Multi-Modal Analysis of Parkinson Disease data Using Advanced Deep Learning Techniques

Dr. P. Swathy Priyadharsini[1], Dr. Sivaranjani S[2], Dr. N. Saranya[3], Vedasree R[4]

[1] *Assistant Professor (Sl.G)Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India. Email: swa.pspd@gmail.com*
[2] *Assistant Professor, Department of CSE, PSG college of Technology, Coimbatore, India. Email: ssr.cse@psgtech.ac.in*
[3] *Assistant Profssor (Sl. G), Department of Information Technology, Sri Ramakrishna Engineering College, Coimbatore, India. Email: saranya.pravin@srec.ac.in*
[4] *Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Sathyamangalam, India. Email: vedasree.cs21@bitsathy.ac.in*

KEYWORDS
Parkinson Disease, Multi-modal data, Deep Learning, Neural Networks, Long Short-Term Memory, Convolutional Neural Networks

ABSTRACT
Early diagnosis is essential for effective treatment of Parkinson's disease (PD), a progressive neurological disorder that affects movement and cognitive functions. This study presents a multi-modal analysis for PD classification using deep learning algorithms applied to clinical, audio, and handwriting image data. Recurrent neural networks (RNNs), including Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bidirectional LSTM, are employed for clinical and audio data analysis, while convolutional neural networks (CNNs) are utilized for handwriting image classification. The results demonstrate varying model performance across different data modalities. Among the clinical data models, GRU achieved the highest accuracy of 82.42%, indicating its effectiveness in capturing sequential dependencies in medical records. For audio-based classification, RNN outperformed all other models with 94.87% accuracy, while LSTM and GRU showed comparable performance, each reaching 92.31% accuracy. In the image modality, CNN without Batch Normalization attained 82.93% accuracy, whereas Batch Normalization improved performance to 85.37%, highlighting its role in stabilizing training and enhancing feature extraction. These findings emphasize the importance of modality-specific deep learning models and their potential to enhance early and accurate PD detection. The study emphasise the significance of multi-modal approaches in medical diagnostics, paving the way for improved, non-invasive, AI-driven assessments.

1. Introduction:

Parkinson's disease (PD) is a complex neurological disorder affecting millions globally, significantly impairing motor and cognitive functions. Early detection and accurate diagnosis are essential for effective management; as timely intervention can enhance the quality of life for affected individuals. Traditional diagnostic methods rely on clinical assessments and biomarkers, often missing subtle early signs and delaying treatment. PD is a slowly progressing neurodegenerative brain disease (Senturk, 2020) [1]. The word "neurodegenerative" describes a disease that causes brain cells to die. The lower brain stem, olfactory tracts, and enteric nervous system are where Parkinson's disease (PD) first manifests (Ko et al., 2021)[2]. PD affects both the brain's outer layer and the substantia nigra, moving from the regions of the initial symptoms to the brain's upper regions. The area of the brain responsible for motor control and cognitive function is affected by damage to the upper regions of the brain. It is thought that the sickness begins several years prior to the onset of motor symptoms, such as constipation, tremor, slowness of movement, loss or diminution of smell. Moreover, vocal issues affect 90% of those with Parkinson's disease (Sakar &Kursun, 2010) [3]. With time, the illness's symptoms worsen, and people who are in more advanced stages may develop dementia and hallucinations (Arora et al., 2021) [4]. Consequently, in an effort to slow the progression of the illness, scientists are currently looking for ways to recognize these non-motor signs as soon as feasible.

Machine Learning (ML) is being utilized more and more to identify medical disorders due to its high accuracy and ease of implementation (Abuhmed et al., 2021; Rashidy et al., 2021) [5][6]. In instance, PD has been treated with ML. Wan et al., (2019), for instance, focuses on studies carried out after PD diagnosis [7]. The authors performed surgeries on PD patients. In that study, the actual region to be operated on during PD brain surgery was identified using an ML-based method. Recent advancements in ML and artificial intelligence (AI) have introduced innovative diagnostic techniques that utilize multiple data. In this research, multi-modal data of PD is considered for the analysis, including voice

recordings, handwriting samples, and clinical evaluations. This approach provides a comprehensive understanding of PD by integrating insights from various data types. Deep learning architectures, such as recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and convolutional neural networks (CNNs), have shown exceptional effectiveness in recognizing patterns in complex datasets. This research aims to develop a robust classification model that accurately differentiates between individuals with Parkinson's disease and healthy individuals, ultimately facilitating timely interventions that can significantly improve patient outcomes.

2. Literature Survey

Smith et al. (2023) investigated the use of voice recordings, handwriting samples, and clinical data in a multimodal deep learning strategy for PD identification [8]. Their neural network model achieved an impressive accuracy of 92.4%, effectively differentiating between Parkinson's patients and healthy individuals. This study highlights the importance of utilizing multimodal information to capture the complex symptoms of PD, such as motor dysfunction and cognitive decline, by integrating temporal data from voice recordings with spatial data from handwriting samples. Similarly, Kwon and Kim (2023) explored tracking the progression of Parkinson's disease using clinical data and wearable sensors [9]. Their deep learning model, which combined sensor data (gait and movement patterns) with clinical assessments, attained a classification accuracy of 90.1%. This research underscores the potential of continuous, real-time monitoring through wearable sensors to enhance our understanding of PD progression, offering a dynamic view of disease evolution over time.

