization · Explainable AI · CAD systems

✉ Rafid Mostafiz
rafid.iit@nstu.edu.bd

Fahmida Khanom
fahmida.khanom@aiub.edu

Shuvo Biswas
it21620@mbstu.ac.bd

Mohammad Shorif Uddin
shorifuddin@juniv.edu

[1] Department of Mathematics, American International University – Bangladesh, 408/1, Kuratoli, Khilkhet, Dhaka 1229, Bangladesh

[2] Department of Information and Communication Technology, Mawlana Bhashani Science and Technology University, Santosh, Bangladesh

[3] Department of Computer Science and Engineering, Jahangirnagar University, Savar, Bangladesh

[4] Institute of Information Technology, Noakhali Science and Technology University, Noakhali, Bangladesh

1 Introduction

The initial manifestations of Parkinson's disease (PD) are typically characterized by subtle and occasionally imperceptible symptoms. However, as the disease advances, these symptoms escalate in severity. Common symptoms include bradykinesia (slow movement), impaired coordination (difficulty with tasks such as walking and writing), tremors (involuntary shaking), muscle rigidity (stiffness that can cause discomfort and limit range of motion), alterations in speech (softening of the voice, slurring, or hesitancy), and postural instability (loss of balance and coordination). Additionally, reduced facial expressions, or a "masked face," are typical indicators of PD. These symptoms can be abbreviated as 'BITMAP' and serve as medical indicators of PD (Sveinbjornsdottir Oct., 2016). However, these measurements are not always satisfactory, especially in the early phases when

Published online: 16 October 2024

symptoms are not explicitly exposed. The obscureness of initial symptoms and lack of proper diagnosis contribute to the rapid increase in PD prevalence. Reports from the World Health Organization (WHO, 2022) indicate that the prevalence of PD has doubled over the last quarter century. In 2019, the global prevalence of PD was estimated at approximately 8.5 million individuals. That year, PD accounted for 5.5 million disability-adjusted life years (DALYs), an 81% increase since 2000. Furthermore, PD-related fatalities reached 329,000, a 100% increase since 2000 ("Parkinson disease". Accessed, 2024). Recent research by the Parkinson's Foundation noted that around 90,000 individuals are diagnosed with PD annually in the United States, marking a substantial 50% increase from the previously projected rate of 60,000 diagnoses per year ("International Congress of Parkinson's Disease & Movement Disorders®", 2024). The gender distribution among PD patients is approximately three men for every two women. While PD typically manifests around the age of sixty, it can occasionally emerge before the age of fifty (Georgiev et al., Dec. 2017).

Given the growing number of individuals impacted by PD, early identification is imperative. The limited manifestation of initial symptoms poses a significant challenge in early detection. To mitigate this, an automated machine learning (ML) approach is essential. In medicine, ML algorithms have demonstrated their capability to manage vast amounts of data effectively and provide insightful recommendations (Lundberg et al., 2018). Utilizing ML technology enhances patient safety (Kuo et al., 2019; Marella et al., 2017; Saria et al., 2010), boosts the quality of healthcare (Biswas et al., 2023; Liang et al., 2019; Nilashi et al., 2022), reduces healthcare expenses, and supports medical personnel in their work. However, effectively using ML technology requires significant dedication from skilled professionals. The "No Free Lunch" theorem emphasizes that no single algorithm excels in solving all possible problems. While healthcare researchers are well-versed in clinical data, they sometimes lack the skills to apply these approaches effectively to large datasets (Rikta et al., 2023). Developing and implementing ML solutions is a complex process that involves thorough data provisioning, identifying appropriate collaborators, and continuous iterative communication between ML and domain experts.

ML has emerged as a promising tool in the detection and diagnosis of PD, showcasing its potential impact on healthcare (Mohi Uddin et al., 2023). Leveraging advanced algorithms, PD models can analyze diverse data sources, including clinical records, imaging scans, and even voice recordings, to discern patterns indicative of PD. These models can identify subtle changes in motor function, tremors, and other symptoms that might be challenging for human observers to detect. Although ML models have shown great effectiveness in numerous classification tasks, they are still distrusted by general people or users without technical skills. This is due to their inadequate knowledge of the inner workings of AI algorithms. To establish confidence in ML algorithms, researchers have implemented explainable artificial intelligence (XAI) to explain and demonstrate how their algorithms generate outcomes (Došilović et al., 2018; Krajna et al., 2022). This study explained the outcomes of the highest-performing algorithm through two XAI methods: SHapely Adaptive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME). However, for the identification of PD, researchers have employed diverse techniques, including MRI images, EEG signals, writing images, Freezing of Gait (FoG), SPECT images, and EMG signals. Dysphonia, or abnormalities in speech, is an initial sign of PD that can be utilized to determine PD. Speech abnormalities have been selected in this research because they are straightforward, inexpensive, and non-invasive. Researchers have employed diverse ensemble machine learning (EML) approaches to develop decision-making systems (DMS), such as preprocessing, feature engineering (FE), classification, and testing stages. Using EML approaches, it is possible to assess the disease patterns in health records and make decisions much more quickly. Outlier detection, data balancing, and normalization preprocessing procedures improve the accuracy and reduce the computational complexity of the EML model. The key contributions of this research are as follows:

- We performed a two-step data preprocessing process. First, we removed unusual data points (outliers) using outlier detection methods such as z-score and Winsor. Second, we addressed the class imbalance problem using the SMOTE-Tomek technique to ensure a balanced dataset.
- Various feature selection methods, including SelectedKBest, minimum redundancy maximum relevance (mRMR), principal component analysis (PCA), and linear discriminant analysis (LDA), were employed to identify the most pertinent features. The curated subset of features was then utilized for both training and testing the four EML classifiers to determine the optimal combination of feature selection technique and classifier performance. The stacking EML model exhibited remarkable results using PCA and LDA feature optimization techniques, achieving an accuracy, precision, recall, and F1 score of 100%.
- SHAP and LIME XAI methods were used to provide explanations for the best-fit EML model, highlighting the important factors of the developed XEMLPD model.

The manuscript is organized as follows: Sect. 2 provides an overview of the related work. Section 3 demonstrates the

proposed methodology. Section 4 outlines the findings and their discussion. Section 5 finally provides a conclusion.

2 Related work

The utilization of machine learning in Parkinson's disease detection not only enhances the accuracy and efficiency of diagnosis but also enables early intervention and personalized treatment strategies. As research in this domain progresses, the integration of machine learning holds great promise for revolutionizing our understanding and management of Parkinson's disease, ultimately leading to improved patient outcomes. Many researchers have concentrated their efforts on developing efficacious ML techniques for PD diagnosis. Table 1 presents a systematic review in tabular style, outlining the key features, shortcomings, and approaches of the relevant research.

The above-mentioned papers have made a substantial contribution to the ability of medical professionals to identify PD in its early phases. PD identification at an early stage is of the utmost importance to avert grievous complications. However, by implementing feature optimization methods, we can improve the results and reduce the computational complexity of an algorithm. Furthermore, medical experts struggle to understand the results generated by these models. Therefore, this research implemented XAI methods to ensure expert comprehension of the model's outcomes.

3 Methodology

The objective of this experiment is to choose a condensed representation of the optimum features for ensemble classification approaches that effectively summarize the crucial information needed for explaining the outcome. The XEMLPD is specifically created to identify a unique and effective set of features that may effectively predict the presence of Parkinson's disease from a complex dataset. Figure 1 represents the proposed approach for predicting PD. It consists of multiple steps, including data preparation, feature optimization, optimum classifier selection, and final prediction. We employed two outlier detection methods (z-score and Winsor) to remove unusual data points. The SMOTE-Tomek data balancing technique is used to handle data irregularity issues which plays a crucial role in the performance of a classifier. The optimal features are captured using mRMR, SelectKBest, PCA, and LDA as those techniques depict maximum accuracy on different combinations. Then four ensemble machine learning models were performed (boosting, bagging, stacking, and voting) on the selected features to select the optimum prediction classifier. The classifier that provides optimum results was considered the final predictor in this manuscript. Finally, the XML approaches (LIME and SHAP) have been evaluated to establish a transparent, robust, and trustworthy model.

Algorithm 1 describes PD prediction's working procedure step by step

```
Input: PD Dataset
Output: PD or non-PD
Begin:
   data ← load dataset;
   lime ← load explainer(lime);
   shap ← load explainer(shap);
   Function ExplainableAI(model, X):
      ex_lime ← lime.explainer(model, X);
      ex_shap ← shap.explainer(model, X);
      shap_val ← explainerShap(ex_shap);

            if plot is equal "bar":
               shap.bar(shap_val);
            else if plot is equal "beeswarm":
               shap.beeswarm(shap_val);
            else if plot is equal "waterfall":
               shap.waterfall(shap_val);
            else
               lime.predict(ex_lime);
   End Function

   Function preprocessing_data ( ):
      if data.dtypes is equal "string" or "object":
         encoding the data;
               X ← data.drop["parkin"];
               Y ← data["parkin"];
      end if
         x_out, y_out = outlier(X, Y); // detecting_outlier
         x_bl, y_bl = SMOTE-Tomek(x_out, y_out); // balancing_data
         aug_x, aug_y ← augmentation(x_bl, y_bl);  // augmenting_data
         opt_x = optimizer(component).fit(aug_x);  // feature_optimizing
       X1, X2, Y1, Y2 ← data_split(opt_x, aug_y);
   End Function

   for i in range(len(models)):
      checking for "parkin";
      model ← train(X1, Y1);
      predict ← test(X2, Y2);
      computes confusion metrics;
      ExplainableAI(model, X);
   end for
End
```

3.1 Experimental data

In this experiment, a total of 195 instances with 24 attributes were gathered from the UCI ML repository (Little, 2008). This UCI ML dataset was created by Little (Little, 2008) of the University of Oxford, Irvine. This dataset is formed of a variety of biomedical voice recordings from people of different ages, including 147 with Parkinson's disease (PD) and 48 with non-PD. The key objective of this dataset is to distinguish non-PD cases from PD cases based on the "status" attribute, where "1" is assigned to the PD person and "0" is assigned to the non-PD person. All features of this dataset are listed in Table 2.

On the other hand, the second dataset was created by Sakar et al. (2018) from the Department of Neurology at the CerrahpaéŸa Faculty of Medicine, Istanbul University.

Table 1 Concurrent research on PD medical diagnosis using ML techniques on the UCI Parkinson's Speech Dataset

Paper	Methodology	Outcomes/strengths	Limitations
Lamba et al. (2022)	Combine three feature selection approaches (mutual information gain, additional tree, evolutionary algorithm) and three classifiers (NB, KNN, and RF)	Obtain approximately 95.58% accuracy	Limited set of vocal features
Senturk (2020)	Implements Recursive Feature Elimination and Feature Importance by utilizing classifiers such as SVM, ANN, and Regression Trees	Minimal effort while achieving high detection accuracy (93.84%)	Limited understand of the underlying features
Sharma et al. (2019)	Applies DT, KNN, and RF classifiers as the foundation for a bio-inspired algorithm Modified Grey Wolf Optimization (MGWO)	Estimated detection rate 98.28% with accuracy of 94.83%	Feature selection is not transparent
Polat (2019)	Adopted SMOTE (Synthetic Minority Over-Sampling Technique) with Random Forest ML-classifier	Success rate is 94.89% in classification to tackle real-world data with class-imbalanced problem in the medical field	Too many synthetic data rather than augmentation
Celik and Omurca (2019)	A comparison made between RF, Logistic Regression, Support Vector Machine, Extra Trees, and Gradient Boosting	Attempted a very clear and better assessment which results in (76.03%) accuracy level through Logistic Regression	Dataset are imbalanced and comparative feature selection is not efficient
Mamun et al. (2022)	Impose the NB, KNN, RF, AdaBoost, bagging, DT, LR, LightGBM, SVM, and XGBoost classifiers after using the SMOTE feature selection process	LightGBM excelled over other ML algorithms and was selected as the recommended model based on its accuracy and AUC score	Multimodal feature-based system needs to be introduced
Avuçlu and Elen (2020)	Conduct NB, KNN, RF, and SVM ML algorithms	SVM yielded the highest accuracy (88.72%) Improves statistical precise value by using biological voice parameters	Several Feature optimization techniques are needed
Solana-Lavalle et al. (2020)	Employ Wrappers feature subset selection with KNN, multilayer perceptron, SVM and RF	Reduce computational complexity	Needs explainable AI (XAI) methods to explain the output of the model
Alshammri et al. (2023)	Address SVM, RF, DT, KNN, and MLP models trained with SMOTE, Feature Selection, and hyperparameter tweaking (GridSearchCV) to improve performance	The MLP model showcased commendable accuracy of 98.31%, recall of 98%, precision of 100%, and an f1-score of 99%. SVM attained an accuracy of 95%, recall of 96%, and precision of 98%	Restricted feature selection and classifier approaches
Dhanalakshmi et al. (2024)	Prosecute RF, KNN, SVM, XGBoost, DT, and LR are being studied as binary classifiers and use of SMOTE-ENN avoids class imbalance by oversampling and under-sampling	SMOTE-ENN in conjunction with SVM results in an accuracy rate of 96.5%	Needs feature optimization and XAI methods
Shastry (2023)	Launches the Tree based Nearest Neighbour (TNN) technique	Showed significant performance enhancements compared to individual regression models	Lack of proper feature reduction approach
Mahesh et al. (2024)	Incorporating KNN, RF, SVM and XGBOOST along with their ensemble methods, having decreased entropy levels	With 98% accuracy and a Matthew's correlation coefficient of 0.93, the findings demonstrate how well the homogenous XGBoost-RF performs in comparison to other ML techniques	Lack of data balancing; Limited dataset;

Table 1 (continued)

Paper	Methodology	Outcomes/strengths	Limitations
Chaurasia and Chaurasia (2023)	Initial base classifiers include LR, KNN, NB, SVM, and DT. All classifiers are combined in the second stage, or stack model. Bagging, AdaBoost, RF, and GBC comprise the third-stage ensemble model	The third ensemble ML (EML), GBC, attained the best accuracy on the testing data, reaching 97.43%. The KNN base model and stacking meta-model both achieved the highest accuracy of 94.87%. The GBC is the sole EML in this study that has the highest accuracy	Architecture is very complex; Computational intensity
Oguri et al. (2023)	A collection of four Tree based algorithms, namely DT, RF, XGBoost, and LightGBM, were utilized	LightGBM achieved an impressive accuracy rate of 97.43%	Black box nature
Nissar et al. (2019)	Use LR, NB, KNN, RF, DT, SVM, MLP, and XGBoost classifiers with mRMR and RFE feature optimizes	XGBoost with mRMR feature optimizer exhibited higher accuracy of 95.39%	Only traditional ML classifiers were used
Nahar et al. (2021)	Four ML classifiers, namely gradient boosting, extreme gradient boosting, bagging and Extra Tree with Boruta, RFE and RF feature optimizers were used	Bagging with RFE was exhibited high accuracy of 82.35%	Feature selection is not transparent
Saleh et al. (2024)	Use an Artificial Neurons Network (ANN) and nineteen ML classifiers	Ensemble voting classifier with Hyperparameters Tuning and Cross-Validation gave best result	Limited dataset
Failed (2024)	Stacking Ensemble model consists of five base learners, namely XGB, SGD, ET, GB, and KNN with Bayesian optimizer	Demonstrated high accuracy by merging outputs from all base learners	Complex architecture and time consuming
Al-Tam et al. (2024)	Evaluated individual and EML models including RF, DT, LR, SVM, GB, Stacking and Bagging	Stacking-based GB + SVM + LR ensemble ML model demonstrated high accuracy with 96.05%	Needs feature optimizer
Bukhari and Ogudo (2024)	Impose the AdaBoost classifier after using the SMOTE data balancing and PCA feature extraction process	Model was fine-tuned to achieve better results and used grid search cross-validation to identify best hyperparameter values	Lack of feature explainability

