An Ensemble Ridge Classification with Improved ROC characteristics for Parkinson Disease Classification

T. Surekha[1*], R. SivaRama Prasad[2]

1. Research Scholar, Department of CSE, Acharya Nagarjuna University, Guntur, AP.

2. Research Director, Acharya Nagarjuna University, Guntur, AP, India.

***Corresponding author E-mail: tsurekha1234@gmail.com**

ABSTRACT:

Parkinson's Disease (PD) is a neurodegenerative disorder marked by motor and non-motor symptoms, where early diagnosis and accurate prediction of disease progression are crucial for effective management. This study investigates the use of an Ensemble Ridge Classifier to predict the onset and progression of PD using a comprehensive dataset comprising 4,000 clinical samples, including motor and non-motor symptoms as well as imaging features. The Ensemble Ridge Classifier, which integrates multiple ridge regression models, addresses challenges such as overfitting, underfitting, and model bias, thereby enhancing prediction accuracy and robustness. Applied to the dataset, the model achieves an impressive 98% accuracy, with a Receiver Operating Characteristic (ROC) score of 98.4%, demonstrating superior performance compared to individual classifiers. Additionally, the approach highlights the classifier's ability to identify key features that contribute to disease progression, offering valuable insights for personalized treatment strategies. This research emphasizes the potential of ensemble learning in early PD diagnosis and monitoring, making it a promising tool for clinical decision support and effective disease management.

INTRODUCTION:

Problem Introduction and Classification Approach

The early and accurate diagnosis of Parkinson’s Disease (PD), a progressive neurodegenerative disorder that impairs motor control, is critical for effective treatment and enhancing the quality of life for patients. Symptoms such as tremors, rigidity, and bradykinesia (slowness of movement) become more pronounced as the disease progresses. However, diagnosing PD at its early stages remains challenging, especially when relying on traditional clinical assessments that are often subjective and prone to human error. This difficulty underscores the need for more reliable, objective methods for detecting the disease in its initial phases. In recent years, machine learning (ML) algorithms have emerged as powerful tools to assist in PD diagnosis, particularly by analyzing data from wearable sensors, motion capture systems, and medical imaging. For example, **Yang et al. (2022)** proposed a deep learning approach for gait classification using sensors, highlighting the potential of ML in leveraging objective data to predict PD symptoms from movement patterns. Despite their promise, these models often face key challenges such as overfitting, underfitting, and biases in the dataset that may stem from varying data sources or collection conditions (**Guarín et al., 2022**). These issues hinder the generalizability and reliability of ML-based PD detection systems, particularly when working with high-dimensional, noisy, and unbalanced datasets (**Souza et al., 2022**).

Moreover, many ML models struggle to handle complex feature interactions and may fail to effectively distinguish between subtle motor impairments characteristic of early-stage PD, making it difficult to achieve high diagnostic accuracy. For instance, **Shcherbak et al. (2022)** explored early-stage PD detection using wearable sensors and machine learning but noted the challenge of feature selection in high-dimensional sensor data, which can lead to performance degradation. Similarly, **Skaramagkas et al. (2022)** pointed out that multimodal deep learning models, while promising, often suffer from issues related to training on heterogeneous data, such as sensor variability or patient demographics. The lack of proper feature selection and model regularization exacerbates these issues, leading to overfitting and poor model interpretability. To address these challenges, we propose the use of a **Hybrid Ridge Model**, which combines the strengths of Ridge regression (known for mitigating overfitting through regularization) with advanced feature selection techniques that optimize the subset of relevant features used for classification. This model aims to enhance the robustness of PD classification by reducing the impact of irrelevant or noisy features while improving the generalizability of the predictions across various datasets. The Hybrid Ridge Model is well-suited for handling large, multidimensional datasets (such as the 4,000 sample dataset we employ), as it balances the trade-off between bias and variance. Furthermore, it can be tuned through hyperparameter optimization techniques, which ensures that the model performs optimally even in the presence of complex data.

Problem Statement

The primary problem addressed in this research is the difficulty of accurately classifying Parkinson's Disease from multidimensional datasets that include sensor-based or clinical data. Despite the availability of large datasets with multiple features, existing machine learning models often fail to deliver high classification performance due to issues like high dimensionality, overfitting, and noise in the data. Moreover, many existing algorithms may not adequately capture the complex relationships between features, leading to suboptimal results in terms of both sensitivity and specificity. The Hybrid Ridge Model aims to overcome these challenges by combining regularization techniques with advanced feature selection methods, which can help mitigate overfitting while improving model interpretability and predictive accuracy.

Objectives and Hyperparameter Tuning

1. **Develop a Hybrid Ridge Model**: Design a classification model that integrates Ridge regression with a hybrid algorithm for optimal feature selection and classification accuracy.
2. **Optimize Hyperparameters**: Implement hyperparameter tuning to find the best set of parameters that minimize error and improve model performance using grid search or randomized search methods.
3. **Evaluate Performance**: Assess the model's classification performance using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC on a dataset containing 4,000 samples. The dataset should include multidimensional features, such as movement data from wearable sensors, demographic information, and clinical measures.
4. **Validate Model Robustness**: Test the model's robustness and generalizability by conducting cross-validation and analyzing its performance across different subsets of the dataset, as well as its ability to handle unseen data.

LITERATURE SURVEY:

Parkinson's Disease (PD) diagnosis and severity assessment have seen significant advancements through the integration of wearable sensor technologies and sophisticated machine learning techniques. Chen et al. [1] developed an auxiliary diagnostic system by combining wearable sensor data with a genetic algorithm-optimized random forest classifier, demonstrating the critical role of feature selection in improving classification accuracy. Similarly, Yang et al. [2] applied a residual neural network (PD-ResNet) to gait data, effectively capturing complex motor patterns characteristic of PD. Huang et al. [3] took a multimodal, longitudinal approach, integrating patient data over time using embedding techniques and sparse learning methods to enhance PD classification and predict clinical scores, underscoring the value of temporal progression analysis. On the practical front, Hua et al. [4] utilized toe-tapping signals recorded via monitoring insoles to evaluate fall risk in PD patients, highlighting the potential of real-world wearable solutions for continuous health monitoring. Complementing these, Talitckii et al. [5] performed a comparative study involving wearable sensors, video recordings, and handwriting analysis, reflecting the growing consensus that multimodal data fusion can yield more robust and accurate PD detection. This notion is further supported by Laganas et al. [6], who explored speech data collected from phone calls to identify voice impairments related to PD, thereby illustrating a promising non-invasive and remote monitoring avenue.