Patel and Malhotra (2023) furthered this exploration by integrating voice recordings and gait analysis for PD diagnosis [10]. Their approach, which achieved an accuracy of 89.5%, combined vocal characteristics from speech with motor data from gait. By merging motor and non-motor data, this study provided a more comprehensive assessment of the impact of PD on both speech and movement. In a different vein, Ranjan and Gupta (2022) developed a machine learning model for early PD diagnosis by merging clinical data with neuroimaging data, such as MRI and PET scans [11]. Utilizing a 3D CNN, they reported an accuracy of 88.7%. This work emphasizes the role of structural and functional imaging biomarkers in early detection, especially for identifying neurodegeneration patterns associated with PD.

Kumar and Ray (2022) proposed a multimodal learning framework that combined voice data, handwriting images, and clinical information, resulting in a high detection accuracy of 95% [12]. The study demonstrated how each data modality provided complementary insights: handwriting captured fine motor control, voice data represented speech motor symptoms, and clinical records offered contextual information about the patient’s overall health. This fusion of modalities significantly improved the model's diagnostic capabilities. Xu and Chen (2022) examined the potential of merging voice recordings with electroencephalography (EEG) to diagnose PD [13] Their hybrid CNN-RNN model achieved an accuracy of 87.9%, capturing both the motor symptoms affecting speech and the cognitive aspects of the disease. This innovative approach suggests that integrating voice data with brain activity can provide a more nuanced understanding of PD symptoms.

Zhou and Li (2021) also focused on hybrid models, proposing one that combines voice recordings with handwriting data, achieving a detection accuracy of 94% [14]. This model effectively illustrates how PD affects various aspects of movement, such as hand coordination and voice articulation, demonstrating that combining multiple motor dysfunctions can enhance diagnostic performance. Lastly, Wang and Zhang (2021) developed a multimodal fusion model for PD diagnosis by integrating data from structural MRI, functional MRI (fMRI), and diffusion tensor imaging (DTI) [15]. Their study reported an accuracy of 93.2%, with each imaging modality contributing unique insights into brain structure and connectivity affected by PD.

3. Materials and Methods

3.1. Dataset Description

The datasets utilized in this study were sourced from Kaggle, a prominent platform for data science and machine learning resources. The primary datasets included:

Voice Recordings Dataset: This dataset comprises 195 audio samples of individuals diagnosed with Parkinson's disease and healthy controls. The 24 attributes of the dataset are the features extracted from these recordings that are the various acoustic parameters essential for analyzing vocal characteristics related to the disease.

Handwriting Samples Dataset: This dataset contains images of handwriting samples from both Parkinson's patients and healthy individuals with atotal of 204 images categorized into four groups: wave_healthy, wave_parkinson, spiral_healthy, and spiral_parkinson with 51 samples each. Key features from these images help assess fine motor control, which is significantly impacted by Parkinson's disease.

Clinical Data: This dataset includes 2105 samples with 35 attributes related to the demographic information, and health-related metrics of participants helping in a comprehensive array of clinical assessments. It provides context and background for understanding the overall health status of individuals in the study.

3.2. Proposed Methodology

Figure 1 shows the proposed model framework of this research work. This research focus on applying deep learning techniques to multi-modal data of PD, which involves combining and processing different types of data, such as images, audio, and clinical information. Multi-modal data poses unique challenges due to the varied structures and formats of the datasets involved, requiring sophisticated fusion methods to effectively integrate these diverse sources. Initial step of the work is to collect the multiple datasets best representing the Parkinson disease. Different modes such as audio signals data of the Parkinson affected people, handwriting samples of the Parkinson patients are collected as images and also real time clinical dataset of PD are collected from different sources.

3.3. Data Pre-processing

Dataset preprocessing is the foundation of machine learning pipelines, directly impacting model performance and interpretability. An extensive cleaning procedure is required since raw data frequently contains noise, missing numbers, and errors. Dataset cleaning involves detecting and correcting errors, handling missing values, and removing outliers. It is a critical step, as poor-quality data can lead to biased or incorrect model predictions. Strategies such as imputation (mean, median, or mode), deletion (removing rows/columns with missing data), or using machine learning models are utilized to predict missing values. Techniques such as Z-score, IQR, and visualization-based methods are applied in order to identify and handle extreme values that may distort model results. Smoothing techniques, such as moving averages or binning, help eliminate random noise in the data. Normalization aims to scale numeric data into a common range without distorting differences in the data. Z- Score normalization technique has been employed to standardize the data by centering it around a mean of 0 with a standard deviation of 1. Then, Feature extraction is applied that reduces the dimensionality of the dataset by transforming raw data into informative features. This step is crucial for improving model efficiency and avoiding overfitting. Principal Component Analysis (PCA) is employed which is a dimensionality reduction technique that transforms features into a set of uncorrelated variables called principal components, retaining most of the variance. Exploratory data analysis is the step where key insights about the data are gathered. EDA helps understand the relationships between variables, detect patterns, and uncover hidden trends, guiding further modeling efforts.