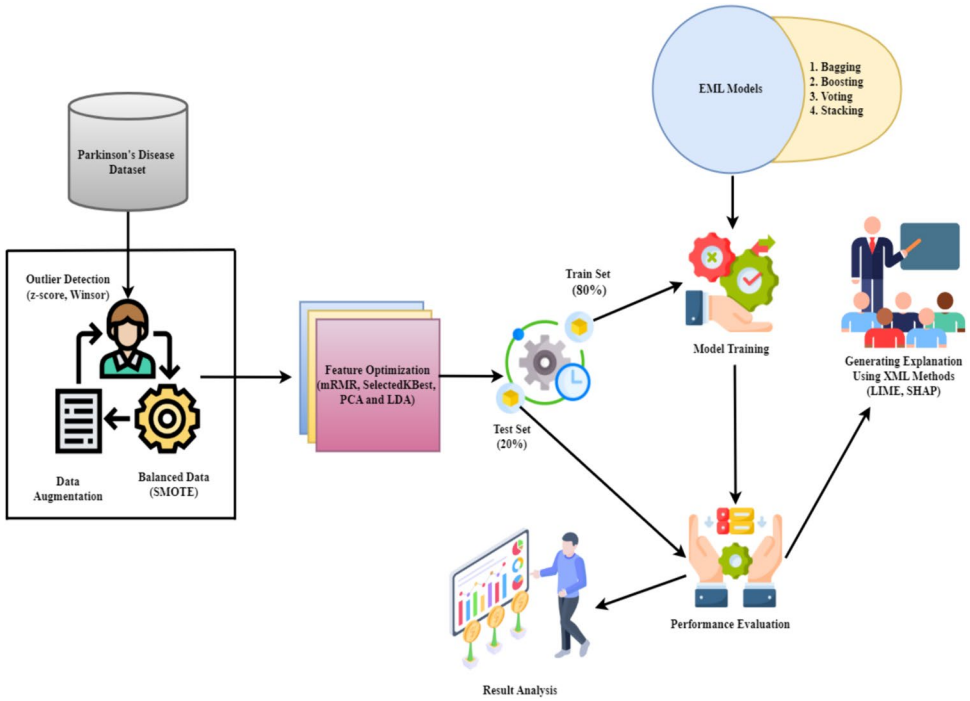


Fig. 1 The proposed approach for predicting PD

This dataset comprised of 188 PD patients, aged 33 to 87 (10 men and 81 women). Furthermore, 64 non-PD people were included in the dataset (23 men and 41 women), aged 41 to 82. This dataset contained 756 instances and 754 features. We randomly partition these two dataset into two sets: training (80%) and testing (20%) to conduct this experiment.

3.2 Data preprocessing

This section consists of three data preprocessing steps, including outlier detection, data balancing, and data augmentation. Each step is briefly described below:

3.2.1 Outlier detection

Detecting outliers is a crucial step in data preparation, as atypical data points can have a detrimental impact on the accuracy and reliability of the model (Boukerche et al., May 2021). The z-score and Winsor methods are commonly employed

Table 2 Description of data attributes

Parameters	Meaning
Name	ASCII subject name and recording number
MDVP: Fo(Hz)	Average vocal fundamental frequency
MDVP: Fhi(Hz)	Maximum vocal fundamental frequency
MDVP: Flo (Hz)	Minimum vocal fundamental frequency
MDVP: Jitter (%), MDVP: Jitter (Abs), MDVP: RAP, MDVP: PPQ, Jitter: DDP	Several measures of variation in fundamental frequency/ Multiple indicators of fundamental frequency fluctuation
MDVP: Shimmer, MDVP: Shimmer(dB), Shimmer: APQ3, Shimmer: APQ5, MDVP: APQ, Shimmer: DDA	Several measures of variation in amplitude/ Multiple amplitude variation measurements
NHR, HNR	Two measures of ratio of noise to tonal components in the voice
Status	Health status of the subject (one)—Parkinson's, (zero)—healthy
RPDE, D2	Two nonlinear dynamical complexity measures
DFA	Signal fractal scaling exponent
Spread1, spread2, PPE	Three nonlinear measures of fundamental frequency variation

for outlier detection in data points. The Z-score quantifies the number of standard deviations by which a data point deviates from the mean, facilitating the detection of outliers (Martinez-Millana et al., 2018). In the context of outlier detection, data points that have z-scores greater than or less than a threshold, often set at ± 3 standard deviations, are categorized as outliers. This approach relies on the traditional normal distribution, in which 99.7% of the data is contained within three standard deviations from the average. This methodology aids in the identification of anomalies and ensures the integrity of statistical analysis (Boukerche et al., May 2021). It is calculated by subtracting the mean of the dataset from the observed value and then dividing the result by the standard deviation of the dataset. Firstly, we calculated mean and standard deviation of the dataset. Then calculate the z-score and set threshold = 3 for outliers. Data points with Z-scores exceeding this threshold are considered outliers. The formula for calculating the Z-score for each data point x_i is shown in Eq. 1.

$$Z_i = \frac{(x_i - \mu)}{\sigma} \tag{1}$$

where Z_i is the Z-score for *i*th data point; x_i is the value of the data point; μ is the mean of the dataset which can be defined as formula $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$; σ is the standard deviation of the dataset which can be defined as formula $\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$. Figure 2 shows the original data with outliers (left side) and cleaned data without outliers (right side) which is extracted at the time of applying the Z-score detection method.

Winsorization, a technique known as Winsor (Hoo et al., 2002), is employed to identify outliers by replacing extremely high or low data values with less extreme values. Instead of directly removing outliers, Winsorization method replaces extreme values with the closest values falling inside a specific percentile range. Winsorization suggests that extreme numbers may arise due to measurement errors or random fluctuations, rather than being outliers. It substitutes high values with less extreme values in order to mitigate the influence of outliers on statistical analysis without eliminating the information they provide. However, in handling imbalanced data using the Winsorization method, the following limitations of lower cap value and upper cap value may occur: (1) influence on minority class instances; (2) reduce the variance of the dataset; and (3) bias towards the majority class. Firstly, this method selects a lower percentile and an upper percentile as thresholds. The upper and lower outliers are calculated based on the following rules: data points less than the lower percentile are considered lower outliers, and similarly, data points higher than the upper percentile are considered upper outliers. Finally, data points below the lower percentile were replaced at the lower percentile, and similarly, points above the upper percentile were replaced at the upper percentile. This is formulated by Eq. 2.

$$Winsorized\ Data\ [i] = \begin{cases} Lower\ Cap\ Value & if\ Data[i] < Lower\ Cap\ Value \\ Upper\ Cap\ Value & if\ Data[i] > Upper\ Cap\ Value \\ Data[i] & Otherwise \end{cases} \tag{2}$$

Here, decide on the percentile values P_{low} (lower percentile) and P_{high} (upper percentile) to cap the extreme values. Thus, Lower Cap Value = $^{\text{Percentile}}\ P_{low}$ and Upper Cap Value = $^{\text{Percentile}}\ P_{high}$.

Figure 3 shows the comparison between the original and Winsorized data for the two columns (MDVP:Fo(Hz) and

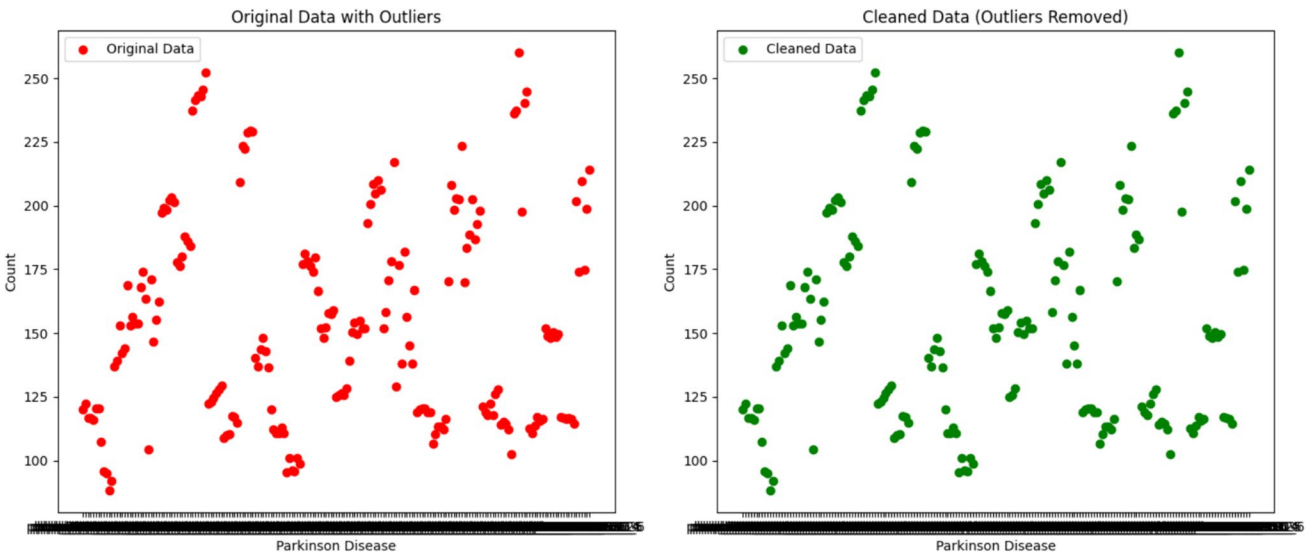


Fig. 2 Original data with outliers (left side) and cleaned data without outliers (right side) using Z-score method

MDVP:Shimmer). The histograms show how Winsorization has reduced the impact of outliers by limiting extreme values, leading to a more "compressed" distribution in the Winsorized data (shown in red).

3.2.2 Data balancing

The dataset shows that the non-PD cases are approximately one-third of the PD cases. This distribution means that PD cases represent approximately 75.4% of the sample, while non-PD cases make up just about 24.6%. To calculate the total percentage of PD and non-PD cases in the dataset, we can use the formula:

$$\begin{aligned} &Total\ Percentage\ of\ PD\ Cases \\ &= (Number\ of\ PD\ Cases / Total\ Number\ of\ Cases) \\ &\times 100 \end{aligned}$$

$$\begin{aligned} &Total\ Percentage\ of\ Non-PD\ Cases \\ &= (Number\ of\ Non-PD\ Cases / Total\ Number\ of\ Cases) \\ &\times 100 \end{aligned}$$

This influence in the sample distribution is referred as an imbalanced dataset, which may bias the evaluation outcomes. This experiment has incorporated a tricky data balancing approach namely SMOTE-Tomek, on the feature vector to address this issue. It combines the SMOTE oversampling method with the Tomek links under-sampling method to improve the balance between classes in a dataset. SMOTE works by generating synthetic samples in the feature space. The basic idea behind SMOTE is to create synthetic samples in the feature space. It randomly selects a minority class instance and computes the k-nearest neighbors for this instance (Fernández et al., 2018). The synthetic instance is then generated by selecting one of the k-nearest neighbors and creating a random linear combination of the features of the selected neighbor and the original instance.

$$\begin{aligned} New\ Instance\ &= Original\ Instance\ +\ Random\ Value \\ &\times (Neighbor\ Instance\ -\ Original\ Instance) \end{aligned}$$

Tomek links are pairs of instances of opposite classes that are closest to each other. These pairs can be identified using the k-nearest neighbors' algorithm. If a Tomek link is found between two instances, the majority class instance of the pair is removed.

Tomek Link = (x_i, x_j); where x_i is from the minority class and x_j is from the majority class, and they are each other's nearest neighbors.

Let x_i be a minority class instance, and x_{nn} be one of its k-nearest neighbors from the same class. Therefore, Synthetic Instance = $x_i + \lambda \times (x_{nn} - x_i)$; Where λ is a random number between 0 and 1. Now, Remove x_j (majority class instance) from each Tomek link (x_i, x_j).

In Fig. 4, the blue color indicates the minority class (non-PD cases), and the green color indicates the majority class (PD cases). After applying the SMOTE-Tomek balancing technique, the minority class increased from 48 to 147 cases, while the majority class remained the same. Figure 4 shows the imbalanced data before and after data balancing.

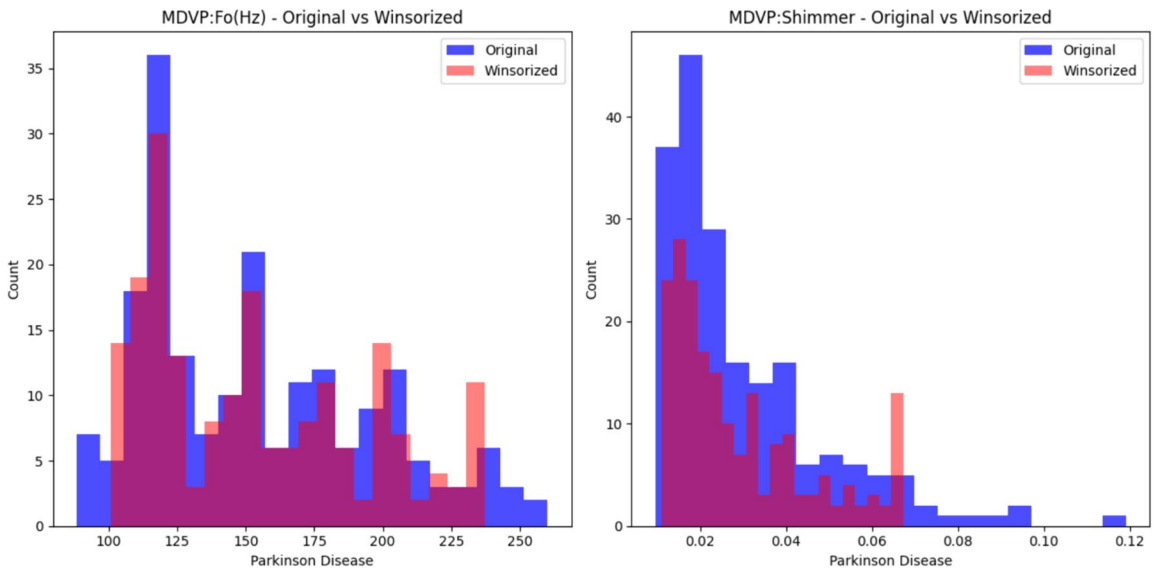


Fig. 3 Histogram analysis using Winsorization method: Before outlier detection (left side) and after outlier detection (right side)

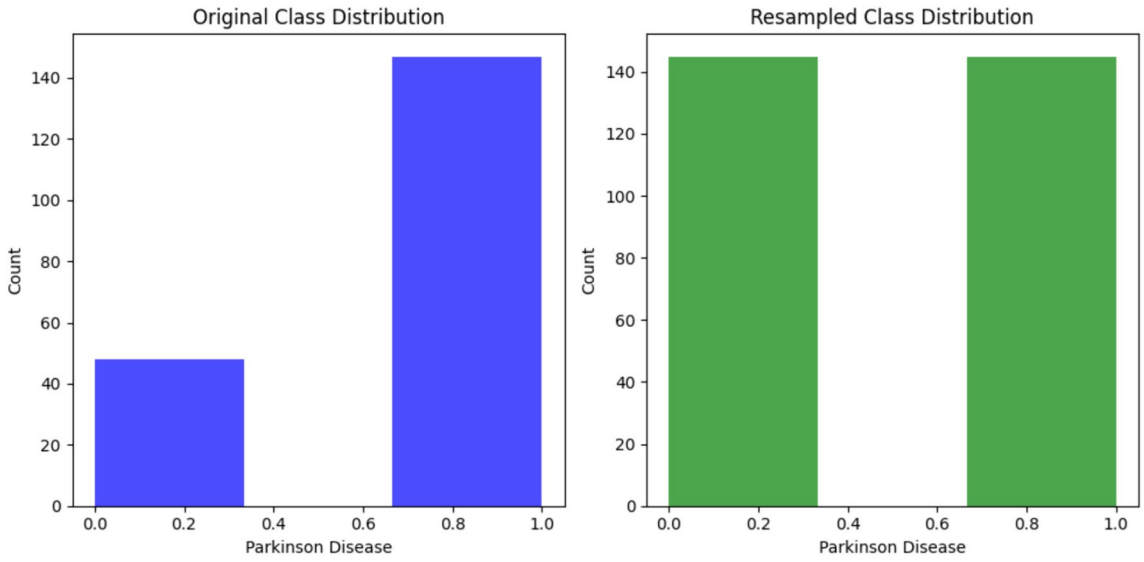


Fig. 4 Data balanced: Before data balancing (left side) and after data balancing (right side)

3.2.3 Data augmentation

Data augmentation is an essential procedure that improves the dataset by implementing several ways to increase its size and diversity (Park, et al., 2019). Here, various augmentation techniques have been employed, including scaling, rotation, and shifting, to generate extra instances for training. This strategy is particularly advantageous in circumstances that require more generalization, as it maximizes the use of the existing datasets. By integrating altered iterations of the original data, it improves the robustness of the model. Table 3 shows the number of examples after data balancing and augmentation.