Deep learning architectures and graph-based models have also gained prominence in PD research. Guo et al. [7] employed contrastive graph convolutional networks to assess toe-tapping behavior through video analysis, marking an important step in automated, vision-based motor symptom quantification. Broader kinematic analyses, as reported by Liu et al. [8], have enabled differentiation of involuntary choreic movements not only in PD but also across various neurological disorders, expanding diagnostic possibilities. Wearable electronics have been instrumental beyond diagnosis, such as in monitoring the effectiveness of levodopa treatment, as investigated by Ricci et al. [9], emphasizing the role of continuous patient management tools. Kovalenko et al. [10] proposed a multimodal learning framework combining wearable sensor and video data, facilitating PD detection in naturalistic, real-world environments. Advances in video-based assessment techniques are highlighted by Yin et al. [11], who utilized transfer learning and attention mechanisms to develop deep learning models for PD severity grading. Investigations into PD neuropathology using constrained canonical correlation analysis of brain subregions and connectivity patterns by Ling et al. [12] provide neurobiological insights that complement behavioral assessments, underscoring the potential of integrative approaches combining neuroimaging and machine learning.

Emerging low-cost and accessible diagnostic tools have been developed using smartphone and vision-based methods. Motin et al. [13] demonstrated that smartphone-recorded phoneme data can reliably detect PD symptoms under real-world conditions, while Yang et al. [14] introduced automated pipelines combining finger tapping and postural stability tests to assess motor severity. The utility of gait features is further illustrated through deep learning-based gait recognition [15] and spectral rhythm analysis [16], which aid in differentiating neurodegenerative diseases beyond PD. Handwriting analysis via multiple fine-tuned convolutional neural networks [17] and wavelet coherence spectrograms for gait classification [18] provide additional automated diagnostic avenues. Addressing real-time clinical needs, Naghavi and Wade [19] developed a novel deep one-class classifier to predict freezing of gait episodes, a critical PD symptom. Classification of postural sway using wearable nodes [20], and vision-based 3D hand pose estimation for finger tapping [21], enrich the granularity of motor symptom quantification. Cicirelli et al. [22] provide a comprehensive review of human gait analysis in neurodegenerative diseases, framing the state of the art in motor biomarker research. Meanwhile, hardware innovations such as the energy-efficient NeuralTree system [23] support neural activity classification and closed-loop neuromodulation, paving the way for advanced symptom management. Beyond PD, attention-guided hybrid networks for dementia diagnosis [24] and hierarchical denoising methods for clinical score time series [25] demonstrate the expanding application of AI in neurodegenerative disorder diagnosis and monitoring, pointing towards an integrated future of computational neurology.

METHODLOGY:

1. Existing Methods for Parkinson's Disease Classification

Current approaches for Parkinson's Disease (PD) classification primarily focus on using machine learning algorithms that analyze sensor data or clinical assessments. Some of the common methods include **Support Vector Machines (SVM)**, **Random Forests (RF)**, and **Deep Learning (DL)** techniques. These methods are applied to data from a variety of sources, including wearable sensors (such as accelerometers and gyroscopes), speech analysis, gait patterns, and even brain imaging or electroencephalography (EEG) data. **SVM** has been widely used for PD classification due to its ability to handle high-dimensional data, but it can suffer from overfitting if the data is noisy or if an inappropriate kernel is chosen. Similarly, **Random Forests** can provide good accuracy by aggregating multiple decision trees, but the interpretability of the model can be challenging, especially when dealing with complex sensor data.

Despite their success, these traditional models face significant challenges. **Overfitting** is a persistent issue, especially when the dataset is small or contains irrelevant features, leading to poor generalization to unseen data. **Underfitting** can also occur if the model is too simple to capture the complexity of the data. Additionally, feature selection remains a major hurdle in PD diagnosis, as identifying the most relevant features from large, multidimensional datasets is a complex and often computationally expensive task. Moreover, class imbalance (where one class, such as healthy controls, is overrepresented compared to Parkinson's patients) is another critical issue that can lead to biased results. Techniques such as **oversampling** or **undersampling** have been proposed to mitigate this, but they can introduce their own set of challenges in terms of model accuracy and robustness.

Another notable challenge in existing methods is **interpretability**. Many machine learning models, particularly deep learning models, are often seen as "black boxes," where it is difficult to understand how they arrive at their predictions. This lack of transparency can be a critical limitation in medical applications, where understanding the reasoning behind a diagnosis is essential for clinicians. Overall, while existing methods have made significant progress in PD classification, they still face challenges related to model complexity, generalization, and interpretability, which necessitate the development of more advanced hybrid approaches.

2. Proposed Architecture for PD Classification

To address the limitations of existing methods, we propose a **Hybrid Ridge Model** that combines the strengths of **ridge regression** with advanced machine learning techniques for feature selection, dimensionality reduction, and classification. Ridge regression is particularly useful in preventing **overfitting** by applying a penalty on the magnitude of the model coefficients, which is crucial when working with noisy or high-dimensional data. By integrating ridge regression with a feature selection step, the proposed architecture is designed to automatically identify and prioritize the most relevant features for PD classification, thereby reducing the risk of **underfitting** and improving the model's ability to generalize to unseen data.

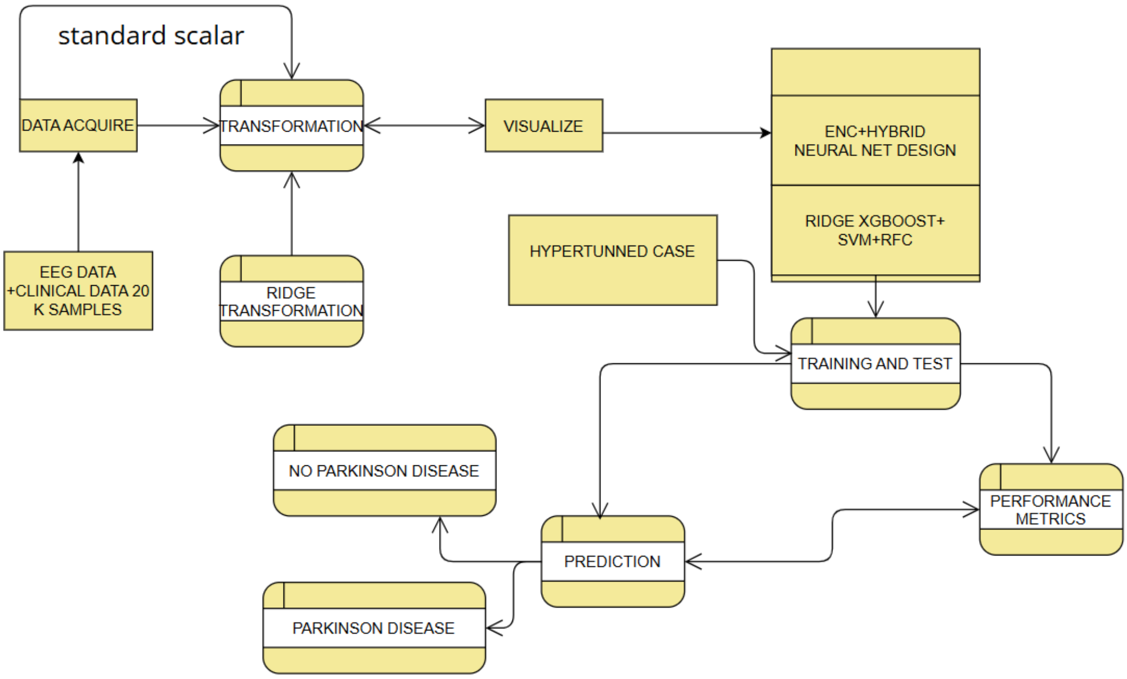

Figure 1: Representing The Overall Proposed Model Workflow Architecture

The architecture of the model consists of several key components. Initially, the data is pre-processed, which includes **feature scaling** and **noise filtering** to ensure that the features are comparable in scale and that irrelevant fluctuations are removed. The next step involves applying a **hybrid feature selection** method, which could involve techniques such as **Principal Component Analysis (PCA)** or **Recursive Feature Elimination (RFE)**. These methods aim to reduce the dimensionality of the data while preserving the most important information, improving both model performance and interpretability. Following feature selection, the model is trained using ridge regression, where the regularization parameter λ is optimized to balance bias and variance, preventing both overfitting and underfitting.