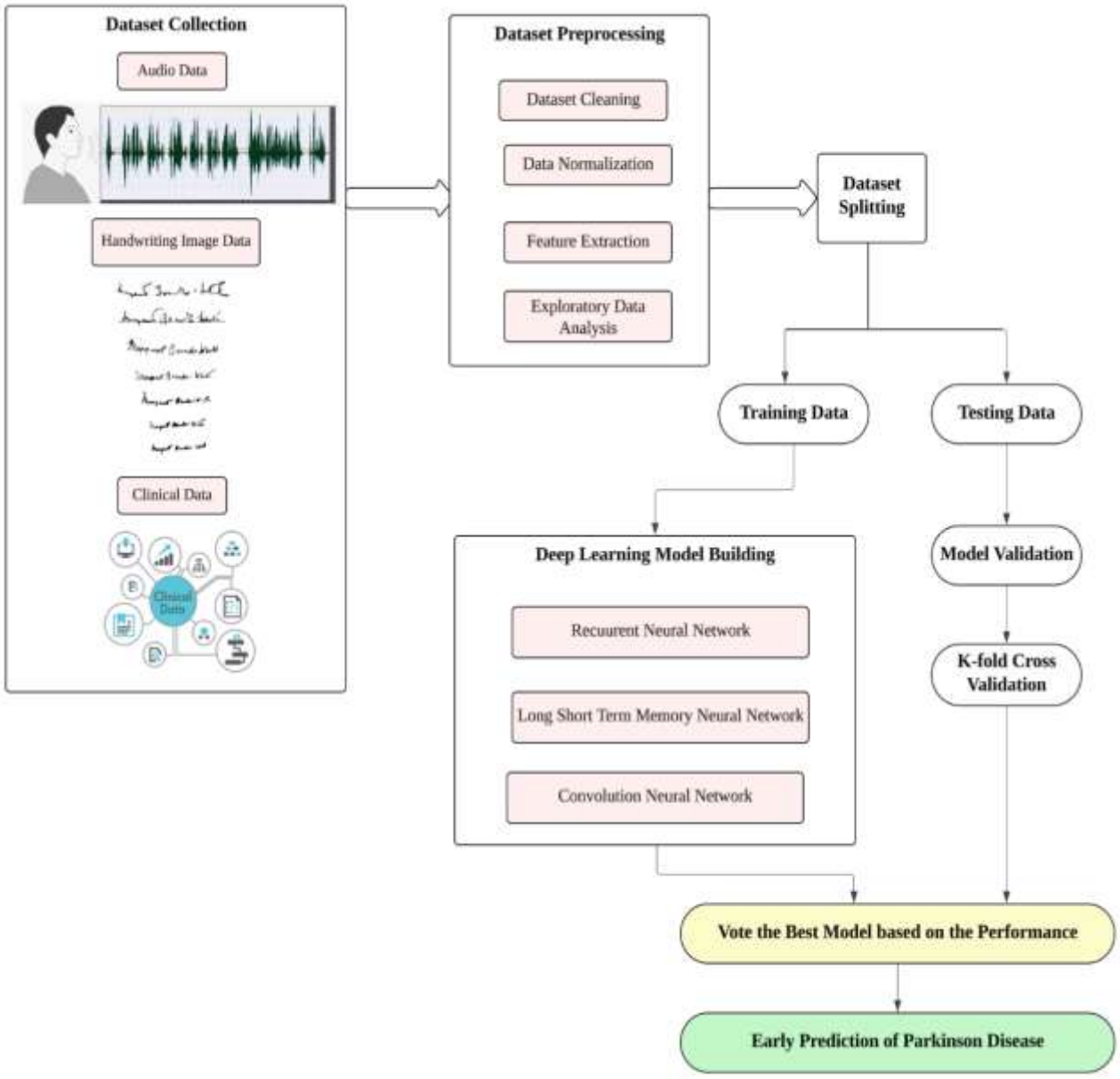

Figure 1: Proposed Model Framework

For the purpose of constructing models and eliminating bias in datasets, researchers as well as practitioners frequently employ the K-fold cross-validation technique. With a k value of 10, the K-fold cross-validation technique has been applied. Ten equal-sized segments were created by randomly dividing the full dataset. Out of the Ten partitions, one partition was kept as the model's validation (testing set), and the other nine partitions are utilized as training data one for each model. With precisely one usage of each of the ten partitions as the validation data, the entire procedure has been repeated ten times. The summing function combines the outcomes of each iteration. To match the performance of the training and testing datasets, the issue of over-fitting and under-fitting has been minimized in the dataset. This method has the advantage of eliminating data bias, which is necessary for creating DL models that produce accurate results.

The research includes the utilization of multiple deep learning architectures to enhance diagnostic accuracy by leveraging their unique strengths in handling different types of data. The models employed are Recurrent Neural Networks (RNNs) were applied to model temporal dependencies in voice recordings, allowing the system to capture patterns over time. This architecture excels at learning sequential relationships, making it effective for analyzing time-series data such as audio. Long Short-

Term Memory (LSTM) Networks is used as a specialized type of RNN, LSTMs were used to capture long-range dependencies in sequential data. LSTMs are particularly effective in addressing the vanishing gradient problem, allowing the model to retain information over longer sequences. This capability was critical for processing the dynamic features of voice and handwriting data, where temporal patterns influence the overall diagnosis. Convolutional Neural Networks (CNNs) were utilized for analyzing handwriting images due to their strong performance in image recognition tasks. CNNs are designed to recognize spatial hierarchies in images, making them ideal for extracting features that relate to motor function from handwriting samples.