3.3 Feature optimization techniques

Feature optimization involves reducing the number of features to simplify computations and enhance the performance of the ensemble model. Various methods for optimizing features are available; the most common approaches such as SelectKBest for feature importance, mRMR for feature selection, then PCA and LDA for feature reduction are utilized in this research. For each feature optimization approach, we have chosen the best fitted techniques based on our experiment. Thus, to analyze the strength of feature optimization in Parkinson diagnosis, this research utilizes four feature optimization techniques. The next subsections provide a detailed description of how each strategy for optimizing PD features is used in this research.

3.3.1 Principal component analysis (PCA)

Principal Component Analysis (PCA) is an algorithm in unsupervised learning that is utilized to investigate the relationships among a set of variables (Song et al., 2010). The proposed XEMLPD model applies PCA to reduce the size of a PD dataset by keeping important patterns and correlations between variables, even if the target variables are unknown at the beginning of execution. Additionally, PCA concentrates on primary components with substantial variance, enabling noise reduction and effectively capturing the underlying data structure (Aich et al., 2019). Let's assume we have a dataset X consisting of n samples and m features. Before applying PCA, it's common to standardize the features to have a zero mean and unit variance as Eq. 3.

Table 3 Number of examples after data balancing and augmentation

Types	Original data		After balancing		After Augmentation	
	Number	Percentages (%)	Number	Percentages (%)	Number	Percentages (%)
PD	147	75.4	147	50	447	50
Non-PD	48	24.6	147	50	447	50
Total	195	100	294	100	894	100

$$X_{std} = \frac{(X - \mu)}{\sigma} \tag{3}$$

where μ is the mean vector and σ is the standard deviation vector. Now, compute the covariance matrix of the standardized data to provide a measure of how two variables change together. If two features of a dataset are X_1 and X_2; we apply Eq. 4 to reduce the dimensionality of this dataset using PCA to compute the covariance matrix of the standardized data.

$$Cov(X_1, X_2) = \frac{1}{n-1} \sum_{k=1}^{n} (X_{1k} - \overline{X}_1)(X_{2k} - \overline{X}_2) \tag{4}$$

Here, $\overline{X}_1$ and $\overline{X}_2$ are the means of the standardized features X_1 and X_2, respectively in total n number of samples. We find the eigenvalues $\lambda 1$ and $\lambda 2$; then corresponding eigenvectors of the covariance matrix are eigvec1 and eigvec2. The eigenvectors represent the directions of maximum variance (principal components), and the eigenvalues represent the magnitude of variance along these directions. The eigenvalues are sorted in descending order and choose the top k eigenvectors corresponding to the k largest eigenvalues to form a $k \times d$ matrix W, where d is the original number of features and k is the desired number of principal components. Finally, Eq. 5 projects the original data onto the new feature subspace using the matrix W; where, y is the transposed value and x is the original feature vector.

$$y = W^T x \tag{5}$$

This transformed data will reduce the dimensionality of while preserving most of the original information that leads to faster computation times reduced storage requirements, and improved model performance.

3.3.2 Linear discriminant analysis (LDA)

Linear Discriminant Analysis (LDA) is a dimensionality reduction technique that is commonly used for feature optimization and classification tasks. LDA focuses on finding the linear combinations of features that best separate different classes in a dataset. It learns a linear combination of features to successfully discriminate between classes by projecting data into a lower-dimensional space (Mostafiz et al., 2018). LDA maximizes the ratio of between-class variation to within-class variance, which computes the best paths in the feature space for class distinction (Amin et al., 2017). It optimizes class reparability, combines class information for supervised learning, and improves classification performance, particularly in circumstances with well-separated classes (Mostafiz et al., 2017). Let's consider a dataset X consisting of n samples and m features, partitioned into k classes. The mean vector μ_i of each class is calculated by Eq. 6, where, Nc is the number of samples in class k, y_i is the class label of sample i, and x_i is the feature vector of sample i. Equations 7 and 8 compute the matrices of the between-class scatter S_b and the within-class scatter S_w respectively.

$$\mu_i = \frac{1}{N_i} \sum_{i:y_i=k} x_i \tag{6}$$

$$S_b = \sum_{i=1}^{k} N_i (\mu_i - \mu)(\mu_i - \mu)^T \tag{7}$$

$$S_w = \sum_{i=1}^{k} \sum_{i:y_i=k} (x_i - \mu_k)(x_i - \mu_k)^T \tag{8}$$

Then, compute the eigenvectors and eigenvalues of the matrix $S_{w-1}\, S_b$ in Eq. 9.

$$S_{w-1} S_b v_i = \lambda_i v_i \tag{9}$$

where v_i is the i-th eigenvector and λ_i is the corresponding eigenvalue.

To form the projection matrix W, select the top k eigenvectors that correspond to the largest eigenvalues. Equation 10, indicates the projection of the dataset x onto the subspace that the selected discriminant directions span; where y is the projected value in the new space, W^T is the transpose of the weight vector, and x is the original feature vector.

$$\mathrm{y} = W^T x \tag{10}$$

This projection focuses on finding the linear combinations of features that maximizes class separability for the optimal projection of the data; leading to improved classification performance and dimensionality reduction.

3.3.3 Minimum redundancy maximum relevance (mRMR)

Minimum redundancy maximum relevance (mRMR) (Radovic et al., 2017) aims to select a subset of features that maximizes the relevance to the target variable while minimizing redundancy among the selected features. For the selected features it optimizes the mutual dependencies and thus reduces the dimensionality of the feature space. The Relevance $R(x)$; measures how much information a feature provides about the target variable. It can be quantified using mutual information $I(x, y)$ between the feature x and the target y. Equation 11 computes the mutual dependencies of x and y. Redundancy measures the amount of information

shared between two features x and z. Equation 12 denotes redundancy $S(x, z)$ for two features x and z.

$$R(x) = I(x, y) \tag{11}$$

$$S(x, z) = I(x, z) \tag{12}$$

The mRMR criterion combines relevance and redundancy to evaluate the importance of a feature. The mRMR criterion for a feature x is given by Eq. 13.

$$mRMR(x) = R(x) - \frac{1}{|S|} \sum_{z \in S} S(x, z) \tag{13}$$

where S is the set of selected features, and $|S|$ is the number of selected features.

The selected features with high relevance and low redundancy are a subset of informative features and reduce the dimensionality of the data, lead to improved model generalization and interpretability.

3.3.4 SelectKBest

The statistical approaches for feature selection involve evaluating the association between each feature and the targeted feature and choosing the input features that have the greatest correlation with the target feature. The study utilized the SelectKBest algorithm with the chi-square approach to extract the most optimal features from the dataset. The SelectKBest function uses this technique as a scoring function to ascertain the correlation between each characteristic and the target feature. The chi-square test is used to calculate the score between each feature and the target feature. If the resulting value is lower, it signifies that the feature is unrelated to the target feature. Conversely, a higher resulting value indicates that the feature is not randomly associated with the target feature.

The chi-square test measures the association between two categorical variables. It calculates the difference between the observed frequencies O_i and expected frequencies E_i of each category combination, under the assumption of independence between the variables. The chi-square test statistic is $\chi 2$ for a contingency. It can be calculated as Eq. 14.

$$\chi 2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{14}$$

For each categorical feature Xi, we compute the chi-square statistic with the target categorical feature y to get the score by Eq. 15.

$$score\big(X_i, y\big) = \chi 2\big(X_i, y\big) \tag{15}$$

After computing the chi-square scores for all features, we rank the features based on their scores in descending order. Finally, we select the top k features with the highest chi-square scores as the best features in Eq. 16.

$$X_s = TopK(score\big(X_i, y\big)) \tag{16}$$

where X_s represents the selected features, and $TopK$ selects the top k features.

This mathematical model outlines the process of using SelectKBest with the chi-square method for feature selection to the target feature, ranking the features based on their scores, and selecting the top k features for further analysis or modeling.

3.4 Ensemble machine learning (EML) algorithms

In this research, an ensemble classifier evaluates each optimized feature set, derived from several feature optimization techniques, individually. Ensemble machine learning (EML) involves the building and combination of several models to improve performance over individual models (Polikar & Ma, 2012). By harnessing the diversity among the basis classifiers, EML can capture multiple features of the data and limit the danger of overfitting (Bind et al., 2015). In this work, four EML classifiers, namely bagging, boosting, stacking, and voting, are evaluated to find the best-suited classifier for accurately predicting PD. The working procedures of these EML classifiers are briefly described in this segment.

3.4.1 Bagging

Bagging (Bootstrap Aggregating) (Breiman Aug., 1996) stands as an ensemble learning strategy designed to enhance the robustness and precision of machine learning models by amalgamating the forecasts of numerous base models. Initially, it generates multiple bootstrap samples by randomly picking data points from the original dataset with replacements. This method employs parallel base estimator generation, such as decision trees (DT) (Yadav & Pal Dec., 2020), to foster diversity within ensemble methodologies. For the original PD dataset D with N samples, base estimators generate B numbers of bootstrap samples denoted asD_i, where$i = 1, 2, \ldots, B$. For each bootstrap sampleD_i, the base model is trained independently on this sample. After the training phase concludes for the base models, each model produces predictions for the original dataset. Subsequently, the final prediction is determined through majority voting, which entails selecting the most prevalent prediction among all the base models. Equation 17 computes the PD classification task and the final prediction is denoted as$\hat{y}$. The prediction results $h_i(x)$ are derived from each base model M_i for input$x$.

$$\hat{y} = argmax_y \sum_{i=1}^{B} 1(h_i(x) = y) \tag{17}$$

Here, $1(\cdot)$ is the indicator function.

Therefore, Bagging serves to diminish the model's variance through the process of averaging or voting across numerous base models (Schapire Jun., 1990). Moreover, it addresses the issue of overfitting by training each base model on distinct subsets of the data, thereby enhancing generalization.

3.4.2 Boosting

Boosting (Freund et al., 1999) is another popular ensemble learning technique that aims to improve the performance of machine learning models by sequentially training multiple weak learners and combining their predictions. The sequential base models are trained iteratively rather than the base models individually so that at every iteration the models learn to correct the errors made by the previous model. Serving as an ensemble meta-algorithm, Boosting adeptly mitigates both bias and variance, thereby enhancing overall predictive performance. Initially, each sample in the training dataset is assigned an equal weight w_i for N number of samples defined as Eq. 18.

$$w_i = \frac{1}{N} \tag{18}$$

This study employs AdaBoost (Reddy et al., 2021) as the boosting classifier that sequentially trained multiple DT (Yadav & Pal Dec., 2020) with limited depth (colloquially termed weak learners) on iteratively modified datasets using the current sample weights. In each iteration, AdaBoost strategically assigns greater emphasis to misclassified instances, compelling subsequent decision trees to minimize the weighted error rate. The weighted error rate is calculated as the sum of the weights of misclassified samples divided by the total weight of all samples using Eq. 19.

$$\in = \frac{\sum_{i=1}^{N} w_i.1(y_i \neq h(x_i))}{\sum_{i=1}^{N} w_i} \tag{19}$$

where $h(x_i)$ is the prediction of the base model for the sample x_i, y_i is the true label of the sample x_i, and $1(\cdot)$ is the indicator function. Equation 20 indicates the weight of the current base model α, computed based on its performance. Finally, sample weights are updated to give more weight to the misclassified samples by Eq. 21, which helps the next base model focus more on these samples. The updated sample weights are normalized by Eq. 22.

$$\alpha = 1/2\ln\left(\frac{1-\epsilon}{\epsilon}\right) \tag{20}$$

$$w_i = w_i.\exp(-\alpha.y_i.h(x_i)) \tag{21}$$

$$w_i = \frac{w_i}{\sum_{j=1}^{N} w_j} \tag{22}$$

Here, exp denotes the exponential function, y_i is the true label of the sample x_i; $h(x_i)$ is the prediction of the base model for the sample x_i. The ultimate prediction denoted by Eq. 23, is obtained from the prediction results of the total T number of base models, where α_t is the weight and $h_t(x)$ is prediction respectively from the t-th base model for the input x_i.

$$\hat{y} = \text{sign}(\sum_{t=1}^{T} \alpha_t.h_t(x)) \tag{23}$$

3.4.3 Stacking

Stacking, or stacked generalization, is an ensemble learning technique that combines the predictions of multiple base classifiers using a meta-learner, generally a logistic regression (LR) model (Wolpert, 1992). The primary idea behind stacking involves training the (stage-1) base learners simultaneously on the same dataset. Finally, their predicted scores are adjusted with the true scores to create an intermediate training set, and then the (stage-2) meta-learner is trained on this intermediate dataset (Liang et al., 2021). PD dataset is used in training for the N number of base models $M_1, M_2, \ldots, M_N$ that produces the prediction results y_i; where i represents the index of the base model. For the input data point x stacking ensemble combines these predictions and obtains a final prediction result denoted by Eq. 24. This combination is typically done using a meta-learner or a second-level model.

$$\hat{y} = \sum_{i=1}^{N} y_i(x) \tag{24}$$

These predictions are used as features (meta-features) for training the meta-learner. The predictions of the meta-learner for data point x can be represented using the vector $\hat{Y}(x)$ by Eq. 25. This study uses a logistic regression (Dreiseitl & Ohno-Machado, 2002) model as the meta-learner $f(\cdot)$ on the input vector $\hat{Y}(x)$ to obtain the final prediction $\hat{y}$ which is denoted by Eq.26.