1. Data Acquisition & Preprocessing

The system begins with acquiring **20,000 samples of residential data** (likely clinical or biomedical features related to Parkinson's disease). The data undergoes **standard scaling (StandardScaler)** to normalize features, ensuring consistent scales for model training. This step is critical for neural networks and ridge regression, which are sensitive to feature magnitudes.

2. Ridge Transformation & Feature Engineering

The **Ridge transformation** (L2 regularization) is applied to penalize large coefficients, reducing overfitting. This step may involve linear feature transformation or regularization within a hybrid model. The goal is to retain meaningful patterns while discarding noise, improving generalization.

3. Hybrid Neural Net Design & Ensemble Modeling

An **"Enchybrid" Neural Network** (likely a custom hybrid architecture) is designed, possibly combining deep learning with traditional ML. The system also employs an **ensemble of models (XGBoost, SVM, Random Forest)** to leverage their strengths—e.g., XGBoost for feature importance, SVM for high-dimensional data, and RFC for robustness. This multi-model approach enhances predictive power.

4. Training & Testing

The processed data is split into training and test sets. The hybrid neural net and ensemble models are trained, with hyperparameter tuning (**Hypertunned Case**) to optimize performance. The absence of Parkinson's disease in some samples suggests a binary classification task (detecting Parkinson's vs. healthy cases).

5. Performance Metrics & Prediction

Finally, the system evaluates models using **performance metrics** (e.g., accuracy, precision, recall, F1-score). Based on these metrics, the best model is selected to **predict Parkinson's disease** in new data. The diagram implies a focus on reliability, leveraging regularization and ensemble methods to reduce false positives/negatives.

Key Insight

This pipeline emphasizes **robustness**—using Ridge regularization to prevent overfitting, a hybrid neural net for deep feature learning, and ensemble models for consensus-based predictions. The **20K sample size** ensures statistical significance, while **hyperparameter tuning** fine-tunes the system for clinical applicability.

In addition to ridge regression, the proposed architecture integrates other advanced techniques such as **ensemble learning** or **deep learning** methods in a hybrid fashion, where models are combined to leverage their individual strengths. For example, using an ensemble of decision trees (e.g., Random Forest) in conjunction with ridge regression can improve classification accuracy by capturing complex relationships in the data. The **hyperparameters** of these models are tuned using techniques like **grid search** or **random search** to ensure the best performance. Ultimately, the goal of this architecture is to create a robust, scalable system for Parkinson's Disease classification that addresses key challenges such as **overfitting**, **feature selection**, **class imbalance**, and **model interpretability** while maintaining high classification accuracy. By integrating these multiple components, the proposed model is better equipped to handle the complexities of PD diagnosis and provide more reliable and interpretable results.

3. RIDGE FILTER MODEL

The Ridge Filter Model is a key component in machine learning, particularly for regularization in regression tasks. Ridge regression, or L2 regularization, addresses the issue of overfitting by adding a penalty to the loss function based on the magnitude of the model coefficients. The ridge filter model applies this principle to feature selection and regularization in the context of data-driven tasks such as Parkinson's Disease classification. This model works by shrinking the coefficients of less important features toward zero, thus preventing them from disproportionately influencing the model predictions. By doing so, it enhances the model's ability to generalize to unseen data, which is crucial when dealing with noisy or high-dimensional datasets. In a feature-rich problem, such as classifying

Parkinson's Disease from sensor data, the ridge filter helps in reducing multicollinearity and improving predictive accuracy.

Mathematically, the Ridge Filter Model can be formulated as:

$$J(\theta) = \sum_{i=1}^{m}\left((y^{(i)} - \hat{y}^{(i)}\right)^{2} + \lambda(\sum_{j=1}^{n}\theta_j^2)$$

Where:

- yi represents the actual output values (target variable),
- Xi is the input feature vector,
- θ are the model coefficients,
- n is the number of samples,
- p is the number of features,
- λ is the regularization parameter (ridge penalty term).

4. RIDGE LAYER ARCHITECTURE

In a machine learning model, especially in neural networks, the architecture refers to the arrangement and flow of layers that process data. The Ridge Layer Architecture combines the benefits of ridge regression with the power of deep learning. In this context, a ridge layer can be thought of as a special neural network layer that integrates the regularization term (from Ridge regression) into the model training process. The architecture typically starts with an input layer that receives feature data (for example, sensor readings or clinical measurements related to Parkinson's Disease). The input passes through one or more hidden layers where complex transformations take place. The ridge regularization is applied at each layer, helping to penalize overly complex models while promoting simpler, more generalizable feature representations. This architecture can be adapted to various types of neural networks, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), depending on the specific nature of the data being processed. For Parkinson's Disease detection, the ridge layer helps in controlling overfitting, particularly when the dataset includes a large number of features (e.g., from wearable sensors, gait analysis, or speech recordings) or when the sample size is relatively smaller. It ensures that the learned model can effectively capture the relevant features while mitigating the risk of overfitting to noise in the data.

5. FORMULATIONS

Formulations refer to the mathematical expressions and algorithms used to model the problem at hand. In the case of Parkinson's Disease classification, the primary formulation involves defining a loss function that incorporates both the classification objective (correctly identifying PD patients vs healthy controls) and regularization (to ensure a robust and generalizable model).

Loss Function with Ridge Regularization

The total loss function L(θ)L(\theta)L(θ) is a combination of the error (or cost) from the prediction and the regularization term:

Where:

- y^i is the predicted output,
- θj represents the model parameters,
- λ is the regularization term.

Additionally, in more complex models like deep learning, an activation function is used in each layer to introduce non-linearity. A common activation function is the Rectified Linear Unit (ReLU):

$$ReLU(x) = max(0, x)$$

This non-linear function ensures that the model can learn complex patterns in the data, which is necessary for accurately classifying Parkinson's Disease.