To determine the most effective model for diagnostic purposes, a voting classifier was implemented. This ensemble method combined the predictions from each model—RNNs, LSTMs, and CNNs—based on their performance. Soft Voting technique is applied in which soft voting was used, where each model provided a probability distribution over the possible labels. The final prediction was based on the average probabilities, offering a more nuanced aggregation of the models' predictions.

By comparing the performance metrics such as accuracy, precision, recall, and F1-score—across individual models and the ensemble, the voting classifier selected the model or combination that demonstrated the highest diagnostic accuracy. This approach ensured a balanced decision-making process, improving the overall robustness of the system by integrating the complementary strengths of the deep learning architectures.

3.4. Deep Learning Algorithms

Many sophisticated deep learning models, each tailored to the unique modalities of the dataset, have been used in this investigation. In particular, we used Convolutional Neural Networks (CNN) for image data and Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bidirectional LSTM models for clinical and audio data.

3.4.1. Recurrent Neural Networks (RNNs)

The RNN is utilized to handle sequential data, particularly suited for our audio and clinical datasets where time-based dependencies exist. The key feature of RNN is its ability to retain memory from previous steps in the sequence. Mathematically, the hidden state ht at time step t is computed as:

$$h_t = \sigma(W_h \cdot x_t + U_h \cdot h_{t-1} + b_h) \quad (1)$$

where xt represents the input at time t, Wh are the weight matrices, bh is the bias term, and σ is the activation function, typically the tanh function.

The RNN model was designed with two layers of SimpleRNN units, with 128 and 64 units respectively, followed by dropout layers to reduce overfitting. The ReLU activation function was employed to introduce non-linearity, improving the model's ability to learn complex patterns in the data. Additionally, a dropout rate of 20% was applied after each RNN layer to prevent overfitting, especially given the relatively small dataset. A Dense layer with a sigmoid activation function was used for binary classification, outputting probabilities for the two classes (healthy vs. Parkinson's).

To further enhance performance, the model was compiled with the Adam optimizer, using a custom learning rate of 0.0005 to ensure smoother convergence. The binary cross-entropy loss function was selected for this classification problem, along with the accuracy metric to evaluate performance. The model was trained for 200 epochs with a batch size of 16, using early stopping to halt training if the validation loss stopped improving for 10 consecutive epochs. Early stopping helps prevent overfitting, ensuring that the model does not over-train on the training data.

3.4.2. Long Short-Term Memory (LSTM) Networks

LSTM is a variant of RNN designed to overcome the vanishing gradient problem, making it particularly useful for long-term dependencies. LSTM incorporates gates to control the flow of

information:

Forget gate: $f_t = \sigma\,(W_f \cdot [h_{t-1}, x_t] + b_f)$ (2)

Input gate: $i_t = \sigma\,(W_i \cdot [h_{t-1}, x_t] + b_i)$ (3)

Cell state update: $\hat{C}_t = tanh\,(W_c \cdot [h_{t-1}, x_t] + b_c)$ (4)

Final cell state: $C_t = f * C_{t-1} + i_t * \hat{C}_t$ (5)

Output gate: $o_t = \sigma\,(W_o \cdot [h_{t-1}, x_t] + b_o)$ (6)

The hidden state is updated by: $h_t = o_t * tanh\,(C_t)$ (7)

LSTM's gating mechanism allows it to selectively retain important information over longer sequences, making it effective for both clinical and audio data.

The LSTM model was constructed with two LSTM layers, with 128 units in the first layer and 64 units in the second. A dropout rate of 20% was applied after each LSTM layer to mitigate overfitting. After the second LSTM layer, a dense layer with 32 units and a ReLU activation function was included, followed by another dropout layer. Batch normalization was added after the first LSTM layer to stabilize the learning process and accelerate convergence. The model's output layer used a sigmoid activation function for binary classification, predicting the likelihood of Parkinson's disease. The model was compiled using the Adam optimizer with a learning rate of 0.0005 and binary cross-entropy loss. Early stopping was employed to halt training when the validation loss no longer improved.

3.4.3. Graded Recurrent Units(GRU)

GRU is a simplified version of LSTM with fewer parameters, which makes it computationally efficient. The update and reset gates control the flow of information.

Update gate: $Z_t = \sigma\,(W_Z \cdot [h_{t-1}, x_t] + b_Z)$ (8)

Reset gate: $r_t = \sigma\,(W_r \cdot [h_{t-1}, x_t] + b_r)$ (9)

The current hidden state ht is computed as:

$$h_t = (1 - z_t) * h_{t-1} + z_t * tanh(W_h \cdot [r_t * h_{t-1}, x_t] + b_h \quad (10)$$

The GRU model was designed with two GRU layers, with the first layer containing 128 units and the second layer containing 64 units. A dropout rate of 20% was applied after each GRU layer to combat overfitting. Batch normalization was included after the first GRU layer to enhance training stability and convergence speed. Following the GRU layers, a dense layer with 32 units and a ReLU activation function was added, along with another dropout layer.