$\hat{Y}(x) =$

$$\hat{Y}(x) = [\hat{y}_1(x),\ \hat{y}_1(x),\ \ \ldots,\ \hat{y}_N(x) \tag{25}$$

$$\hat{y} = f(\hat{Y}(x)) \tag{26}$$

This approach aims to combine the strengths of different base models and improve predictive performance. It is trained on a separate validation set or through cross-validation to prevent overfitting.

3.4.4 Voting

The voting ensemble learning approach aggregates predictions from numerous independent base estimators to make a final prediction (Jani et al., 2022). It uses the "wisdom of the crowd" notion to create more accurate predictions by taking into account the aggregate judgment of numerous models rather than depending on a single model. The final prediction is made by summing the expected probabilities across all models and selecting the label with the highest average probability (Mostafiz et al., 2021). The main concept is to select the average prediction based on a set of selected classifiers. This research uses three influential ML models, namely LR (Dreiseitl & Ohno-Machado, 2002), Support Vector Machine (SVM) (Jakkula, 2006), and DT (Yadav & Pal Dec., 2020), to construct the classification model for PD prediction. This prediction is obtained for N base models denoted as $H_1, H_2, \ldots, H_N$; where each base model H_i predicts a probability distribution over the possible class labels y for a given input x, presented by Eq. 27.

$$H_i = p_i(y|x) \tag{27}$$

The final prediction $\hat{y}$ is determined by averaging these predicted probabilities across all base models in Eq. 28.

$$\hat{y} = argmax_y \frac{1}{N} \sum_{i=1}^{N} p_i(y|x) \tag{28}$$

This is a nuanced approach to decision-making that considers the confidence levels of predictions rather than just the class labels.

3.5 Proposed algorithm

This research averages the projected probabilities from each classifier, selecting the class with the highest average probability as the final prediction. An advanced ensemble strategy is used to train a meta-learner on the predictions of the base classifiers to make the final decision, learning to integrate the predictions optimally. Figure 5 illustrates the comprehensive architecture of the proposed EML model employed in this paper. Algorithm 2 describes the step-by-step EML model's working procedure. The integration of multiple models into the proposed model can increase computational requirements and make the system harder to manage. To mitigate this issue, we applied several feature optimizers like mRMR, PCA, and LDA that reduce model complexity by decreasing the dimensionality of the feature space, thereby simplifying the model. mRMR selects the most relevant and least redundant features, PCA transforms the original features into a smaller set of uncorrelated components that capture most of the variance, and LDA finds linear combinations that best separate classes. By reducing the number of features, these methods lead to faster computation, a lower risk of overfitting, and more interpretable models, ultimately making the model more efficient and easier to handle.

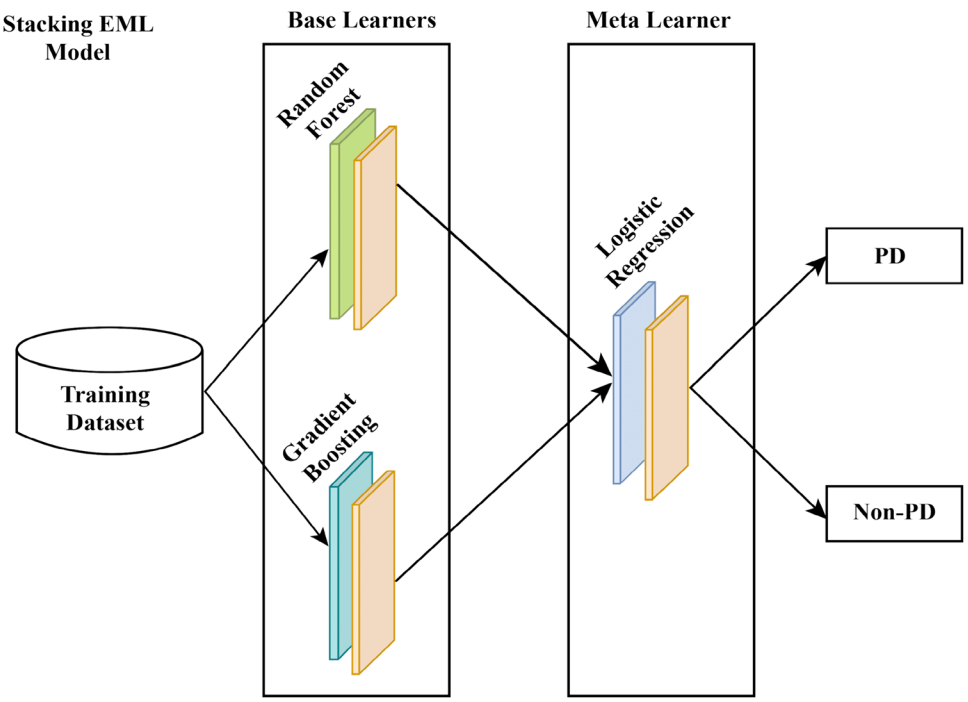


Fig. 5 The overall architecture of the stacking EML model

Algorithm 2 Stacking

Input: *Training data,* $S_{x,y} = (x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$
Output: *A stacked EML model* M_{stack}
The stage-1 base learning procedures T;
Procedure:
Step 1: *Train stage-1 base learners* M_{base}*;*
for t = 1, . . ., T:
Fit a base-learner M_{base} *using* $S_{x,y}$*;*
end for
Step 2: *Form a new dataset* $\check{S}$ *from* $S_{x,y}$*;*
for t = 1, . . ., T:
Form a new dataset $\check{S}$ *containing* $\{\check{x}_i, y_i\}$*, here* $= \{m_1(x_1), m_2(x_2), \ldots, m_T(x_i)\}$*;*
end for
Step 3: *Train the meta-learner* $\check{m}$ *utilizing the new dataset;*
return $M(x) = \check{m}(m_1(x_1), m_2(x_2), \ldots, m_T(x_i))$*;*

3.6 Explainable machine learning (XML) algorithms

To foster trust in ML approaches, it is essential to visually illustrate and clarify how ML models arrive at decisions (Lundberg & Lee, 2017). The use of explainable AI (XAI) is critical in ensuring that people can easily grasp both the algorithm's decision-making steps and the data used in its training process (Biswas et al., Mar. 2024; Biswas et al., May 2024). The XEMLPD framework incorporates two XAI algorithms, namely SHAP and LIME to enhance the transparency and interpretability of ML models' decision-making processes.

3.6.1 Shapley additive exPlanations (SHAP)

SHAP (SHapley Additive exPlanations) is a method for explaining individual predictions of machine learning models based on cooperative game theory (Alotaibi et al., 2023). In cooperative game theory, the Shapley value is a method to fairly distribute the "payout" among players based on their contribution to the coalition. In the context of SHAP, each feature in a prediction is considered a player, and the payout is the difference between the prediction for a specific instance and the average prediction for all instances (Samudrala et al., 2024). SHAP considers all possible subsets of features (coalitions) and calculates the contribution of each feature to the prediction by measuring the change in prediction when that feature is included in the coalition.

The Shapley value $\varnothing_i$ for feature i is defined as the average marginal contribution of feature i across all possible coalitions in Eq. 29. Here, $v(S)$ represents the value of coalition S, which is the difference between the model's prediction with the features in S and the model's average prediction. The power set 2^N of all features N includes all possible coalitions. For each feature, SHAP calculates its Shapley value by considering all possible coalitions that include that feature and averaging the marginal contributions across all permutations of the coalition. Equation 30 indicates the prediction for an instance x by the sum of the base value $\varnothing_0$ and the contributions of individual features.

$$\varnothing_i = \sum_{S \subseteq N - \{i\}} \frac{|S|!.(|N| - |S| - 1)!}{|N|!} (v(S \cup \{i\}) - v(S)) \tag{29}$$

$$\hat{f}(x) = \varnothing_0 + \sum_{i=1}^{N} \varnothing_i \tag{30}$$

The magnitude of SHAP values quantifies the influence of each feature on the PD prediction. Positive SHAP values indicate that the presence of a feature increases the PD prediction relative to the average prediction, while negative SHAP values indicate the opposite influence.

3.6.2 Local interpretable model-agnostic explanations (LIME)

Local Interpretable Model-agnostic Explanations (LIME) priorities describing the model's prediction for specific instances rather than offering a comprehensive comprehension of the model across the full dataset (Polat, 2019).

Implementing the LIME approach provides valuable insights into how various features affect PD, differentiating between those that positively contribute and those that exert a negative influence. LIME focuses on explaining the predictions of a model in the vicinity of a particular instance of interest x and defines a neighborhood around x represented by a set of perturbed instances, denoted as x'; which are similar to x but have some small variations. Each perturbed sample x' is assigned a weight (proximity measure) (x') quantifies how close x' is to x in the feature space based on Euclidean distance or cosine similarity. A subset X'_S of perturbed samples are selected to assign weights (x') to each selected sample based on their proximity to the original data point x by Eq. 31. Fit an interpretable model g to approximate the behavior of the complex model f around the data point x in Eq. 32; where L is a loss function measuring the difference between the predictions of the complex model f and the local interpretable model g, and $\Omega(g)$ is a regularization term on the complexity of the local model g.

$$w(x\prime) = \frac{\pi(x\prime)}{\sum_{x\prime_i \in X'_S} \pi(x\prime_i)} \tag{31}$$

$$g(x) = \arg\min_{g} \sum_{x'_i \in X'_S} w\left(x'_i\right).L(f(x), f\left(x'_i\right) + \Omega(g) \tag{32}$$

The feature importance is obtained from the attributes of the local model g. This will generate an explanation by highlighting the most influential features and their contributions to the prediction. This can be written as Explanation(x) = Feature Importance(g).

Table 4 Experimental resources of the XEMLPD

Resources	Details
GPU	Tesla K80
GPU Memory	16 GB
CPU	Intel Core i5-12600 K @ 3700 MHz
RAM	64 GB
Cache	128 MB
Disk Space	500 GB
Session	12 h

4 Result and discussion

This section presents a comparative analysis of feature selection methods using different ensemble classifiers based on feature explainability. The experimental data points are acquired through two preprocessing steps, involving outlier handling and data balancing with augmentations. This process will yield a total of 894 instances in the working dataset, randomly divided into 715 (approximately 80%) for training and 179 (approximately 20%) for testing purposes. Four feature optimization techniques (SelectKBest, mRMR, PCA, LDA) are applied in this research, based on three common approaches: feature importance, feature selection, and feature reduction. The whole process is evaluated by four ensemble techniques bagging, boosting, stacking, and voting for each of the individual feature optimizers. After that, analyze their explainability to select the best model for PD diagnosis. Each obtained results are validated using tenfold cross-validation. To obtain the results, various environments were employed in this study. Table 4 outlines the environment setup necessary for the experiment. All techniques were executed on a local machine using Jupyter Notebook in Python. Jupyter Notebook (Failed, 2016) is a collaborative development platform for ML model experimentation that permits Python programming language. This platform integrates Python libraries or modules such as Scikit Learn, NumPy, Pandas, Seaborn, and Matplotlib. The Scikit-Learn library was developed using top Python libraries like NumPy, SciPy, and Matplotlib. The scikit-Learn library is easily adapted into existing Python workflows. Panda is a powerful Python package optimized for data manipulation and analysis. It simplifies data processing tasks like data cleaning, aggregation, and filtering. It also permits importing data from diverse formats, including Excel, CSV, etc. NumPy package is used for calculating large operations like multi-dimensional matrices and arrays. Seaborn is a data visualization library built based on Matplotlib. Seaborn package is used for drawing diverse plots, including bar plots, scatter plots, box plots, violin plots, etc. Matplotlib is a data analysis and visualization tool for generating diverse plots, including line graphs, bar charts, scatter plots, histograms, etc.

The statistical parameters namely- Accuracy (*ACC*), specificity (*SPE*), sensitivity (*SEN*), precision (*PRE*),

Table 5 Performance measure of EML models for all features

Methods	ACC (%)	SPE (%)	SEN (%)	PRE (%)	F1 score (%)	Kappa (%)
Bagging	81.01	76.92	86.67	73.03	79.27	61.98
Boosting	84.36	79.61	89.61	76.67	82.64	67.78
Stacking	92.74	91.40	94.19	91.01	92.57	85.47
Voting	77.65	97.40	78.82	97.10	87.01	75.07

Table 6 Optimized Features

Optimization methods	No. of selected features	Features
SelectKBest	12	PPE, D2, spread2, spread1, DFA, RPDE, HNR, NHR, Shimmer:DDA, MDVP:APQ, Shimmer:APQ5, Shimmer:APQ3
mRMR	12	MDVP:Fhi(Hz), MDVP:Flo(Hz), status, HNR, D2, spread1, spread2, PPE, MDVP:Shimmer(dB), MDVP:Jitter(Abs), RPDE, MDVP:Jitter(%)
PCA	13	MDVP:Shimmer(dB), MDVP:Shimmer, MDVP:PPQ, MDVP:Jitter(%), MDVP:APQ, Shimmer:DDA, Shimmer:APQ3, Shimmer:APQ5, D2, PPE, MDVP:RAP, spread1, MDVP:Jitter(Abs)
LDA	9	Shimmer:APQ3, MDVP:RAP, MDVP:PPQ, MDVP:Shimmer, PPE, D2, spread2, DFA, MDVP:Shimmer(dB)

*F*1*score*, *Kappa* are used to measure the performance of the working models. Equations (32)–(37) represent the calculating process of these evaluation metrics. In the evaluation metrics, each model's accuracy (*ACC*) indicates the overall accurate prediction rate, specificity (*SPE*) determines the correct non-PD prediction cases whereas sensitivity (*SEN*) is the correct PD prediction cases, precision (*PRE*) means the correct PD prediction rate from all predicted PD cases. The *F*1*score* is employed to characterize the robustness of the model. The *Kappa* (kappa statistics) is used to realize the efficiency of the ensemble model. Measurements of these parameters close to 1 indicate the better classifier. Model evaluation is a critical component of developing a strong EML model. The Models' caliber is determined by a confusion matrix where the correctly classified PD instances are True Positive (*TP*) and True Negative (*TN*) and the incorrectly classified PD instances are False Positive (*FP*) and False Negative (*FN*). These evaluation metrics help to judge how perfectly well the proposed system processed the input raw data. Table 5 shows the values of performance results considering all the features on each ensemble method before feature optimization (Table 6).