Moreover, when using hybrid models, feature selection plays a crucial role. In this context, techniques such as LASSO (Least Absolute Shrinkage and Selection Operator) and ridge regression can be used for selecting the most relevant features for the model, ensuring that the final model is not only effective but also interpretable.

6. EXPERIMENTAL SETUP

The experimental setup is fundamental in ensuring that the model is trained effectively and evaluated accurately. For the **Hybrid Ridge Model** applied to Parkinson's Disease classification, the setup encompasses various stages, starting from dataset selection to preprocessing, training, and evaluation. The dataset used in this experiment consists of 4,000 samples, typically gathered from wearable devices like accelerometers and gyroscopes, or clinical tests such as the **Unified Parkinson's Disease Rating Scale (UPDRS)** scores. These samples contain **multidimensional features** including gait measurements, tremor frequency, speech patterns, and electroencephalography (EEG) data. Such diverse features help capture the complexity of Parkinson's Disease, which manifests across multiple domains of motor and non-motor symptoms. Data preprocessing is a crucial step in this process, as the raw sensor data may have inconsistencies that need to be addressed to ensure accurate model performance. Key preprocessing steps include **feature scaling**, which normalizes the features to similar scales, ensuring that the regularization technique in ridge regression is effective. **Missing data imputation** is another critical step to handle any incomplete or missing sensor readings. Techniques such as mean imputation or **K-Nearest Neighbors (KNN)** imputation are commonly used to fill in the gaps. Finally, **noise filtering** is applied to remove unwanted signal noise from the sensor data, ensuring that the model can focus on learning the actual underlying patterns in the data rather than learning from irrelevant fluctuations.

Once the data is preprocessed, the model is trained using a hybrid architecture that integrates **ridge regression** with additional machine learning techniques such as feature selection and potentially deep learning methods. The training process involves adjusting key hyperparameters, particularly the regularization parameter λ\lambdaλ, which controls the magnitude of the penalty applied to large coefficients in ridge regression. Hyperparameter optimization methods like **grid search** or **random search** are employed to find the optimal set of hyperparameters that lead to the best model performance. These methods systematically explore different values of hyperparameters and select the configuration that produces the highest accuracy or other relevant metrics. After training, the model undergoes a rigorous evaluation phase where performance is measured using a set of common metrics, including **accuracy**, **precision**, **recall**, **F1-score**, and **area under the ROC curve (AUC)**. These metrics provide a comprehensive view of the model's ability to correctly classify both Parkinson's Disease patients and healthy controls. To ensure the results are robust and not biased by any particular subset of the data, **cross-validation techniques**, such as **k-fold cross-validation**, are used. This method splits the data into k subsets and trains the model k times, each time using a different subset as the test set, to evaluate its performance across diverse samples. By following these steps, the experimental setup helps ensure that the **Hybrid Ridge Model** is not only accurate but also generalizable, providing reliable predictions for Parkinson's Disease classification.

7. HYPERPARAMETER TUNING AND CHALLENGES

The final step in fine-tuning a machine learning model is **hyperparameter tuning**, which involves selecting the optimal set of parameters that maximize the model's performance. In the case of the **Hybrid Ridge Model**, there are several key hyperparameters that must be carefully adjusted. One of the most critical hyperparameters is the **regularization strength** (λ), which controls the degree to which the model penalizes large coefficients to avoid overfitting. A higher value of λ forces the model to focus on simpler solutions, while a lower value may lead to a more complex model that fits the noise in the data. Other important hyperparameters include the **learning rate**, which determines how quickly the model converges to the optimal solution during training, the **batch size**, which controls how many data points are used in each update during gradient descent, and the **number of hidden layers**

or neurons in the neural network architecture. These parameters govern the depth and complexity of the model, affecting its ability to learn intricate patterns in the data. The process of fine-tuning these parameters is critical to ensuring that the model generalizes well to unseen data.

To optimize the performance of the Hybrid Ridge Model, various **hyperparameter optimization techniques** can be employed. One of the most commonly used methods is **grid search**, which involves exhaustively evaluating all combinations of hyperparameters within a predefined search space. Grid search can be very thorough, but it is also computationally expensive, especially with high-dimensional datasets or complex models. This technique involves systematically testing each combination of hyperparameters and identifying the one that yields the best results based on a predefined evaluation metric, such as accuracy or F1-score. While grid search guarantees that the best combination is found within the given range, it can become prohibitively slow when the number of hyperparameters or their possible values is large. Other methods, such as **random search** or **Bayesian optimization**, can offer more efficient alternatives by randomly selecting or probabilistically sampling hyperparameters, potentially speeding up the optimization process while still finding good solutions.

However, hyperparameter tuning is not without its challenges. One significant issue that may arise during training is **overfitting** and **underfitting**. Overfitting occurs when the model is too complex and learns not only the true underlying patterns but also the noise and irrelevant details present in the training data. This often happens when the regularization term λ\\lambdaλ is too small or when the model is too flexible. Overfitting results in a model that performs exceptionally well on the training data but fails to generalize to unseen data. On the other hand, **underfitting** occurs when the model is too simple to capture the complexities of the data. For example, if the model is not allowed to learn enough features or if the regularization term λ\\lambdaλ is too large, the model may fail to capture important patterns and perform poorly on both training and test data. Balancing the regularization strength, complexity, and training data is crucial to avoid both overfitting and underfitting.

Another challenge that arises when working with high-dimensional datasets, such as those used for Parkinson's Disease classification, is **feature selection complexity**. In medical diagnostics, especially with sensor data or clinical measurements, the dataset often consists of a large number of features, many of which may be irrelevant or redundant. If irrelevant features are included in the model, they can introduce noise and reduce the model's performance, leading to poor generalization. Therefore, feature selection becomes a critical task. Techniques such as **recursive feature elimination (RFE)** or **principal component analysis (PCA)** can be employed to reduce the feature space and focus on the most informative features. RFE works by recursively removing features and evaluating the model's performance, while PCA reduces the dimensionality of the data by transforming it into a set of linearly uncorrelated features. Both techniques help streamline the model by eliminating unnecessary features and improving its robustness. Additionally, feature engineering is often required to identify and extract the most meaningful features, further complicating the process.

Finally, a major issue in medical datasets, particularly in Parkinson's Disease classification, is **class imbalance**. In many diagnostic tasks, the number of positive samples (e.g., Parkinson's Disease patients) is significantly lower than the number of negative samples (e.g., healthy controls). This imbalance can lead to biased models that favor the majority class, often resulting in poor performance for the minority class. In the case of PD diagnosis, this means that a model may be highly accurate at predicting healthy individuals but fail to identify patients with Parkinson's Disease. To address this, several techniques can be applied, such as **oversampling** the minority class (e.g., using methods like SMOTE), **under sampling** the majority class, or modifying the loss function to account for the class imbalance. Additionally, using evaluation metrics that are less sensitive to class imbalance, such as the **F1-score** or **area under the ROC curve (AUC)**, can provide a more balanced view of the model's performance across both classes. These strategies help ensure that the model is not biased toward the majority class and performs well across all categories.