The output layer used a sigmoid activation function for binary classification, predicting the likelihood of Parkinson's disease. The model was compiled with the Adam optimizer, utilizing a learning rate of 0.0005 and binary cross-entropy loss. An early stopping callback was also implemented to halt training when validation loss ceased to improve.

3.4.4. Bidirectional LSTM (Bi-LSTM)

The Bi-LSTM processes the sequence data in both forward and backward directions, improving the model's ability to capture contextual information from the entire sequence. The forward hidden state F(ht) and backward hidden state B(ht) are computed separately:

$$F(h_t) = LSTM(x_t, h_{t-1}) \quad (11)$$

$$B(h_t) = LSTM\,(x_t, h_{t-1}) \quad (12)$$

The final output is the concatenation of the two hidden states:

$$h_t = [F(h_t), B(h_t)] \quad (13)$$

Bi-LSTM is particularly useful when the sequence's context from both past and future states matters, which improves the classification performance on the clinical and audio data.

The Bi-LSTM model was constructed with two layers of Bidirectional LSTMs, enhancing its ability to capture temporal dependencies in both directions of the input sequence. A dropout layer with a rate of 20% was added to mitigate overfitting. Batch normalization was applied after the first Bidirectional LSTM layer to promote training stability. The second Bidirectional LSTM layer contains 64 units, also followed by a dropout layer. To further refine the output, a dense layer with 32 units and a ReLU activation function was added, along with another dropout layer.

The final output layer uses a sigmoid activation function for binary classification, estimating the likelihood of Parkinson's disease. The model was compiled using the Adam optimizer with a learning rate of 0.0005 and binary cross-entropy as the loss function. An early stopping callback was included to prevent overfitting by halting training when validation loss does not improve.

3.4.5. Convolutional Neural Networks (CNNs)

CNN was used for image data classification of wave and spiral drawings. The model consists of multiple convolutional layers followed by pooling layers to extract spatial features. Mathematically, a convolutional layer applies a filter W to the input image x:

$$(W * x)(i,j) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} W(m,n).x(i+m,j+n) \quad (14)$$

where W is the convolution filter of size M x N, and (i, j) denotes the location in the image. Pooling layer further reduces the dimensionality of the features, which are then passed through fully connected layers for final classification. In this study, two Convolutional Neural Network (CNN) architectures were employed for binary image classification. Both models were designed to handle color images with dimensions (img_width, img_height, 3) and consisted of convolutional layers for feature extraction, followed by fully connected layers for classification. The objective of both models was to extract hierarchical features from the images and perform binary classification using a sigmoid activation function in the final layer.

The first CNN model is composed of three convolutional blocks. The first block utilizes 32 filters with a 3x3 kernel, followed by a ReLU activation function to capture low-level features from the input images. A 2x2 max-pooling layer is used to reduce the spatial dimensions of the feature maps, followed by two more convolutional blocks with 64 and 128 filters respectively, each with 3x3 kernels and ReLU activation. After each convolutional block, max-pooling is applied to down-sample the feature maps. The resulting feature maps are flattened into a one-dimensional vector before passing through fully connected layers. The first dense layer contains 128 neurons with ReLU activation, followed by a dropout layer with a 50% dropout rate to prevent overfitting. Another dense layer with 64 neurons follows, and finally, a single neuron with a sigmoid activation function is used for binary classification. This model is compiled with the Adam optimizer, a learning rate of 0.001, binary cross-entropy as the loss function, and accuracy as the evaluation metric.

The second CNN model builds upon the architecture of the first model but introduces several modifications for improved regularization and convergence. It includes four convolutional blocks, with the number of filters progressively increasing from 32 in the first block to 256 in the fourth. Each convolutional layer is followed by batch normalization, which helps stabilize and accelerate training by normalizing the activations within each mini-batch. This is followed by a 2x2 max-pooling layer and a dropout layer with a rate of 0.25 to prevent overfitting. In this model, dropout is applied after every max-pooling layer, further reducing the risk of overfitting. After the convolutional layers, the feature maps are flattened, and two fully connected layers are introduced. The first dense layer contains 256 neurons with ReLU activation, followed by a dropout layer with a 50% dropout rate for stronger regularization. A second dense layer with 128 neurons and ReLU activation follows, again with a dropout layer at a 50% rate. The final output layer consists of a single neuron with a sigmoid activation

function for binary classification. This model is compiled using the Adam optimizer with a reduced learning rate of 0.0001, which aids in better convergence during training, especially for deeper networks. The loss function remains binary cross-entropy, and the model evaluates its performance using accuracy.

Both architectures were designed with the goal of effectively capturing image features while employing techniques like dropout and batch normalization to prevent overfitting and enhance generalization. The first model is simpler, while the second introduces more layers and regularization techniques, making it potentially more robust for complex datasets.

Through the use of these algorithms, the study aimed to develop a robust classification model capable of accurately distinguishing between individuals with Parkinson's disease and healthy controls.