Accuracy (ACC): The fraction of correctly predicted instances to the total number of predicted instances is called accuracy. The mathematical definition of accuracy is defined in Eq. 33.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{33}$$

Specificity (SPE): The fraction of TN to the overall number of actual negative instances. The mathematical definition of specificity is defined in Eq. 34.

$$SPE = \frac{TN}{TN + FP} \times 100\% \tag{34}$$

Sensitivity (SEN)/Recall: The ratio of actual positives to positive label cases is called recall. The mathematical definition of sensitivity is defined in Eq. 35.

$$SEN = \frac{TP}{TP + FN} \times 100\% \tag{35}$$

Precision (PRE): The fraction of actual positives to the overall number of positive classifications. The mathematical definition of precision is defined in Eq. 36.

$$PRE = \frac{TP}{TP + FP} \times 100\% \tag{36}$$

F1 score: The F1 score utilizes harmonic mean to compute SEN plus PRE at almost the equal rate. The mathematical definition of F1 score is defined in Eq. 37.

$$F1score = 2 \times \frac{SE \times PR}{SE + PR} \times 100\% \tag{37}$$

Kappa: It measures the agreement between true and predicted predictions. The mathematical definition of Kappa is defined in Eq. 38.

$$Kappa = \frac{TotalACC - RandomACC}{1 - RandomACC} \times 100\% \tag{38}$$

Table 7 Performance measure of EML models for SelectKBest feature set

Methods	ACC (%)	SPE (%)	SEN (%)	PRE (%)	F1 score (%)	Kappa (%)
Bagging	96.65	48.63	98.82	94.38	96.55	93.29
Boosting	97.21	98.85	95.65	98.88	97.24	94.41
Stacking	98.32	97.78	97.75	97.75	97.75	95.53
Voting	90.50	87.63	93.90	86.52	90.06	91.70

Table 8 Performance measure of EML models for mRMR feature set

Methods	ACC (%)	SPE (%)	SEN (%)	PRE (%)	F1 score (%)	Kappa (%)
Bagging	89.94	87.50	92.77	86.52	89.54	79.88
Boosting	87.71	83.33	93.51	80.90	86.75	75.40
Stacking	94.41	90.82	98.77	89.89	94.12	88.82
Voting	86.59	83.00	91.14	80.90	85.72	73.17

Table 9 Performance measure of EML models for PCA feature set

Methods	ACC (%)	SPE (%)	SEN (%)	PRE (%)	F1 score (%)	Kappa (%)
Bagging	92.74	95.29	90.43	95.51	92.90	85.47
Boosting	87.71	90.48	85.26	91.01	88.04	75.42
Stacking	98.32	97.80	98.86	97.75	98.30	96.65
Voting	89.39	83.33	93.51	80.90	86.75	75.40

Table 10 Performance measure of EML and traditional ML models for LDA feature set

Approaches	Methods	ACC (%)	SPE (%)	SEN (%)	PRE (%)	F1 score (%)	Kappa (%)
Traditional	SVM	88.83	88.15	89.16	87.06	88.10	77.57
	KNN	99.44	100	98.80	100	99.39	98.88
	NB	81.56	94.79	66.27	91.67	76.92	62.22
	LR	89.39	89.58	89.16	88.10	8862	78.68
Ensemble	Bagging	98.88	98.89	98.88	98.88	98.88	97.76
	Boosting	99.44	100	98.88	100	99.43	98.88
	Stacking	100	100	100	100	100	100
	Voting	96.64	97.73	95.60	97.75	96.66	93.30

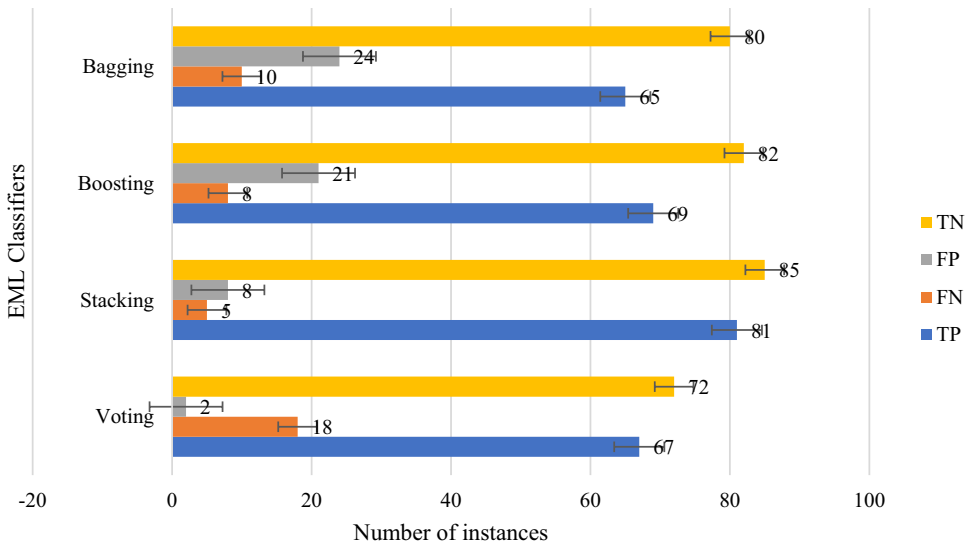


Fig. 6 Results from the Confusion matrix of the EML classifiers for all feature set

In Table 5, the kappa value is less than others because, in this case, we evaluated these EML models on all features. This value increased when we evaluated these EML models using several feature optimization techniques like LDA, PCA, mRMR, and SelectKBest. These optimizers produce an optimized feature set by selecting the best features from all features. Thus, the kappa value increased after evaluating these EML models on this optimized feature set (see Tables 7, 8, 9, 10). Figure 6 illustrates the data obtained from the Confusion Metrix (CM) in all feature schemes.

The feature importance score is calculated and selected features are tabulated in Table 6. The features' importance's are obtained using SelectKBest, then find the 12 important feature set. 12 Relevant Features of no redundancy are selected by mRMR. The PCA and LDA outperform other optimization techniques and obtain 13 and 9 features, respectively.

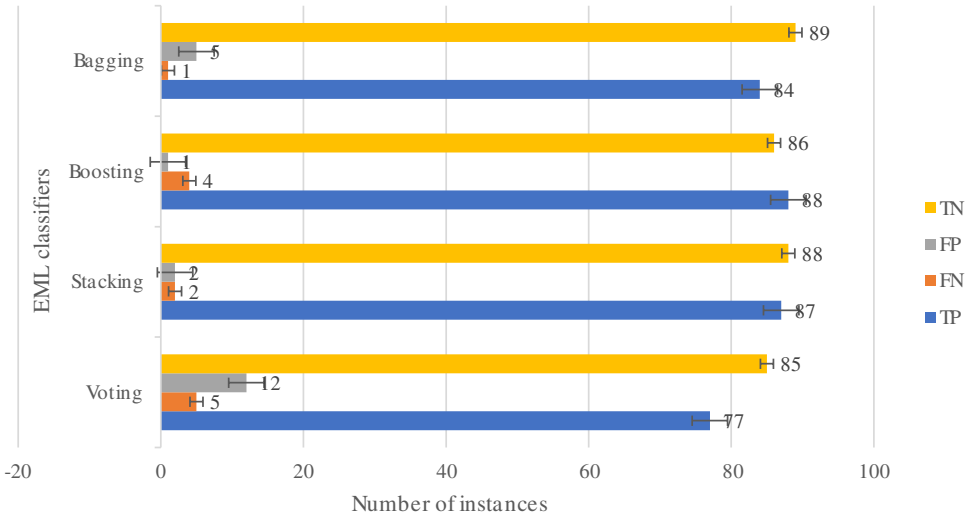


Fig. 7 Results from the Confusion matrix of the EML classifiers for SelectKBest feature set

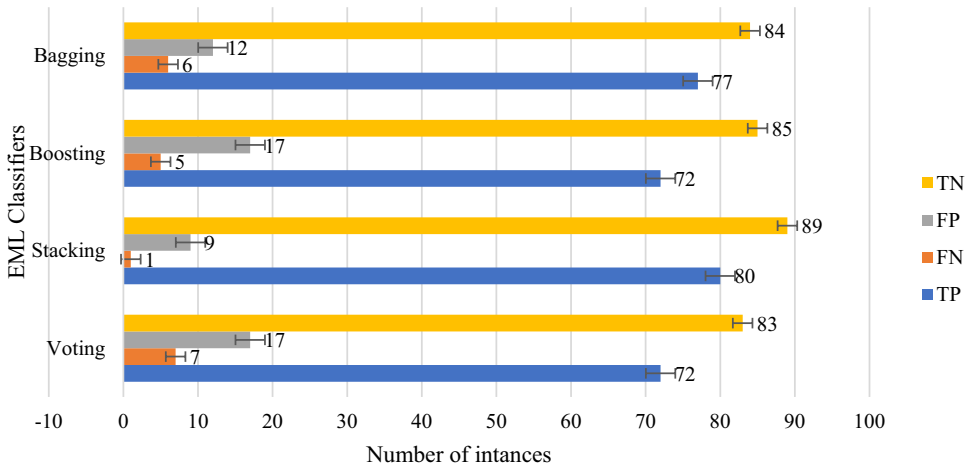


Fig. 8 Results from the Confusion matrix of the EML classifiers for the mRMR feature set

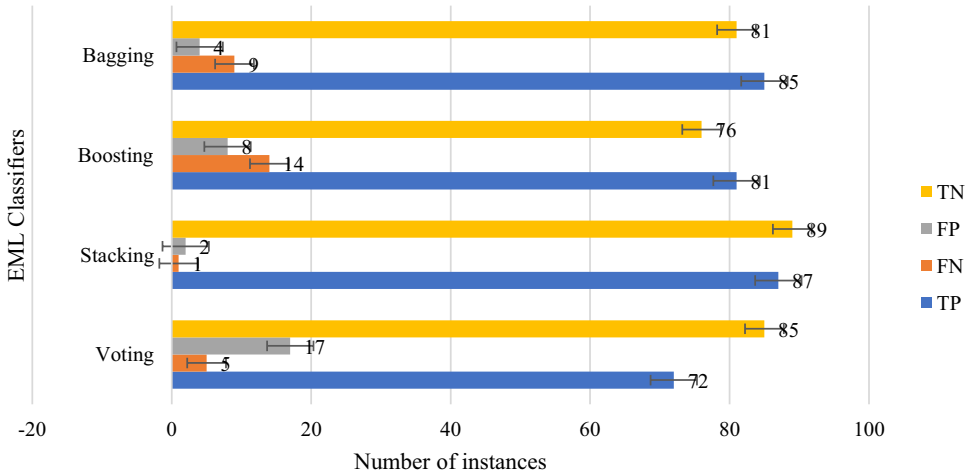


Fig. 9 Results from the Confusion matrix of the EML classifiers for the PCA feature set

The performance of EML classifiers was assessed on various feature subsets after the feature optimization using different feature selection methods. The performance metrics of EML classifiers on the optimized feature set using SelectKBest are presented in Table 7. The stacking EML model achieves an accuracy of 98.32% with a Kappa score of 95.53%, indicating the effectiveness of this model. Figure 7 shows the normalized form of CM obtained from SelectKBest feature set on each EML models.

The performance measurement of EML models on the feature set selected by the mRMR is depicted in Table 8. The efficiency of the mRMR-selected feature set appears lower efficiency compared to all previous feature sets. Figure 8 summarizes the outcome of CM for all the EML models using the mRMR feature set.

The performance evaluation of EML models operating on the reduced feature subsets selected by PCA selection methods is presented in Table 9. The stacking EML model demonstrates high accuracy, sensitivity, specificity, and F1-score, while the boosting EML model slightly outperforms in terms of precision. However, Kappa indicates that the most efficient outcome is achieved by Stacking on the PCA feature set. Figure 9 shows the overall results of CM for all the EML models using PCA feature set.

Table 10 shows the value of performance measurement metrics of various EML models in this research working on the reduced feature subsets of LDA optimization. The stacking EML model demonstrates high accuracy, sensitivity, specificity, and F1-score, while the boosting EML model also performs equally high in terms of specification and precision. However, the Kappa indicator assists in finding the most efficient model. Stacking EML with an LDA feature set becomes the most efficient model. Figure 10 shows the comparison of the CM results obtained from each EML model using the LDA feature set. We evaluated several traditional ML approaches to the LDA feature set for a more comprehensive comparative analysis. Table 10 shows the comparative results of traditional ML approaches with ensemble ML approaches.

This article used two performance metrics, including the Confusion Metrix (CM) and AUC-ROC curve to examine the calibers of the proposed system. The ROC (Receiver Operating Characteristics)-AUC (area under the curve) curve is a plot of FPR (false positive rate) vs. TPR (true positive rate). The ROC-AUC value ensures the discrimination ability between PD and non-PD labels for the related EML model. It assesses an EML algorithm's overall prediction score. If an algorithm's predictions are 100% incorrect, then the AUC score is 0.00, or conversely, the AUC score is 1.00. Figure 11 represents the AUC-ROC curve for different feature selection methods in each EML model. The outcomes revealed that the stacking predictor achieved a remarkable AUC-ROC of 100% in the prediction of PD.

We have augmented and balanced the dataset using data augmentation and data balancing techniques. That's why our model has not seen overfitting on unseen data. Table 11 shows the comparative results of different EML models for the LDA feature set using two different datasets. We have re-evaluated our proposed model with a new dataset (Sakar et al. dataset) using the LDA feature optimizer. In this new dataset, the LDA feature optimizer selects 322 best features from 754 features. We have optimized these features continuously, but this LDA optimizer selected 322 features each time.

In this work, we re-evaluated feature optimization techniques like boosting, voting, and bagging using a nature-inspired-based optimization technique named Ant Colony Optimization (ACO). ACO explores potential solutions and refines them based on simulated trials but does not directly reduce the dimension of the feature set. On the other hand,

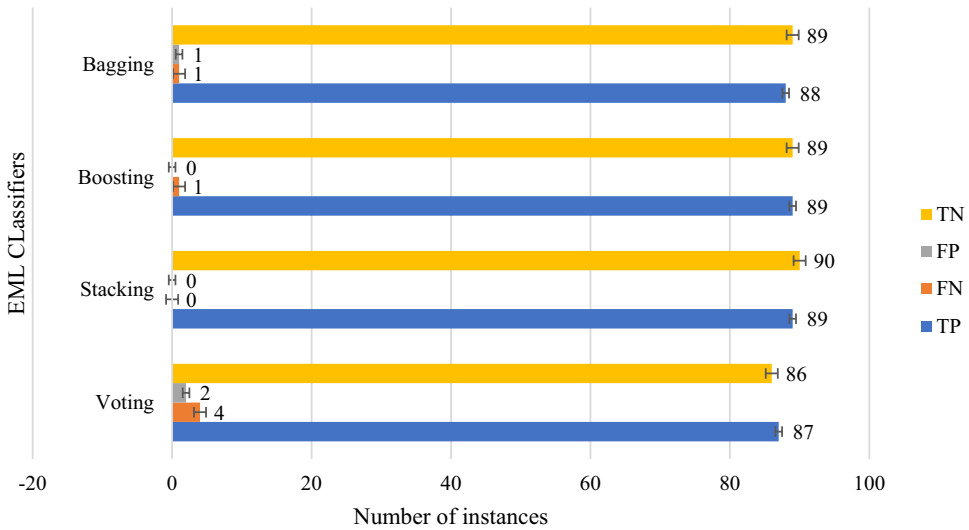


Fig. 10 Results from the Confusion matrix of the EML classifiers for the LDA feature set