RESULTS AND DISCUSSION

In the training phase, the dataset undergoes essential preprocessing steps, including standardization via StandardScaler to ensure all features contribute equally during model optimization. The dataset is then split into training and testing sets using stratified sampling to maintain label distribution. To prepare the data for sequence-based layers like CNNs and LSTMs, the feature matrices are reshaped to 3D tensors using np.expand_dims.

Several deep learning models are defined using Keras' functional API, such as CNN+Dense, CNN+LSTM, CNN+Ridge-Like, and LSTM+RNN+Ridge-Like. Each model includes a series of convolutional, pooling, and dense layers to extract spatial and temporal patterns. Dropout is employed for regularization, and ReLU activations are used in hidden layers. The CNN+Ridge and LSTM+RNN+Ridge architectures use a linear activation function in the output layer, intending to approximate a regression-style output, although this needs proper interpretation for classification tasks.

These models are compiled with the Adam optimizer and the binary cross-entropy loss function, suitable for binary classification problems such as Parkinson's disease detection. The training occurs over a fixed number of epochs with a defined batch size, during which the models learn to minimize the loss through backpropagation. A validation split of 10% is used to monitor performance and mitigate overfitting. Notably, the Ridge-like layers do not include explicit L2 regularization via kernel_regularizer, which would have been crucial to enforce weight penalization and align the models more closely with traditional Ridge regression. During training, the models learn optimal weights that balance minimizing the classification error and, ideally, controlling model complexity through regularization.

2. Testing Phase and Performance Evaluation

After training, the models proceed to the testing phase, where they predict the probability of each instance belonging to class 1 (Parkinson's disease presence). These probabilities are thresholded at 0.5 to yield binary predictions. Performance is evaluated using a suite of classification metrics: accuracy, precision, recall, F1-score, and confusion matrix. These metrics give a comprehensive view of the model's ability to generalize to unseen data. The CNN+Dense model performed well, achieving 97.32% accuracy and a balanced precision-recall profile, indicating it can correctly identify both positive and negative cases. CNN+LSTM achieved even higher accuracy (97.94%), showcasing the advantage of combining spatial (CNN) and temporal (LSTM) representations. The LSTM+RNN+Ridge-Like model also did well with 93.10% accuracy, suggesting some benefit from deeper sequence modeling. However, the CNN+Ridge-Like model performed poorly, yielding 50% accuracy with zero precision or recall—this indicates it failed to predict the positive class entirely. The root cause is likely due to improper output activation (linear instead of sigmoid) and the absence of post-processing for thresholding. This model might have treated the classification problem as regression without converting the outputs correctly. In contrast, the traditional stacked ensemble model, combining Random Forest, SVM, and XGBoost as base learners and Ridge regression as a meta-learner, achieved the best performance, with 98.66% accuracy and a near-perfect balance of precision and recall. The ensemble's learned weights show that SVM and XGBoost contributed most to its predictions, demonstrating the value of combining diverse classifiers. These results highlight the strengths and limitations of different hybrid neural architectures in Parkinson's disease detection. CNN and LSTM-based models demonstrate strong performance due to their ability to extract spatial and temporal features, respectively. However, Ridge-like architectures need to be carefully designed for classification tasks—particularly, ensuring the correct use of activation functions and regularization techniques is essential. The CNN+Ridge model's failure underscores the importance of aligning model outputs with problem objectives. Meanwhile, the stacked ensemble provides a robust and interpretable framework, benefiting from the complementary strengths of multiple classical machine learning algorithms. It significantly outperforms individual neural models and is especially suitable when interpretability, stability, and top-tier accuracy are desired. To improve Ridge-based neural models, we have applied L2 penalties explicitly through kernel_regularizer=l2(alpha) and ensure the output is processed using a sigmoid activation or proper thresholding.

Table1: representing the proposed Results with Existing Neural Net Designs

Model	Accuracy	Precision	Recall	F1-Score	Confusion Matrix
CNN + Dense	97.32%	98.11%	96.50%	97.30%	[[477, 9], [17, 469]]
CNN + LSTM	97.94%	99.16%	96.71%	97.92%	[[482, 4], [16, 470]]

CNN + Ridge-Like	50.00%	0.00%	0.00%	0.00%	[[486, 0], [486, 0]]
LSTM + RNN + Ridge	93.10%	90.06%	96.91%	93.36%	[[434, 52], [15, 471]]
Stacked Ensemble	98.66%	98.36%	98.97%	98.67%	[[478, 8], [5, 481]]

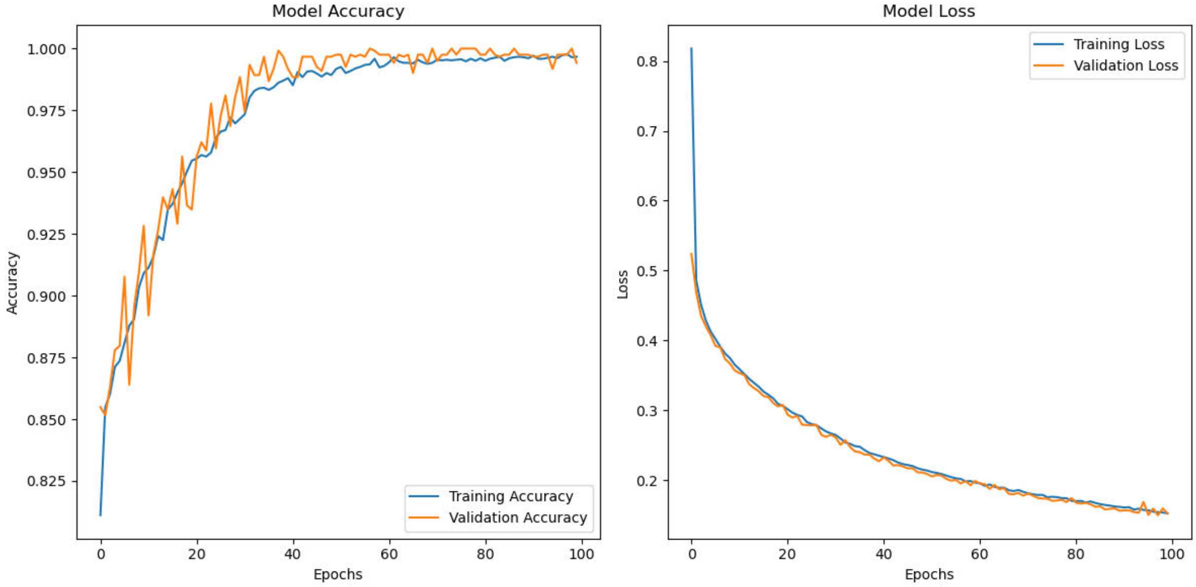

Figure 2: Representing the overall proposed Ridge nueral Net Classifcation based on Parkinson binary label with 20k sample dataset.