4. Results and Discussion

4.1. Result Analysis of Models used with Clinical Data

Figure 2 represents the results obtained from the RNN model. The RNN model achieved an accuracy of 81.24% on the clinical dataset, demonstrating its effectiveness in classifying Parkinson's disease. The confusion matrix revealed that the model correctly identified 226 Parkinson's patients and 116 healthy individuals, but misclassified 34 healthy patients and 45 Parkinson's patients. Precision for class 1 (Parkinson's) was 0.869, with a recall of 0.834, indicating strong performance in minimizing false positives. The F1-scores of 0.746 and 0.851 for healthy and Parkinson's classes, respectively, reflect a good balance between precision and recall.

```
====== CONFUSION MATRIX OF RNN MODEL ======
 [[116  34]
 [ 45 226]]
========== ACCURACY OF RNN MODEL ==========
                0.8123515248298645
====== CLASSIFICATION REPORT OF RNN MODEL ======
                     0           1  accuracy   macro avg  weighted avg
precision     0.720497    0.869231  0.812352    0.794864      0.816238
recall        0.773333    0.833948  0.812352    0.803641      0.812352
f1-score      0.745981    0.851224  0.812352    0.798602      0.813726
support     150.000000  271.000000  0.812352  421.000000    421.000000
```

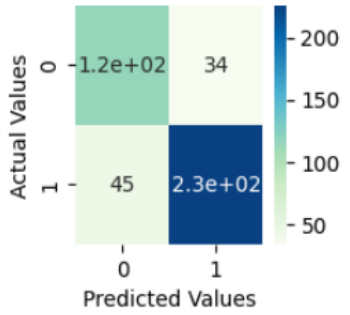

Figure 2. Result of RNN model with Clinical Data

The LSTM model demonstrated solid performance, achieving an overall accuracy of 80.29% as depicted in Figure 3. The precision for class 1 (Parkinson's) was 0.862, indicating the model's effectiveness in minimizing false positives. However, the precision for class 0 (healthy) was slightly lower at 0.708, revealing challenges in distinguishing some healthy individuals from those with Parkinson's. The F1-scores of 0.733 and 0.844 for healthy and Parkinson's classes, respectively, reflect a balance between precision and recall, although the model showed a slightly lower performance in detecting healthy cases. The confusion matrix highlights that while the model successfully identified a majority of Parkinson's patients, it misclassified 47 cases, suggesting a need for further refinement.

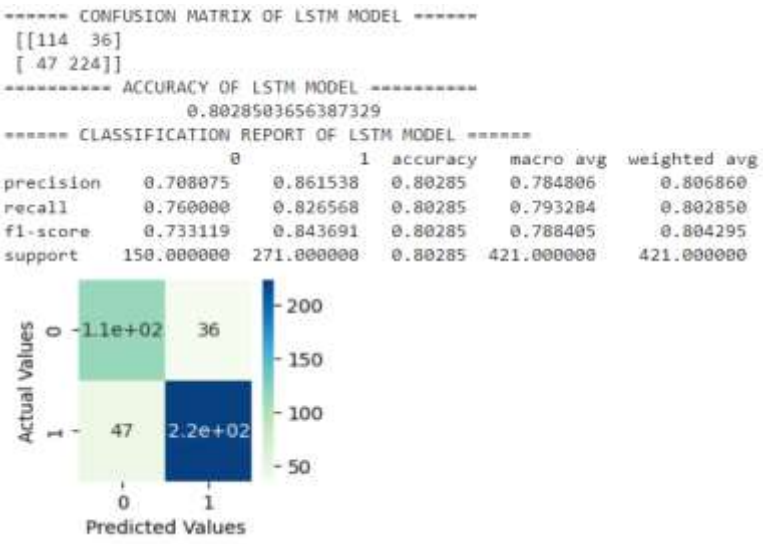

Figure 3. Result of LSTM with Clinical Data

Figure 4 shows the results of GRU model that demonstrated robust performance with an accuracy of 82.42%, outperforming both the RNN and LSTM models. The precision for class 1 (Parkinson's) was 0.886, indicating a strong ability to minimize false positives. Meanwhile, the precision for class 0 (healthy) was 0.729, suggesting that some healthy cases were misclassified as Parkinson's. The F1-scores of 0.766 for healthy individuals and 0.859 for Parkinson's patients illustrate a well-balanced performance in terms of precision and recall. Notably, the model misclassified 45 Parkinson's cases, indicating potential areas for improvement, possibly through further feature exploration or hyperparameter tuning. The Bidirectional LSTM model was evaluated on the clinical data for Parkinson's disease classification, yielding an accuracy of approximately 80.05% as the Figure 5 shows. The confusion matrix revealed that the model correctly classified 107 instances of healthy patients and 230 instances of Parkinson's disease patients, while misclassifying 43 healthy instances and 41 Parkinson's instances. The classification report highlighted the precision for healthy patients (label 0) at 72.30% and for Parkinson's patients (label 1) at 84.25%. The recall values were 71.33% and 84.87%, respectively, indicating that the model is effective in identifying Parkinson's patients but slightly less so for healthy patients. The F1-scores were 71.81% for healthy patients and 84.56% for Parkinson's patients, demonstrating a balance between precision and recall.