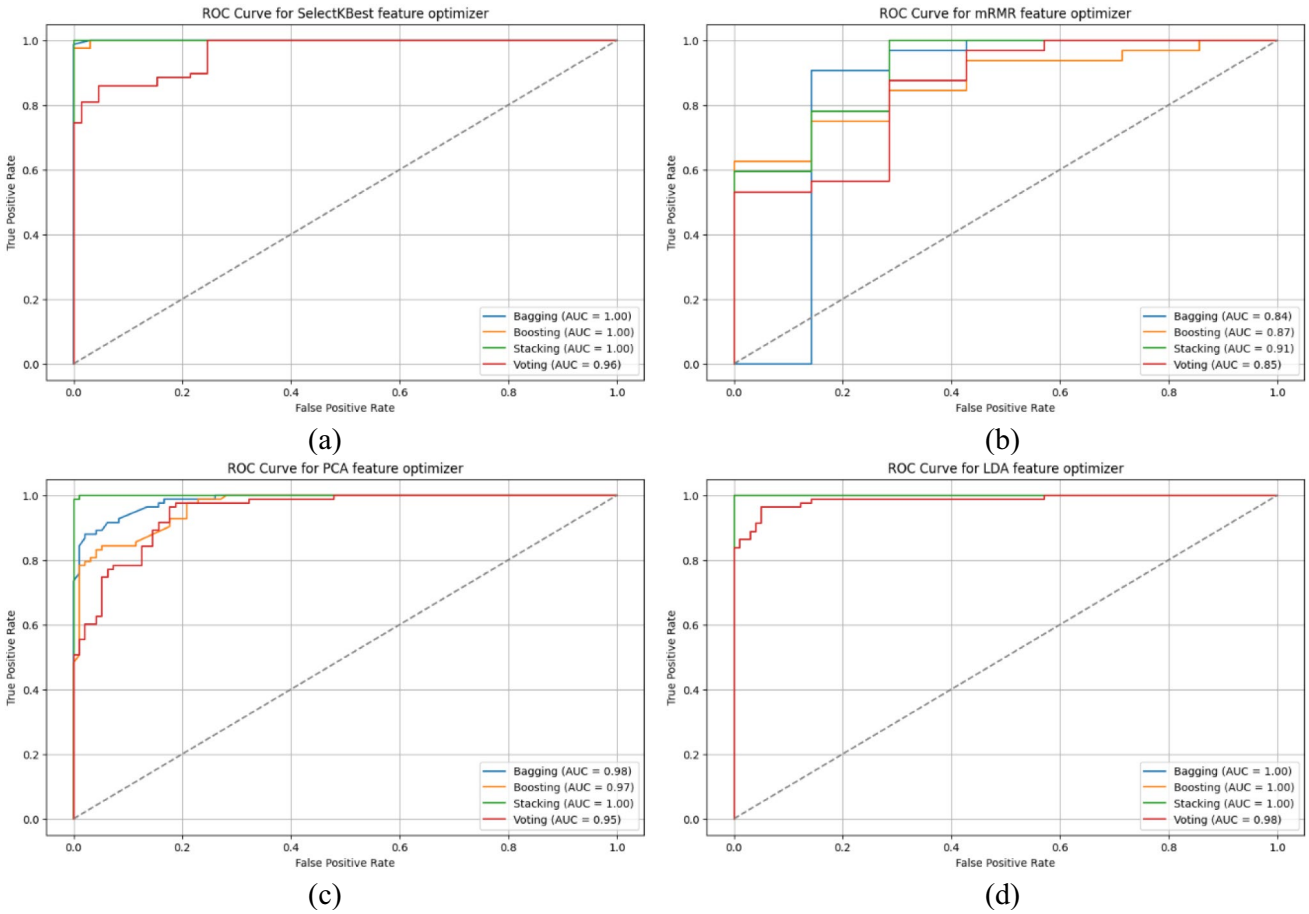


Fig. 11 ROC curve of the EML models: **a** using SelectKBest feature sets, **b** using mRMR feature set, **c** using PCA feature set, **d** using LDA feature set

Table 11 Performance analysis of EML models for LDA feature set using two different datasets

Models	Max Little Dataset Total instances = 197, Total features = 22 Selected features using LDA = 9				Sakar et al. Dataset Total instances = 756, Total features = 754 Selected features using LDA = 322			
	ACC (%)	PRE (%)	REC (%)	F1 (%)	ACC (%)	PRE (%)	REC (%)	F1 (%)
Bagging	98.88	98.88	98.88	98.88	96.27	96.97	95.52	96.24
Boosting	99.44	100	98.88	99.43	97.01	98.46	95.52	96.97
Stacking	100	100	100	100	96.64	96.99	96.27	96.63
Voting	96.64	97.75	95.60	96.66	95.52	96.21	94.78	95.49

Table 12 Performance analysis of different EML models for ACO nature inspired based optimizer

Models	LDA optimizer				ACO optimizer			
	ACC (%)	PRE (%)	REC (%)	F1 (%)	ACC (%)	PRE (%)	REC (%)	F1 (%)
Bagging	98.88	98.88	98.88	98.88	98.88	98.78	98.78	98.78
Boosting	99.44	100	98.88	99.43	96.09	93.10	98.78	95.86
Stacking	100	100	100	100	100	100	100	100
Voting	96.64	97.75	95.60	96.66	100	100	100	100

Table 13 Performance comparisons of the proposed method with prior works

Ref./year	Dataset	Labels	Model	Results
Lamba et al. (2022/2022)	Max Little	Healthy/PD	SMOTE + Genetic Algorithm + RF	Acc = 95.58%
Senturk (2020)/2024)	Max Little	Healthy/PD	LightGBM	Acc = 95%, Pre = 93.3%, Rec = 100%, F1 = 90%
Polat (2019)/2019)	Sakar *at el*	Healthy/PD	SMOTE + RF	Acc = 94.89%, Pre = 95.1%, Rec = 94.9%, F1 = 94.9%
Alshammri et al. (2023)/2023)	Max Little	Healthy/PD	MLP + SMOTE + GridSearchCV	Acc = 98.31%, Pre = 100%, Rec = 98%, F1 = 99%
Mahesh et al. (2024/2024)	Max Little	Healthy/PD	XGBoost + RF + SMOTE	Acc = 98%, Pre = 97.24%, Rec = 97.56%, F1 = 97.40%
Nissar et al. (2019)/2019)	Sakar *at el*	Healthy/PD	XGBoost + mRMR	Acc = 95.39%, Pre = 95%, Rec = 95%, F1 = 95%
Nahar (2021/2021)	Private	Healthy/PD	Bagging + RFE	Acc = 82.35%, Pre = 80%, Rec = 83%, F1 = 82%
Al-Tam et al. (2024)/2024)	Sakar *at el.* + Max Little	Healthy/PD	Stacking	Acc = 96.05%
Bukhari and Ogudo (2024)/2024)	Sakar at el	Healthy/PD	SMOTE + PCA + AdaBoost	Acc = 96%, Pre = 98%, Rec = 93%, F1 = 95%
Das Mar. (2010/2010)	Max Little	Healthy/PD	NN	Acc = 92.9%
Chen et al. (2016)/2016)	Max Little	Healthy/PD	mRMR + KELM	Acc = 96.47%
Yasar et al. (2019)/2019)	Max Little	Healthy/PD	ANN	Acc = 94.93%
Rasheed et al. (2020)/2020	Max Little	Healthy/PD	BPVAM + PCA	Acc = 97.5%
Rehman et al. (2023)/2023)	Max Little	Healthy/PD	Hybrid LSTM + GRU	Acc = 98%
Proposed1	Max Little	Healthy/PD	SMOTE-Tomek + LDA + Stacking	Acc = 100%, Pre = 100%, Rec = 100%, F1 = 100%
Proposed2	Sakar *at el*	Healthy/PD	SMOTE-Tomek + LDA + Boosting	Acc = 97.01%, Pre = 98.46%, Rec = 95.52%, F1 = 96.97%

Here *Acc* stands for accuracy, *Pre* stands for precision, *Rec* stands for recall and *F1 stands* for F1-score

the LDA feature optimizer focuses on maximizing the separation between classes in the dataset. It reduces overfitting and computational complexity by preserving the most discriminative features, thus giving higher predictive accuracy than the ACO. From Table 12, we see that the performance of the bagging and boosting models has been reduced compared to the proposed stacking model in the ACO optimizer. However, the voting and stacking classifiers achieved higher accuracy of 100% in ACO optimizer. Recently, many authors have developed automatic PD prediction approaches utilizing several EML algorithms. However, their approach found a few shortcomings, such as inconsistently optimized feature vectors and data imbalances. In this article, we implemented some techniques to overcome these shortcomings and also tried to compare it with other prior articles based on performance metrics. Table 13 reveals such comparison outcomes. It also ensures that the predictor is effective by a significant margin. Figure 12 represents various performance indicator curves like AUC, AUPR, and calibration of the proposed EML model using LDA feature set.

In this work, combining multiple ML models and performing extensive feature optimization reduces the overfitting issue by boosting the generalization capacity of ML approaches. Ensemble approaches such as bagging, boosting, and stacking used in this work combine predictions from various base learners, thereby minimizing variance and improving robustness against outliers and noise. This combination helps to reduce the risk of overfitting that a single classifier might not mitigate due to its specific biases and variances. Moreover, extensive feature optimization approaches, like feature selection and dimensionality reduction (e.g., through optimizers such as mRMR, PCA, or LDA) focus the model on the most important characteristics and eliminate redundancy. These optimizers simplify the ensemble model as well as reduce the chance of noise in the training data. The above advantages make the proposed model a more generalized and balanced system that performs better on unseen data.

4.1 Time complexity analysis

Let f be the number of original features, k be the number of optimized features, and m be the total number of samples in the dataset. We estimate the time complexity in this work using two criteria: the optimization of new features and the architecture design. In optimizing new features, we compute

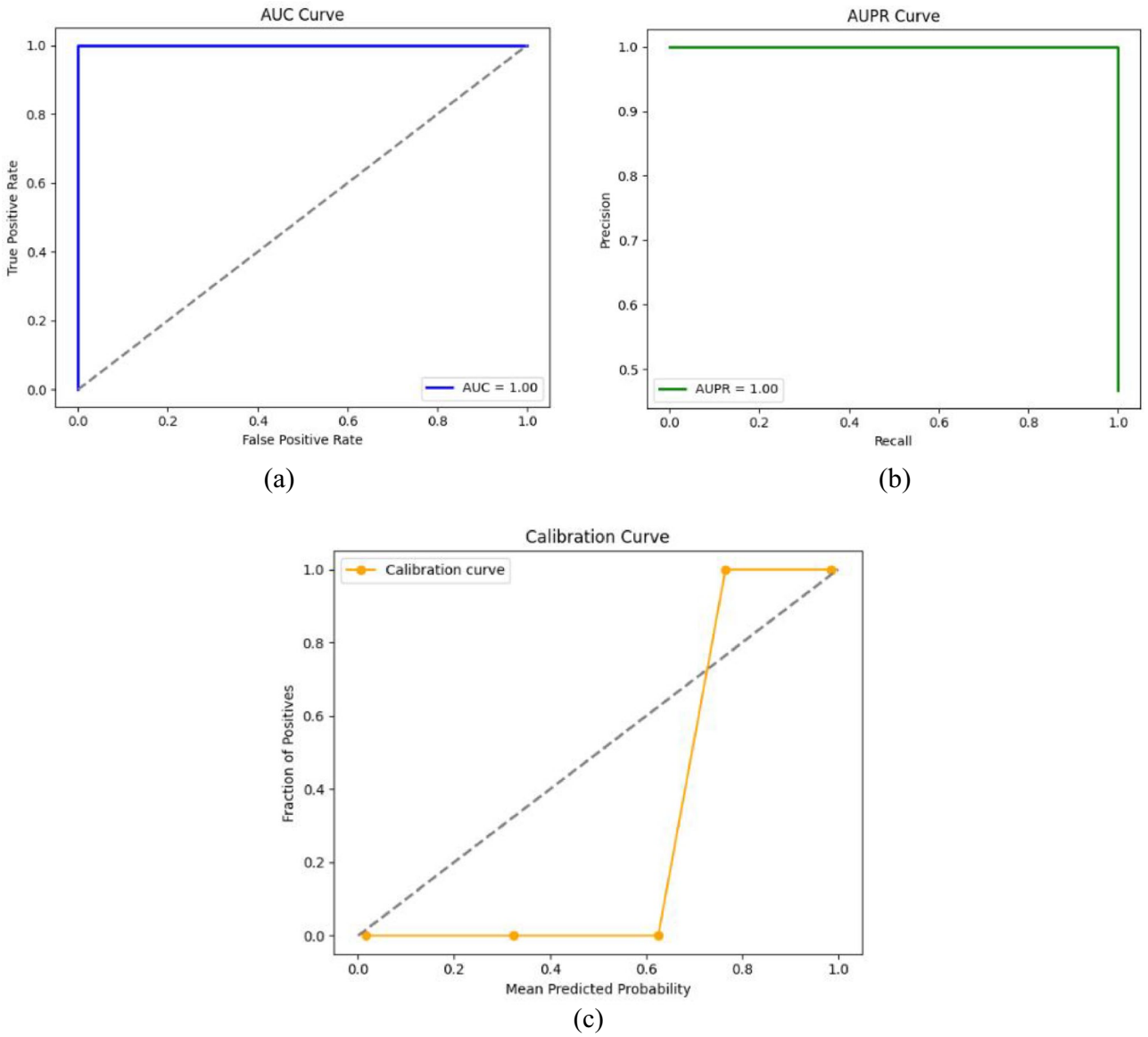


Fig. 12 Performance indicator curves of the proposed model for LDA feature set: **a** AUC Curve, **b** AUPR curve, **c** calibration curve

the relevance and redundancy scores of each sample, which requires O(mk) time. The overall time complexity of the feature optimization technique is O(mk), as $k <= f$. On the other hand, the total time complexity without feature optimization is O(mf). However, our selected EML models, such as bagging, boosting, stacking, and voting, consist of ML algorithms, where the time complexity of these EML models relies on the architecture of ML models (Kearns, 1990); thus, they aren't directly commensurate with the proposed approach on the basis of these aspects, i.e., the number of characteristics, the number of instances, etc. However, among the comparable approaches, considering $k << f$, feature optimization techniques execute with the least computational complexity, O(mk).