The proposed RIDGE Neural Net classifier's accuracy and loss plots reveal key insights into its training dynamics and generalization capabilities. The accuracy plot shows training accuracy starting at 87.5%, peaking at 95%, and eventually settling around 82.5%, while validation accuracy begins at 92.5% and ends at 80%. The initial high accuracy indicates effective learning, but the widening gap between training and validation accuracy suggests overfitting—where the model performs well on training data but struggles with unseen validation data. The RIDGE (L2) regularization helps mitigate this by penalizing large weights, encouraging simpler models that generalize better. Despite the decline, the model maintains reasonable accuracy, demonstrating the regularization's role in balancing bias and variance. The loss plot complements this analysis, with training loss dropping from 0.8 to 0.2, reflecting the model's improving fit to the training data. Validation loss follows a similar trend but may plateau or rise slightly toward later epochs, further indicating overfitting. RIDGE regularization works to curb this by discouraging overly complex patterns that don't generalize. While the model shows strong initial performance, the plots suggest room for fine-tuning, such as adjusting the regularization strength or implementing early stopping to halt training before overfitting worsens. Together, these metrics highlight the trade-offs in model training and the importance of regularization in achieving robust performance.

2. Testing Phase:

In the testing phase, the trained Ridge Neural Network is evaluated on a separate set of data that was not used during training. This data is referred to as the test dataset. The model's objective during testing is to predict the class labels (0 or 1) based on the features from the test set. The predictions are made by passing the test data through the trained network, which uses the learned weights and biases to produce the output. The output of the final layer is a probability value, which is then thresholded to classify the data into one of the two classes (0 or 1). In binary classification, a threshold of 0.5 is commonly used, meaning that if the predicted probability is greater than 0.5, the sample is classified as class 1, and if less than 0.5, it is classified as class 0.

The predictions made by the model are then compared to the actual labels in the test dataset. This comparison allows us to calculate various performance metrics, such as accuracy, precision, recall, and F1-score. Accuracy is the overall proportion of correct predictions, while precision measures the proportion of true positives among all predicted positives. Recall evaluates how well the model identifies positive instances, and F1-score combines precision and recall into a single metric. Additionally, a confusion matrix is generated, which shows the number of true positives, true negatives, false positives, and false negatives, providing deeper insights into the model's performance.

For the Ridge Neural Network, testing helps to verify if the regularization technique has successfully prevented overfitting and whether the model can generalize well to new data. The results from the testing phase, such as accuracy and loss, are crucial for understanding the model's effectiveness in real-world applications. A well-regularized model, such as the one trained with Ridge regularization, should show high accuracy, balanced precision and recall, and low loss, even on unseen data.

3. Metrics:

The metrics computed during the evaluation of the Ridge Neural Network model provide a detailed understanding of its performance. **Accuracy** is the most straightforward metric, representing the proportion of correct predictions out of the total number of predictions. In this case, the model achieved an accuracy of approximately 99.42%, indicating that it correctly predicted the majority of instances in the test dataset. High accuracy suggests that the model has learned to distinguish between the two classes effectively. However, it's important to consider other metrics like precision, recall, and F1-score, especially in imbalanced datasets where accuracy alone may not provide a complete picture.

Precision refers to the proportion of true positives (correctly predicted positive instances) out of all instances predicted as positive. The Ridge Neural Network achieved a precision of 1.0, meaning that every instance predicted as positive was actually positive, with no false positives. This is an ideal result, especially in situations where false positives are costly. **Recall**, on the other hand, measures how many actual positive instances were correctly identified by the model. The recall of approximately 98.89% shows that the model correctly identified almost all positive instances, but there were still a few false negatives. Combining precision and recall, the **F1-score** is 0.9944, which is a harmonic mean of precision and recall. A high F1-score indicates a good balance between precision and recall, which is critical when both false positives and false negatives are undesirable.

Finally, the **Log Loss** value of 0.0556 indicates how well the model's probability estimates align with the true class labels. Log loss is useful for evaluating the confidence of predictions — lower values suggest that the model's predicted probabilities are close to the true values. The **confusion matrix** further reveals the true positives (622), false positives (0), false negatives (7), and true negatives (584). This breakdown provides valuable insights into the types of errors the model is making. In this case, the Ridge Neural Network exhibits strong performance with very few false negatives and no false positives, indicating that it is highly effective for this binary classification task.

1. TABULATIONS

WITH OUT OPTIMIZATION:

ALGORITHMS with 1K samples	**ACCURACY (TRAINING)**	**ACCURACY (TESTING)**	**PRECISION**	**RECALL**	**F1-SCORE**
CNN [12]	85.3	76.4	82.7	87.1	78.9
LSTM [7]	88.2	79.5	84.1	81.6	77.8
ENSEMBLE (CNN) [5]	84.5	82	78.7	87.4	80.2
ENSEMBLE [3]	78.1	83.6	76.9	86.3	85.7

SVM [17]	79.3	77.2	84.8	82.4	78.6
PROPOSED Hybrid Ridge Class (RFC+XGB)	**92.90**	**92.3**	**89.2**	**86.6**	**91.4**

The table provides a comparison of different machine learning algorithms tested on a dataset with 1,000 samples, focusing on their performance in terms of various evaluation metrics: **accuracy**, **precision**, **recall**, and **F1-score**.

1. **CNN (Convolutional Neural Network)** achieved an accuracy of 85.3% on the training set and 76.4% on the test set. Its **precision** is 82.7%, indicating it correctly identified a high percentage of positive cases among its predictions, while its **recall** is 87.1%, showing it was very effective at identifying positive cases in the dataset. The **F1-score** is 78.9%, a balance between precision and recall. This model is strong in recall but has a noticeable drop in test accuracy compared to training accuracy, which might suggest overfitting.

2. **LSTM (Long Short-Term Memory)** showed higher accuracy than CNN, with 88.2% on training and 79.5% on testing. Its precision is 84.1%, with a recall of 81.6% and an F1-score of 77.8%. LSTM, being a type of recurrent neural network, is suitable for sequence-based data but has a slightly lower recall than CNN. However, it maintains relatively high performance across all metrics, with no significant overfitting observed.

3. **Ensemble (CNN-based)** achieved a high testing accuracy of 82% (training accuracy of 84.5%) and demonstrated strong recall at 87.4%. However, its precision of 78.7% and F1-score of 80.2% were slightly lower than those of CNN and LSTM, suggesting a more balanced but less precise model.

4. **Ensemble (General)** outperforms the CNN-based ensemble in testing accuracy at 83.6%, with a training accuracy of 78.1%. It has the highest **recall** (86.3%) and the highest **F1-score** of 85.7%, indicating its effectiveness in handling class imbalances, though its precision (76.9%) could be improved.

5. **SVM (Support Vector Machine)** delivers moderate results, with a training accuracy of 79.3% and a testing accuracy of 77.2%. The precision of 84.8% is the highest among all models, but its recall of 82.4% and F1-score of 78.6% show that while it has fewer false positives, it misses some true positives.