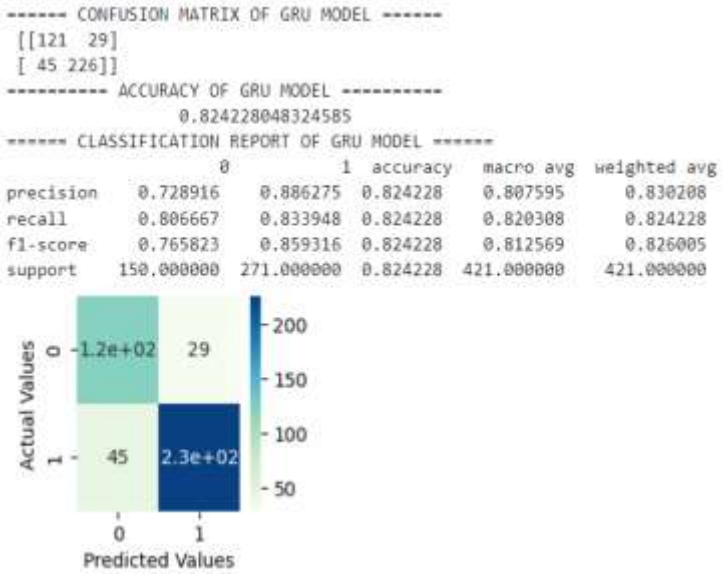

Figure 4. Results of GRU with Clinical Data

```
====== CONFUSION MATRIX OF BIDIRECTIONAL LSTM MODEL ======
 [[107  43]
 [ 41 230]]
========== ACCURACY OF BIDIRECTIONAL LSTM MODEL ==========
                0.8004750609397888
====== CLASSIFICATION REPORT OF BIDIRECTIONAL LSTM MODEL ======
                     0           1  accuracy   macro avg  weighted avg
precision     0.722973    0.842491  0.800475    0.782732      0.799907
recall        0.713333    0.848708  0.800475    0.781021      0.800475
f1-score      0.718121    0.845588  0.800475    0.781855      0.800172
support     150.000000  271.000000  0.800475  421.000000    421.000000
```

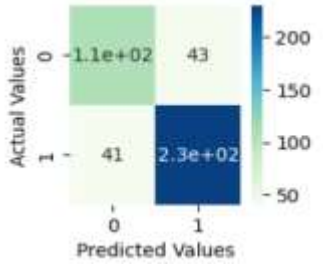

Figure 5. Results of Bidirectional LSTM with Clinical Data

4.2. Result Analysis of Models with Audio Data

The RNN model achieved an impressive accuracy of approximately 94.87% on the audio dataset as depicted by Figure 6. The confusion matrix indicated that the model correctly classified 5 out of 7 negative samples and all 32 positive samples. The precision for class 0 was perfect at 1.00, while class 1 had a precision of 0.94. The recall for class 0 was 0.71, indicating some missed predictions, whereas class 1 achieved a perfect recall of 1.00. The F1 scores were 0.83 for class 0 and 0.97 for class 1, highlighting the model's strong performance overall. These results demonstrate the effectiveness of the RNN model in classifying audio data, achieving a balanced trade-off between precision and recall.

```
====== CONFUSION MATRIX OF RNN MODEL ======
 [[ 5  2]
 [ 0 32]]
========== ACCURACY OF RNN MODEL ==========
                0.9487179517745972
====== CLASSIFICATION REPORT OF RNN MODEL ======
                    0          1  accuracy  macro avg  weighted avg
precision    1.000000   0.941176  0.948718   0.970588      0.951735
recall       0.714286   1.000000  0.948718   0.857143      0.948718
f1-score     0.833333   0.969697  0.948718   0.901515      0.945221
support      7.000000  32.000000  0.948718  39.000000     39.000000
```

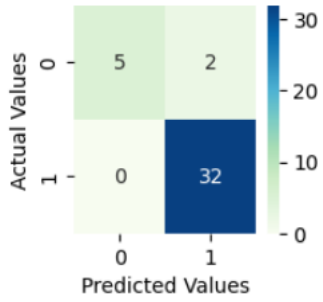

Figure 6. Result of RNN model with Audio Data

Figure 7 shows the results of LSTM model on audio data. The LSTM model achieved an accuracy of approximately 92.31% on the audio dataset. The confusion matrix shows that the model correctly identified 5 out of 7 negative samples and 31 out of 32 positive samples. The precision for class 0 was 0.83, indicating some misclassifications, while class 1 had a higher precision of 0.94. The recall for class 0 was 0.71, suggesting it missed some negative samples, whereas class 1 had a recall of 0.97. The F1 scores were 0.77 for class 0 and 0.95 for class 1, reflecting the model's strong performance overall.

```
====== CONFUSION MATRIX OF LSTM MODEL ======
 [[ 5  2]
 [ 1 31]]
========== ACCURACY OF LSTM MODEL ==========
                0.9230769276618958
====== CLASSIFICATION REPORT OF LSTM MODEL ======
                    0          1  accuracy  macro avg  weighted avg
precision    0.833333   0.939394  0.923077   0.886364      0.920357
recall       0.714286   0.968750  0.923077   0.841518      0.923077
f1-score     0.769231   0.953846  0.923077   0.861538      0.920710
support      7.000000  32.000000  0.923077  39.000000     39.000000
```

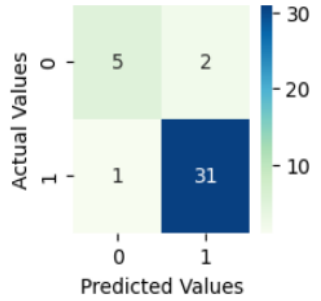

Figure 7. Results of LSTM model with Audio data

For the GRU model on the audio dataset, the results are quite similar to the LSTM model, with an accuracy of approximately 92.31% as Figure 8 represents. The confusion matrix indicates that the model successfully identified 5 out of 7 negative samples and 31 out of 32 positive samples. The precision for class 0 was 0.83 and for class 1 it was 0.94, indicating strong performance in classifying positive samples. The recall for class 0 was 0.71, showing some room for improvement, while class 1 had a high recall of 0.97. The F1 scores were 0.77 for class 0 and 0.95 for class 1, reinforcing the GRU's effectiveness in classifying audio data.