4.2 SHAP result analysis

In Fig. 13, we present the top 9 SHAP value features and sum of other 13 features less important for each class in the PD data prediction model (PD and Non-PD classes). A Beeswarm scatter plot showcases the distribution of SHAP values for each feature, with characteristics arranged in descending order based on their highest SHAP value. This particular type of scatter plot effectively addresses the challenge of data overlap by distributing numerous distinct assessments. In this representation, the x-axis corresponds to data points, while the y-axis illustrates the average data density. In our comprehensive analysis, the color blue indicates features with relatively lower impact, while the color purple

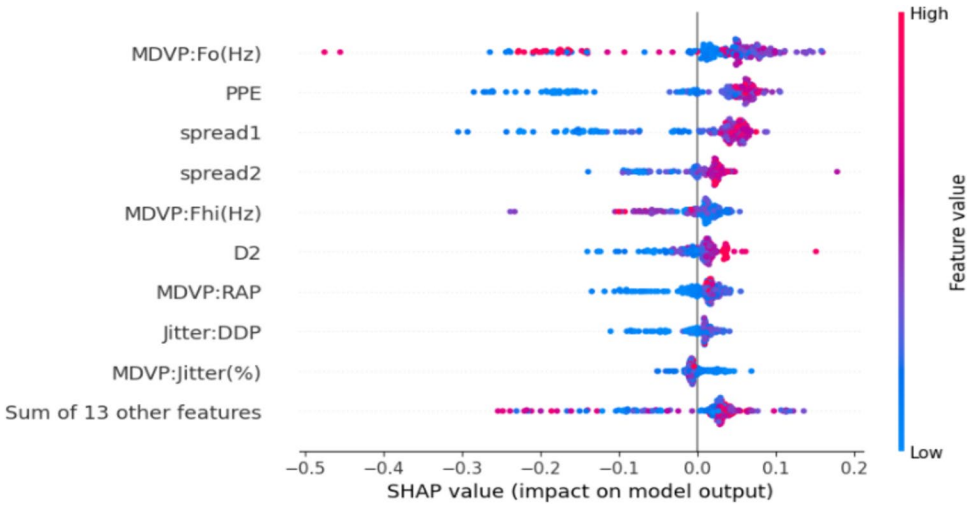


Fig. 13 Beeswarm plot for stacking EML model

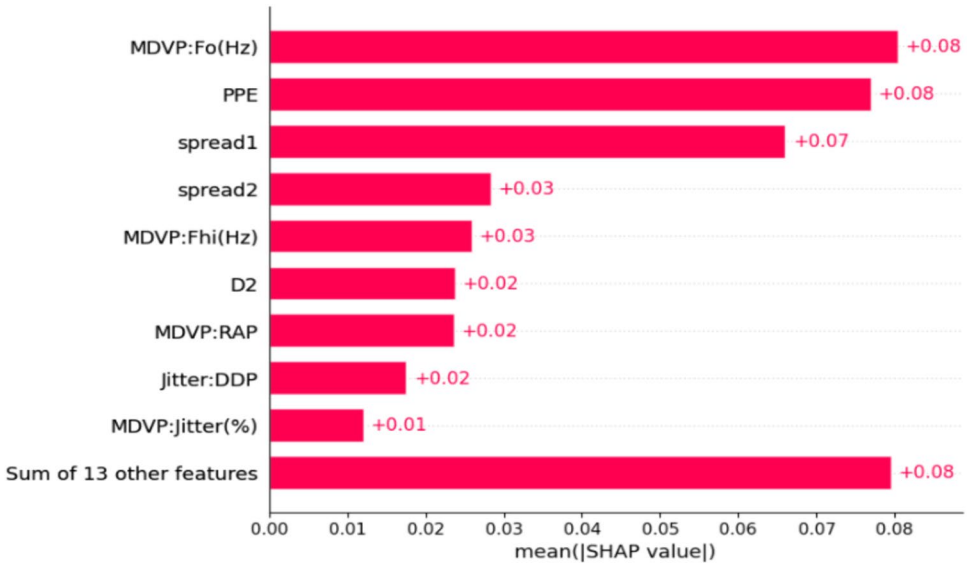
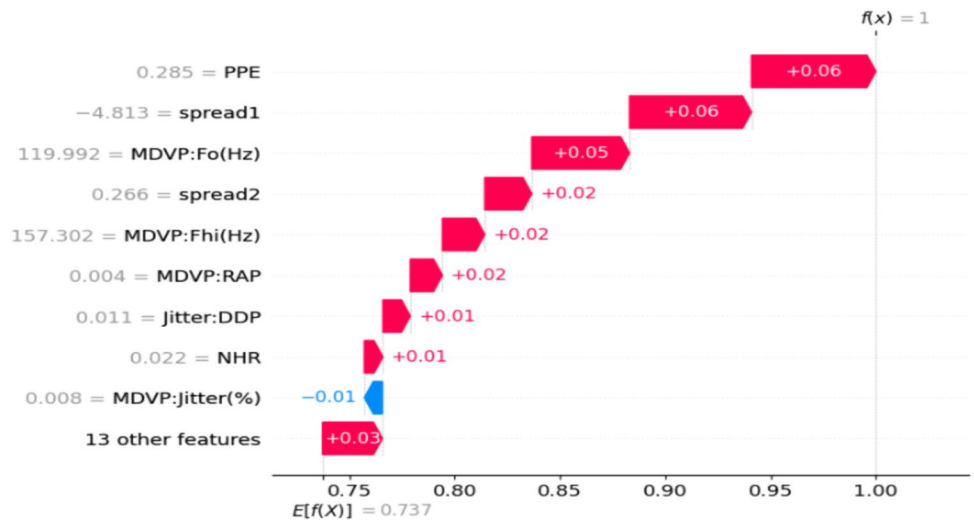


Fig. 14 Bar plot for stacking EML model

suggests that feature effects are progressively increasing. Finally, the color red highlights the feature with the most pronounced impact within our model framework. Analysis of Fig. 13 also reveals MDVP:Fo(Hz) as the primary feature influencing the stacking EML model's prediction of PD. As the MDVP feature value increases, its effect on the model's predictions intensifies, whereas a decrease in the MDVP feature value diminishes its impact on the predictive model. Following closely are PPE and Spread1, which emerge as the subsequent significant attributes for PD prediction. Similarly, Spread2, MDVP:Fhi(Hz), and D2 rank prominently as key factors for the stacking EML model. On the other hand, 13 features combinedly have few impact on PD detection.

In our endeavor to enhance comprehension of model outputs, we present additional SHAP plots for broader accessibility of PD detection professionals. Figure 14 illustrates a bar plot depicting the proposed stacking EML model. In this figure of SHAP global interpretations, the x-axis represents absolute SHAP values, while the y-axis indicates the parameters of significance in descending order. Here, MDVP:Fo(Hz) and PPE hold the highest importance notably greater (+0.08) than any other feature, while, Spread1 is the second-highest feature (+0.07). Similarly, Spread2 and MDVP:Fhi(Hz) exhibit absolute SHAP values of +0.03. Features such as D2, MDVP: RAP, and Jitter:DDP record scores of +0.02, while MDVP:Jitter (%) concludes with a score of +0.01. While, the combined impact of 13 features on PD detection is minimal, amounting to approximately 0.08.

For further exploration, the waterfall plot depicted in Fig. 15 delineates the impact of each optimized feature on the stacking EML model's output. Notably, both PPE and spread1 exhibit a SHAP score of +0.06, affirming their favorable influence on outcomes. Conversely, MDVP:Jitter



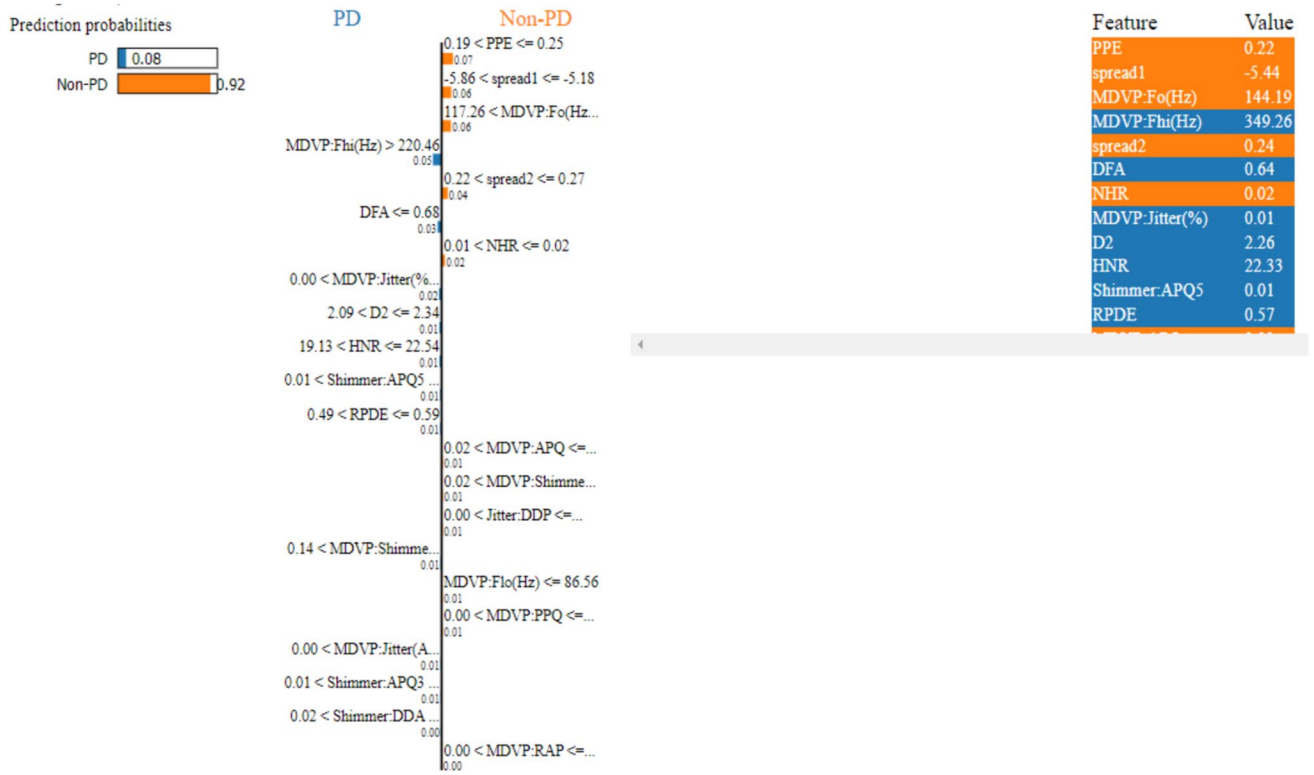


Fig. 15 Waterfall plot for stacking EML model

Fig. 16 Prediction probabilities graph produce by the LIME XAI method

(%) contributes negatively to the model output, having a SHAP score of -0.01. Features situated both above and below MDVP:Jitter (%) portray positive SHAP scores, thereby enhancing predictive accuracies. The computation of SHAP scores involves E[f(x)]—f(x), offering valuable insights into feature effects. On the other hand, the combined impact of 13 features on PD detection is minimal, amounting to approximately 0.03.

4.3 LIME result analysis

In elucidating tabular-format Parkinson's disease UCI datasets for particular instances, LIME emerges as a dependable tool for interpretable depiction. Figure 12 illustrates the graphical portrayal of prediction probabilities derived from the LIME XAI approach, explaining the anticipated outcomes of the stacking EML model. Here, the upper left portion of Fig. 10 shows the prediction score of patients

diagnosed with affected PD cases was 8% and non-PD cases were 92%.

In Fig. 16, we designate the color orange to signify target non-PD class and blue to denote target PD cases. These colors visually indicate the weight assigned to each feature for their respective predicted classes. The color intensity reflects local positive or negative weights, with longer color bars indicating greater weight in non-PD patients. Further, we visualize that five important features presented by the bar diagram: PPE (0.22), spread1 (-5.44), MDVP:Fo(Hz) (144.19), spread2 (0.24), and NHR (0.02) contributed to the prediction of non-PD affected patients. The feature MDVP:Fo(Hz) around 144 suggests the significance of this key feature in the stacking EML model for predicting non-PD cases.

Integrating advanced machine learning approaches into clinical practice requires a technological strategy. Clinicians may need substantial training to understand the workings and limitations of ML models, particularly in terms of interpretability and decision-making transparency. Developing user-friendly interfaces and incorporating explainable AI techniques can help clinicians trust and effectively use these systems. Additionally, ML models must integrate into existing clinical workflow and ensure interoperability with electronic health record (EHR) systems. To overcome resistance to new technology, phased implementation with extensive testing and validation in real-world scenarios, coupled with continuous feedback from clinicians, can build confidence. Furthermore, cost concerns can be solved through evidence of improved patient outcomes, efficiency, and reduction of medical errors.

Implementing the proposed system in a clinical setting involves addressing several practical challenges through careful planning and execution. Integration with existing workflows can be made by designing the system to seamlessly interface with electronic health records (EHRs) and clinical decision support systems, ensuring that it complements rather than disrupts current processes. Cost concerns can be managed by leveraging cloud-based solutions to minimize upfront expenses and reduce maintenance costs. To address the need for clinician training, the system should include user-friendly interfaces and comprehensive training modules that facilitate easy adoption. Additionally, ongoing support and updates can help clinicians stay current with system functionalities and improvements. Collaborating with clinical stakeholders during development ensures the system meets practical needs and integrates smoothly into daily operations, ultimately enhancing its effectiveness and user acceptance.

The practical implications of the proposed approach are substantial for both clinical practice and patient management. In real life, such an approach can be implemented to analyze diverse data sources—like clinical records, genetic information, and motor symptoms—to identify patterns and predict the likelihood of Parkinson's disease with high accuracy. In real-world scenarios, this approach can be applied to enable earlier diagnosis and personalized treatment plans, potentially improving patient outcomes and quality of life. By adapting this predictive model to electronic health records (EHRs) and decision support systems, clinicians can proactively monitor at-risk individuals more effectively. Additionally, this approach can support research into disease progression and treatment efficacy, ultimately contributing to advancements in Parkinson's disease management and therapeutic development.

4.4 Relating proposed LDA feature selection approach to explainability

We applied several feature optimization techniques, including PCA, LDA, and mRMR, alongside ensemble machine learning models such as bagging, boosting, voting, and stacking to improve the model's interpretability. These methods reduce data dimensionality by selecting or transforming features into simpler forms. For example, PCA transforms features into uncorrelated components, allowing the model to focus on the most informative aspects, while LDA maximizes class separation, making decision boundaries clearer, and mRMR, in turn, selects features that are both relevant and non-redundant. By integrating these optimization techniques, the ensemble models can prioritize the most critical features, leading to enhanced performance with greater transparency and interpretability. The LDA feature optimization technique demonstrates promising results in distinguishing between PD and non-PD patients, further validated by feature explainability methods such as SHAP and LIME. Table 10 summarizes the performance metrics of various EML models using LDA-optimized reduced feature subsets. The stacking EML model achieves near-perfect scores, approximately 100%, in accuracy, sensitivity, specificity, and F1-score. However, the Kappa indicator is crucial for identifying the most efficient model. Therefore, the stacking EML model, utilizing nine important features selected by LDA from a total of 24, proves to be the most effective.

The SHAP and LIME values indicate that features such as MDVP: Fo, PPE, spread1, spread2, MDVP: Fhi, D2, MDVP: RAP, jitter: DDP, and MDVP: jitter(%) significantly impact the stacking EML method, surpassing the combined effect of the other 13 features. This impact is illustrated through beeswarm, bar, and waterfall plots in Figs. 9, 10, and 11, respectively. Thus, our proposed ensemble method with feature optimization using LDA is well-fitted also for an explainable approach.

5 Conclusion and future scope

Accurate and timely diagnosis of Parkinson's Disease (PD) is critical for effective treatment. This research aimed to develop a highly accurate and transparent diagnostic model by leveraging optimized clinical features. Our data preprocessing strategy included three main steps: outlier detection to eliminate irrelevant data, SMOTE-Tomek to address class imbalance, and data augmentation to expand the dataset. Four feature optimization techniques—SelectKBest, minimum redundancy maximum relevance (mRMR), principal component analysis (PCA), and linear discriminant analysis (LDA)—were applied to extract the most relevant features from the clinical data. While feature optimization typically enhances machine learning model efficiency, selecting the appropriate method is important. Our findings revealed that not all optimization techniques improved model performance; in some cases, performance declined compared to models without optimization. Nevertheless, the LDA-based feature selection, when combined with our stacking ensemble machine learning (EML) model, achieved the 97.01% accuracy with a new large dataset. Throughout the study, the stacking technique consistently outperformed other EML models across all feature sets, even in the absence of feature optimization. To enhance model transparency and build clinician trust, we incorporated explainable AI methods—SHAP and LIME—into our framework. These tools provided both global and local interpretability, ensuring the model's decision-making process was understandable and transparent. Future research will focus on expanding the dataset and further refining the model to enhance its generalizability and reliability for broader clinical applications.

Funding No funding is received for this research.

Data availability We used two publicly available datasets. The first dataset can be accessed via the link: https://doi.org/10.24432/C59C74 and the second dataset can be accessed via the link: https://doi.org/10.24432/C5MS4X

Declarations

Conflict of interest The authors declare that there are no conflicts of interest.

Consent to participate Not required.

Ethical approval Not required.