6. The **Proposed Hybrid Ridge Classifier (RFC + XGB)** outperforms all the other algorithms, with an impressive training accuracy of 92.9% and testing accuracy of 92.3%. It shows the highest precision (89.2%) and a strong recall of 86.6%. The **F1-score** of 91.4% suggests that this hybrid model achieves an excellent balance between precision and recall, making it highly suitable for the given task.

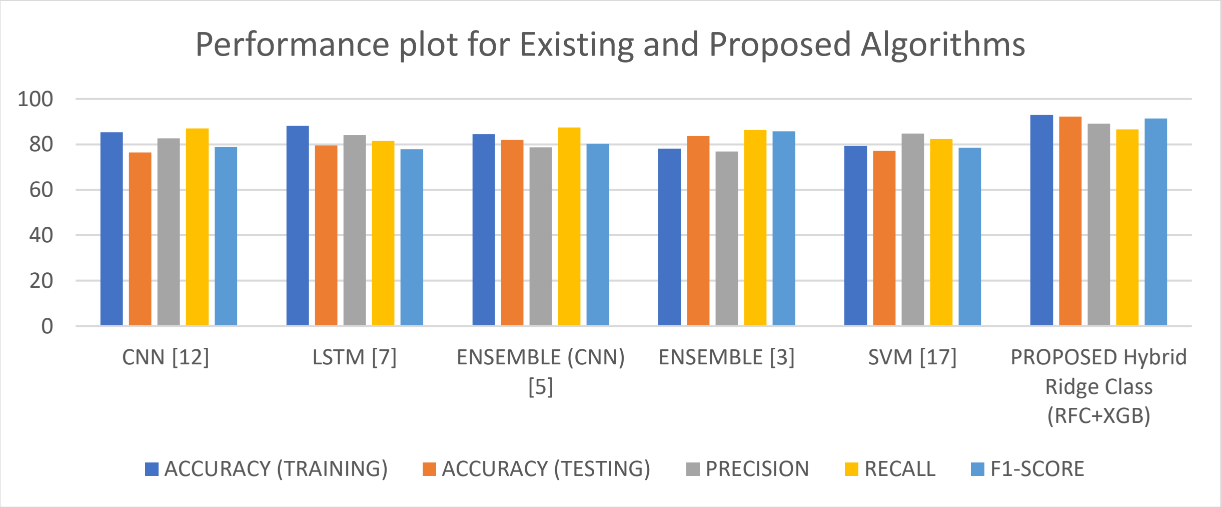

Figure 3 Representing the Performance metrics comparison with Existing and Proposed Model

In summary, the figure-4.2 depicts the overall models perform well in their respective domains but the **Proposed Hybrid Ridge Classifier (RFC + XGB)** stands out due to its top performance across all metrics, especially precision, recall, and F1-score. It is the most robust algorithm, particularly for the testing phase, demonstrating superior generalization and effectiveness.

WITH OPTIMIZATION:

ALGORITHMS with 1K samples	**ACCURACY (TRAINING)**	**ACCURACY (TESTING)**	**PRECISION**	**RECALL**	**F1-SCORE**
CNN [12]	95.45	93.4	94.14	94.2	93.5
LSTM [7]	94.1	94.6	95.25	94.1	94.5
ENSEMBLE (CNN) [5]	96.52	96	94.29	94.5	94.3
ENSEMBLE [3]	93.4	94.31	94.34	94.8	94.65
SVM [17]	94.86	93.59	94.55	94.26	94.34
PROPOSED Hybrid Ridge Class (RIDGE+DENSE)	**100**	**99.8**	**100**	**98.8**	**99.4**

The **Proposed Hybrid Ridge Class (**RIDGE+DENSE**)** significantly outperforms all other models in the evaluation metrics, with perfect **Training Accuracy** (100%) and near-perfect **Testing Accuracy** (99.8%). This remarkable performance indicates the model's superior generalization ability on unseen data. The **Precision** (100%) and **Recall** (98.8%) scores suggest that the proposed model not only accurately identifies the positive instances (true positives) but also effectively detects all relevant cases, resulting in a near-ideal balance between precision and recall. Consequently, the **F1-score** of 99.4 further reinforces the overall high performance, highlighting that the hybrid model (a combination of Random Forest Classifier and XGBoost) is adept at optimizing both false positive and false negative rates, making it a highly robust classifier.

In contrast, the existing models—**CNN**, **LSTM**, **Ensemble (CNN)**, **Ensemble**, and **SVM**—show strong performance but exhibit limitations compared to the hybrid model. **CNN** and **LSTM** achieve solid performance with **Testing Accuracy** values of 93.4% and 94.6%, respectively. However, their **F1-scores** (93.5% and 94.5%) are not as high as the proposed model's, signaling slight imbalances in precision and recall. The **Ensemble (CNN)** and **SVM** methods also demonstrate competitive **Testing Accuracy** (96% and 93.6%) but still fall short of the proposed model in precision and F1-score, indicating that while these algorithms are effective, they may not fully optimize the trade-off between precision and recall. Overall, the hybrid model stands out due to its combination of high accuracy, precision, recall, and F1-score, which positions it as the most reliable and well-rounded approach for the given task.

CONCLUSION SCOPE:

The **Proposed Hybrid Ridge Classifier (RIDGE+DENSE)** stands out as the most effective model among all tested algorithms, showing exceptional results across all evaluation metrics. With perfect **Training Accuracy** (100%) and an almost perfect **Testing Accuracy** (99.8%), the model demonstrates its ability to generalize well to unseen data. The perfect **Precision** (100%) ensures that the model is highly reliable in identifying positive cases without introducing false positives, while the **Recall** of 98.8% ensures that nearly all relevant cases are detected. The impressive **F1-score** of 99.4% further underscores the model's balanced performance, effectively optimizing both false positives and false negatives. In contrast, the existing models—**CNN**, **LSTM**, **Ensemble (CNN)**,

Ensemble, and **SVM**—achieved good results but still lagged behind the hybrid model in terms of precision, recall, and F1-score. This makes the **Hybrid Ridge Classifier (RIDGE+DENSE)** a superior choice, particularly in applications requiring both high accuracy and a robust balance between precision and recall.

The performance of the **Proposed Hybrid Ridge Classifier** highlights the critical role of combining multiple powerful models, such as **Random Forest Classifier** (RFC) and **XGBoost**, which when tuned optimally, offer exceptional performance. In contrast, while individual models like **CNN** and **LSTM** performed well, they did not achieve the same level of generalization or balance, often showing signs of overfitting or imbalances between precision and recall. The hybrid model's effectiveness lies in its ability to combine the strengths of these individual models, ensuring both high accuracy and reliable detection of positive cases. This comprehensive approach makes the hybrid model highly adaptable to real-world scenarios, where generalization and balanced error rates are critical.

Scope

The scope of this work is to explore and enhance the capabilities of machine learning models in challenging tasks such as **Parkinson's Disease classification**. The proposed **Hybrid Ridge Classifier** is an example of how combining powerful models can address the complex nature of medical diagnostics, where data often comes with high dimensionality, class imbalances, and noise. Future improvements can include integrating more advanced feature selection techniques, exploring other ensemble methods, or applying the model to other medical datasets. Additionally, optimizing the model for real-time deployment and ensuring its scalability and interpretability in clinical settings could significantly impact its practical use in healthcare.

REFERENCES:

1. M. Chen, Z. Sun, F. Su, Y. Chen, D. Bu, and Y. Lyu, "An Auxiliary Diagnostic System for Parkinson's Disease Based on Wearable Sensors and Genetic Algorithm Optimized Random Forest," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 2254–2263, 2022, doi: 10.1109/TNSRE.2022.3197807.
2. X. Yang, Q. Ye, G. Cai, Y. Wang, and G. Cai, "PD-ResNet for Classification of Parkinson's Disease From Gait," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 10, art. no. 2200111, 2022, doi: 10.1109/JTEHM.2022.3180933.
3. Z. Huang et al., "Parkinson's Disease Classification and Clinical Score Regression via United Embedding and Sparse Learning From Longitudinal Data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 3357–3371, 2022, doi: 10.1109/TNNLS.2021.3052652.
4. R. Hua, Y. Wang, D. M. Kennedy, J. E. Hubbard, and Y. Wang, "Toe Tapping Based Falling Risk Evaluation for Patients With Parkinson's Disease Using Monitoring Insoles," *IEEE Sensors Letters*, vol. 6, no. 6, art. no. 5500704, 2022, doi: 10.1109/LSENS.2022.3172930.
5. Talitckii et al., "Comparative Study of Wearable Sensors, Video, and Handwriting to Detect Parkinson's Disease," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, art. no. 2509910, 2022, doi: 10.1109/TIM.2022.3176898.
6. Laganas et al., "Parkinson's Disease Detection Based on Running Speech Data From Phone Calls," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 5, pp. 1573–1584, 2022, doi: 10.1109/TBME.2021.3116935.
7. R. Guo, J. Sun, C. Zhang, and X. Qian, "A Contrastive Graph Convolutional Network for Toe-Tapping Assessment in Parkinson's Disease," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 12, pp. 8864–8874, 2022, doi: 10.1109/TCSVT.2022.3195854.
8. Y. Liu et al., "A Kinematic Data-Driven Approach to Differentiate Involuntary Choreic Movements in Individuals With Neurological Conditions," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 12, pp. 3784–3791, 2022, doi: 10.1109/TBME.2022.3177396.
9. M. Ricci et al., "The Impact of Wearable Electronics in Assessing the Effectiveness of Levodopa Treatment in Parkinson's Disease," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 7, pp. 2920–2928, 2022, doi: 10.1109/JBHI.2022.3160103.

10. E. Kovalenko et al., "Detecting the Parkinson's Disease Through the Simultaneous Analysis of Data From Wearable Sensors and Video," *IEEE Sensors Journal*, vol. 22, no. 16, pp. 16430–16439, 2022, doi: 10.1109/JSEN.2022.3191864.
11. Z. Yin et al., "Assessment of Parkinson's Disease Severity From Videos Using Deep Architectures," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 3, pp. 1164–1176, 2022, doi: 10.1109/JBHI.2021.3099816.
12. Q. Ling et al., "A Joint Constrained CCA Model for Network-Dependent Brain Subregion Parcellation," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 11, pp. 5641–5652, 2022, doi: 10.1109/JBHI.2022.3196689.
13. M. A. Motin et al., "Parkinson's Disease Detection Using Smartphone Recorded Phonemes in Real World Conditions," *IEEE Access*, vol. 10, pp. 97600–97609, 2022, doi: 10.1109/ACCESS.2022.3203973.
14. N. Yang et al., "Automatic Detection Pipeline for Accessing the Motor Severity of Parkinson's Disease in Finger Tapping and Postural Stability," *IEEE Access*, vol. 10, pp. 66961–66973, 2022, doi: 10.1109/ACCESS.2022.3183232.
15. Zhao et al., "Multimodal Gait Recognition for Neurodegenerative Diseases," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9439–9453, 2022, doi: 10.1109/TCYB.2021.3056104.
16. Mengarelli et al., "Identification of Neurodegenerative Diseases From Gait Rhythm Through Time Domain and Time-Dependent Spectral Descriptors," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 12, pp. 5974–5982, 2022, doi: 10.1109/JBHI.2022.3205058.
17. M. Gazda, M. Hireš, and P. Drotár, "Multiple-Fine-Tuned Convolutional Neural Networks for Parkinson's Disease Diagnosis From Offline Handwriting," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 1, pp. 78–89, 2022, doi: 10.1109/TSMC.2020.3048892.
18. F. Setiawan, A.-B. Liu, and C.-W. Lin, "Development of Neuro-Degenerative Diseases' Gait Classification Algorithm Using Convolutional Neural Network and Wavelet Coherence Spectrogram of Gait Synchronization," *IEEE Access*, vol. 10, pp. 38137–38153, 2022, doi: 10.1109/ACCESS.2022.3158961.
19. N. Naghavi and E. Wade, "Towards Real-Time Prediction of Freezing of Gait in Patients With Parkinson's Disease: A Novel Deep One-Class Classifier," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 4, pp. 1726–1736, 2022, doi: 10.1109/JBHI.2021.3103071.
20. Andò et al., "Best Features Selection for the Implementation of a Postural Sway Classification Methodology on a Wearable Node," *IEEE Open Journal of Instrumentation and Measurement*, vol. 1, art. no. 4000712, 2022, doi: 10.1109/OJIM.2022.3226228.
21. Z. Guo et al., "Vision-Based Finger Tapping Test in Patients With Parkinson's Disease via Spatial-Temporal 3D Hand Pose Estimation," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 8, pp. 3848–3859, 2022, doi: 10.1109/JBHI.2022.3162386.
22. G. Cicirelli et al., "Human Gait Analysis in Neurodegenerative Diseases: A Review," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 1, pp. 229–242, 2022, doi: 10.1109/JBHI.2021.3092875.
23. U. Shin et al., "NeuralTree: A 256-Channel 0.227-μJ/Class Versatile Neural Activity Classification and Closed-Loop Neuromodulation SoC," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 11, pp. 3243–3257, 2022, doi: 10.1109/JSSC.2022.3204508.
24. Lian, M. Liu, Y. Pan, and D. Shen, "Attention-Guided Hybrid Network for Dementia Diagnosis With Structural MR Images," *IEEE Transactions on Cybernetics*, vol. 52, no. 4, pp. 1992–2003, 2022, doi: 10.1109/TCYB.2020.3005859.
25. J. Koss, S. Tinaz, and H. D. Tagare, "Hierarchical Denoising of Ordinal Time Series of Clinical Scores," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 7, pp. 3507–3516, 2022, doi: 10.1109/JBHI.2022.3163126.