References

Aich, S., Kim, H.-C., Hui, K. L., Al-Absi, A. A., & Sain, M. (2019). A supervised machine learning approach using different feature selection techniques on voice datasets for prediction of Parkinson's disease. In *2019 21st international conference on advanced communication technology (ICACT 2019)* (pp. 1116–1121). IEEE.

Al-Tam, R. M., Hashim, F. A., Maqsood, S., Abualigah, L., & Alwhaibi, R. M. (2024). Enhancing Parkinson's disease diagnosis through stacking ensemble-based machine learning approach. *IEEE Access, 12*, 79549–79567. https://doi.org/10.1109/ACCESS.2024.3408680

Alotaibi, A., et al. (2023). Explainable ensemble-based machine learning models for detecting the presence of cirrhosis in Hepatitis C patients. *Computation, 11*(6), 104.

Alshammri, R., Alharbi, G., Alharbi, E., & Almubark, I. (2023). Machine learning approaches to identify Parkinson's disease using voice signal features. *Frontiers in Artificial Intelligence, 6*, 1084001.

Asmae, O., Saleh, S., Abdelhadi, R., & Bachir, B. (2024). Enhancing Parkinson's disease diagnosis: A stacking ensemble approach leveraging machine learning techniques. In *2024 4th international conference on innovative research in applied science, engineering and technology (IRASET)* (pp. 1–7). FEZ. https://doi.org/10.1109/IRASET60544.2024.10549375.

Avuçlu, E., & Elen, A. (2020). Evaluation of train and test performance of machine learning algorithms and Parkinson diagnosis with statistical measurements. *Medical & Biological Engineering & Computing, 58*, 2775–2788.

Bind, S., Tiwari, A. K., & Sahani, A. K. (2015). A survey of machine learning based approaches for Parkinson disease prediction. *International Journal of Computing Science and Information Technology, 6*(2), 1648–1655.

Biswas, S., Mostafiz, R., Paul, B. K., Uddin, K. M. M., Hadi, Md. A., & Khanom, F. (2024). DFU_XAI: A deep learning-based approach to diabetic foot ulcer detection using feature explainability. *Biomedical Materials & Devices*. https://doi.org/10.1007/s44174-024-00165-5

Biswas, S., Mostafiz, R., Paul, B. K., Uddin, K. M. M., Rahman, M. M., & Shariful, F. N. U. (2023). DFU_MultiNet: A deep neural network approach for detecting diabetic foot ulcers through multi-scale feature fusion using the DFU dataset. *Intelligence-Based Medicine, 8*, 100128.

Biswas, S., Mostafiz, R., Uddin, M. S., & Paul, B. K. (2024). XAI-FusionNet: Diabetic foot ulcer detection based on multi-scale feature fusion with explainable artificial intelligence. *Heliyon, 10*(10), e31228.

Boukerche, A., Zheng, L., & Alfandi, O. (2021). Outlier detection: Methods, models, and classification. *ACM Computing Surveys, 53*(3), 1–37. https://doi.org/10.1145/3381028

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140.

Bukhari, S. N. H., & Ogudo, K. A. (2024). Ensemble machine learning approach for Parkinson's disease detection using speech signals. *Mathematics, 12*(10), 1575.

Celik, E., & Omurca, S. I. (2019). Improving Parkinson's disease diagnosis with machine learning methods. In *2019 scientific meeting on electrical-electronics & biomedical engineering and computer science (EBBT)*. IEEE.

Chaurasia, V., & Chaurasia, A. (2023). Detection of Parkinson's disease by using machine learning stacking and ensemble method. *Biomedical Materials & Devices, 1*(2), 966–978. https://doi.org/10.1007/s44174-023-00079-8

Chen, H.-L., Wang, G., Ma, C., Cai, Z.-N., Liu, W.-B., & Wang, S.-J. (2016). An efficient hybrid kernel extreme learning machine approach for early diagnosis of Parkinson's disease. *Neurocomputing, 184*, 131–144.

Das, R. (2010). A comparison of multiple classification methods for diagnosis of Parkinson disease. *Expert Systems with Applications, 37*(2), 1568–1572.

Dhanalakshmi, S., Das, S., & Senthil, R. (2024). Speech features-based Parkinson's disease classification using combined SMOTE-ENN and binary machine learning. *Health Technology, 14*(2), 393–406. https://doi.org/10.1007/s12553-023-00810-x

Došilović, F. K., Brčić, M., & Hlupić, N. (2018). Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)* (pp. 210–215). IEEE.

Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: A methodology review. *Journal of Biomedical Informatics, 35*(5–6), 352–359.

Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research, 61*, 863–905.

Freund, Y., Schapire, R., & Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society for Artificial Intelligence, 14*(771–780), 1612.

Georgiev, D., Hamberg, K., Hariz, M., Forsgren, L., & Hariz, G.-M. (2017). Gender differences in Parkinson's disease: A clinical perspective. *Acta Neurologica Scandinavica, 136*(6), 570–584. https://doi.org/10.1111/ane.12796

Hoo, K. A., Tvarlapati, K. J., Piovoso, M. J., & Hajare, R. (2002). A method of robust multivariate outlier replacement. *Computers & Chemical Engineering, 26*(1), 17–39.

International Congress of Parkinson's Disease and Movement Disorders®. (2024). Accessed April 16, 2024. Available: https://www.mdscongress.org/.

Jakkula, V. (2006). Tutorial on support vector machine (SVM). *School of EECS, Washington State University, 37*(2), 3.

Jani, R., Shanto, M. S. I., Kabir, M. M., Rahman, M. S., & Mridha, M. F. (2022). Heart disease prediction and analysis using ensemble architecture. In *2022 international conference on decision aid sciences and applications (DASA 2022)* (pp. 1386–1390). IEEE.

Kearns, M. J. (2024). The computational complexity of machine learning. MIT Press, 1990. Accessed 28 May 2024. https://books.google.com/books?hl=en&lr=&id=y5Txq1AkJoMC&oi=fnd&pg=PA1&dq=Kearns,+M.J.,+1990.+The+computational+complexity+of+machine+learning.+MIT+press.&ots=_RFHz_dyLk&sig=YxT4XlJunS0qcktt61NFyAfnrBs.

Kluyver, T., et al. (2016). Jupyter Notebooks: A publishing format for reproducible computational workflows. In *Positioning and power in academic publishing: Players, agents and agendas*.

Krajna, A., Kovac, M., Brcic, M., & Šarčević, A. (2022). Explainable artificial intelligence: An updated perspective. In *2022 45th jubilee international convention on information, communication and electronic technology (MIPRO)* (pp. 859–864). IEEE.

Kuo, C.-C., et al. (2019). Automation of the kidney function prediction and classification through ultrasound-based kidney imaging using deep learning. *NPJ Digital Medicine, 2*(1), 29.

Lamba, R., Gulati, T., Alharbi, H. F., & Jain, A. (2022). A hybrid system for Parkinson's disease diagnosis using machine learning techniques. *International Journal of Speech Technology, 8*, 1–11.

Liang, H., et al. (2019). Evaluation and accurate diagnoses of pediatric diseases using artificial intelligence. *Nature Medicine, 25*(3), 433–438.

Liang, M., et al. (2021). A stacking ensemble learning framework for genomic prediction. *Frontiers in Genetics, 12*, 600040.

Little, M. (2008). Parkinsons. *UCI Machine Learning Repository*. https://doi.org/10.24432/C59C74

Lundberg, S. M., Nair, B., Vavilala, M. S., Horibe, M., Eisses, M. J., Adams, T., Liston, D. E., Lo, D. K.-W., Newman, S.-F., Kim, J., & Lee, S.-I. (2018). Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering, 2*(10), 749–760.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems, 30*, 89.

Mahesh, T. R., Bhardwaj, R., Khan, S. B., Alkhaldi, N. A., Victor, N., & Verma, A. (2024). An artificial intelligence-based decision support system for early and accurate diagnosis of Parkinson's disease. *Decision Analytics Journal, 10*, 100381.

Mamun, M., Mahmud, M. I., Hossain, M. I., Islam, A. M., Ahammed, M. S., & Uddin, M. M. (2022). Vocal feature guided detection of Parkinson's disease using machine learning algorithms. In *2022 IEEE 13th annual ubiquitous computing, electronics & mobile communication conference (UEMCON 2022)* (pp. 566–572). IEEE.

Marella, W. M., Sparnon, E., & Finley, E. (2017). Screening electronic health record–related patient safety reports using machine learning. *Journal of Patient Safety, 13*(1), 31–36.

Martinez-Millana, A., et al. (2018). Optimisation of children z-score calculation based on new statistical techniques. *PLoS ONE, 13*(12), e0208362.

Mohi Uddin, K. M., Biswas, N., Rikta, S. T., Dey, S. K., & Qazi, A. (2023). XML-LightGBMDroid : A self-driven interactive mobile application utilizing explainable machine learning for breast cancer diagnosis. *Engineering Reports, 5*(11), 12666. https://doi.org/10.1002/eng2.12666

Mostafiz, R., Rahman, M. M., Kumar, P. K. M., & Islam, M. A. (2017). Speckle noise reduction for D ultrasound images by optimum threshold parameter estimation of wavelet coefficients using Fisher discriminant analysis. *International Journal of Imaging and Robotics, 17*(4), 73–88.

Mostafiz, R., Rahman, M. M., Kumar, P. K. M., & Islam, M. A. (2018). Speckle noise reduction for 3D ultrasound images by optimum threshold parameter estimation of bi-dimensional empirical mode decomposition using Fisher discriminant analysis. *International Journal of Signal and Imaging Systems Engineering, 11*(2), 93. https://doi.org/10.1504/IJSISE.2018.091886

Mostafiz, R., Uddin, M. S., Alam, N.-A., Hasan, M. M., & Rahman, M. M. (2021). MRI-based brain tumor detection using the fusion of histogram-oriented gradients and neural features. *Evolutionary Intelligence, 14*, 1075–1087.

Nahar, N., Ara, F., Neloy, Md. A. I., Biswas, A., Hossain, M. S., & Andersson, K. (2021). Feature selection based machine learning to improve prediction of Parkinson disease. In M. Mahmud, M. S. Kaiser, S. Vassanelli, Q. Dai & N. Zhong (Eds.), *Brain informatics* (pp. 496–508). Lecture Notes in Computer Science. Springer. https://doi.org/10.1007/978-3-030-86993-9_44

Nilashi, M., et al. (2022). Predicting Parkinson's disease progression: Evaluation of ensemble methods in machine learning. *Journal of Healthcare Engineering, 2022*(1), 2793361.

Nissar, I., Rizvi, D., Masood, S., & Mir, A. (2019). Voice-based detection of Parkinson's disease through ensemble machine learning approach: A performance study. *EAI Endorsed Transactions on Pervasive Health and Technology, 5*(19), 162806. https://doi.org/10.4108/eai.13-7-2018.162806

Oguri, V. S. B., Poda, S., Satya, A. K., & Prasanna, N. K. (2023). Parkinson's disease detection using tree based machine learning algorithms. *Current Trends in Biotechnology and Pharmacy, 17*(2), 808–818.

Park, D. S. et al. (2019). SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech* 2019 (pp. 2613–2617). https://doi.org/10.21437/Interspeech.2019-2680.

Parkinson disease. https://www.who.int/news-room/fact-sheets/detail/parkinson-disease. Accessed 16 Apr 2024.

Polat, K. (2019). A hybrid approach to Parkinson disease classification using speech signal: The combination of smote and random forests. In *2019 scientific meeting on electrical-electronics & biomedical engineering and computer science (EBBT)* (pp. 1–3). IEEE.

Polikar, R. (2012). Ensemble learning. In C. Zhang & C. Ma (Eds.), *Ensemble machine learning: Methods and applications* (pp. 1–34). Springer.

Radovic, M., Ghalwash, M., Filipovic, N., & Obradovic, Z. (2017). Minimum redundancy maximum relevance feature selection approach for temporal gene expression data. *BMC Bioinformatics, 18*(1), 9. https://doi.org/10.1186/s12859-016-1423-9

Rasheed, J., Hameed, A. A., Ajlouni, N., Jamil, A., Özyavaş, A., & Orman, Z. (2010). Application of adaptive back-propagation neural networks for Parkinson's disease prediction. In *2020 international conference on data analytics for business and industry: Way towards a sustainable economy (ICDABI 2020)* (pp. 1–5). IEEE.

Reddy, K. V. A., Ambati, S. R., Reddy, Y. S. R., & Reddy, A. N. (2021). AdaBoost for Parkinson's disease detection using robust scaler and SFS from acoustic features. In *2021 smart technologies, communication and robotics (STCR 2021)* (pp. 1–6). IEEE.

Rehman, A., Saba, T., Mujahid, M., Alamri, F. S., & ElHakim, N. (2023). Parkinson's disease detection using hybrid LSTM-GRU deep learning model. *Electronics, 12*(13), 2856.

Rikta, S. T., Uddin, K. M. M., Biswas, N., Mostafiz, R., Sharmin, F., & Dey, S. K. (2023). XML-GBM lung: An explainable machine learning-based application for the diagnosis of lung cancer. *Journal of Pathology Informatics, 14*, 100307.

Sakar, C., Serbes, G., Gunduz, A., Nizam, H., & Sakar, B. (2018). Parkinson's disease classification. *UCI Machine Learn. Repository, 10*, 7.

Saleh, S., Cherradi, B., El Gannour, O., Hamida, S., & Bouattane, O. (2024). Predicting patients with Parkinson's disease using machine learning and ensemble voting technique. *Multimedia Tools and Applications, 83*(11), 33207–33234.

Saria, S., Koller, D., & Penn, A. (2010). Learning individual and population level traits from clinical temporal data. In *Proceedings of neural information processing systems* (pp. 1–9). Citeseer.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning, 5*(2), 197–227. https://doi.org/10.1007/BF00116037

Senturk, Z. K. (2020). Early diagnosis of Parkinson's disease using machine learning algorithms. *Medical Hypotheses, 138*, 109603.

Sharma, P., Sundaram, S., Sharma, M., Sharma, A., & Gupta, D. (2019). Diagnosis of Parkinson's disease using modified grey wolf optimization. *Cognitive Systems Research, 54*, 100–115.

Shastry, K. A. (2023). Ensemble machine learning regression model based predictive framework for Parkinson's UPDRS motor score prediction from speech data. *International Journal of Speech Technology, 26*(2), 433–457.

Solana-Lavalle, G., Galán-Hernández, J.-C., & Rosas-Romero, R. (2020). Automatic Parkinson disease detection at early stages as a pre-diagnosis tool by using classifiers and a small set of vocal features. *Biocybernetics and Biomedical Engineering, 40*(1), 505–516.

Song, F., Guo, Z., & Mei, D. (2010). Feature selection using principal component analysis. In *2010 international conference on system science, engineering design and manufacturing informatization* (pp. 27–30). IEEE.

Sveinbjornsdottir, S. (2016). The clinical symptoms of Parkinson's disease. *Journal of Neurochemistry, 139*(S1), 318–324. https://doi.org/10.1111/jnc.13691

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5*(2), 241–259.

Yadav, D. C., & Pal, S. (2020). Prediction of thyroid disease using decision tree ensemble method. *Human-Intelligent Systems Integration, 2*(1–4), 89–95. https://doi.org/10.1007/s42454-020-00006-y

Yasar, A., Saritas, I., Sahman, M. A., & Cinar, A. C. (2019). Classification of Parkinson disease data with artificial neural networks. *IOP Conference Series: Materials Science and Engineering, 675*(1), 012031.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com