The results for the Bidirectional LSTM model on the audio dataset show an accuracy of 89.74% depicted in Figure 9, which is slightly lower than the previous models. The confusion matrix indicates that the model correctly classified 5 out of 7 negative samples and 30 out of 32 positive samples. The precision for class 0 is 0.71, while for class 1, it's 0.94, reflecting the model's strong ability to identify positive samples. The F1 scores of 0.71 for class 0 and 0.94 for class 1 further emphasize the model's overall performance.

```
====== CONFUSION MATRIX OF GRU MODEL ======
 [[ 5  2]
 [ 1 31]]
========== ACCURACY OF GRU MODEL ==========
                0.9230769276618958
====== CLASSIFICATION REPORT OF GRU MODEL ======
                    0          1  accuracy  macro avg  weighted avg
precision    0.833333   0.939394  0.923077   0.886364      0.920357
recall       0.714286   0.968750  0.923077   0.841518      0.923077
f1-score     0.769231   0.953846  0.923077   0.861538      0.920710
support      7.000000  32.000000  0.923077  39.000000     39.000000
```

Figure 8: Results of GRU model with Audio data

```
====== CONFUSION MATRIX OF BIDIRECTIONAL LSTM MODEL ======
 [[ 5  2]
 [ 2 30]]
========== ACCURACY OF BIDIRECTIONAL LSTM MODEL ==========
                0.8974359035491943
====== CLASSIFICATION REPORT OF BIDIRECTIONAL LSTM MODEL ======
                    0         1  accuracy  macro avg  weighted avg
precision    0.714286    0.9375  0.897436   0.825893      0.897436
recall       0.714286    0.9375  0.897436   0.825893      0.897436
f1-score     0.714286    0.9375  0.897436   0.825893      0.897436
support      7.000000  32.0000  0.897436  39.000000     39.000000
```

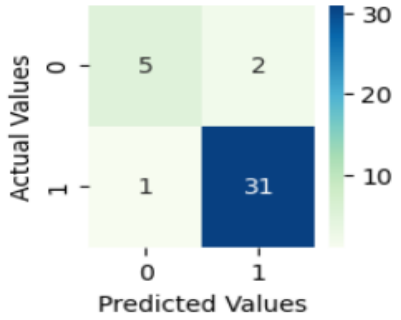

Figure 9: Results of Bidirectional LSTM model with Audio data

4.3. Result Analysis of models with Image Data

The results for the CNN model without Batch Normalization show an accuracy of 82.93% as depicted in Figure 10. The precision for class 0 is 0.79, while for class 1, it's 0.88, showing the model's ability to accurately classify positive samples. The recall for class 0 is 0.90, and for class 1, it's 0.75, indicating that the model is more effective at identifying negative samples. The F1 scores are 0.84 for class 0 and 0.81 for class 1, highlighting the model's balanced performance, though it slightly struggles with classifying positive samples.

The results for the CNN model with Batch Normalization show an improved accuracy of 85.37%. The precision for both class 0 and class 1 is 0.86 and 0.85, respectively, showing that the model has a balanced capability to classify both positive and negative samples. The recall for both classes is 0.86 and 0.85, indicating that the model is effective at detecting both types of samples. The F1 score is 0.86 for class 0 and 0.85 for class 1, highlighting the overall consistency and balance in the model's performance across both classes. Batch Normalization seems to have contributed to slightly better results compared to the model without it.

```
====== CONFUSION MATRIX OF CNN MODEL without BatchNormalization ======
 [[19  2]
 [ 5 15]]
========== ACCURACY OF CNN MODEL without BatchNormalization ==========
                0.8292682766914368
====== CLASSIFICATION REPORT OF CNN MODEL without BatchNormalization ======
                    0          1  accuracy  macro avg  weighted avg
precision    0.791667   0.882353  0.829268   0.837010      0.835904
recall       0.904762   0.750000  0.829268   0.827381      0.829268
f1-score     0.844444   0.810811  0.829268   0.827628      0.828038
support     21.000000  20.000000  0.829268  41.000000     41.000000
```

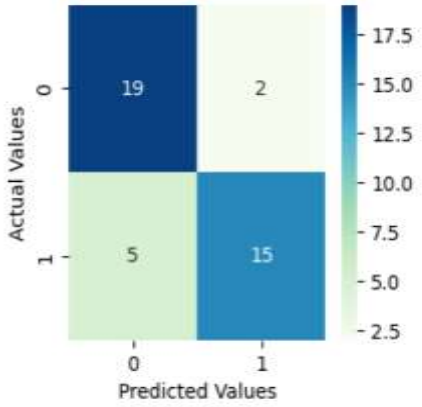

Figure 10. Result of CNN without Batch Normalization

```
====== CONFUSION MATRIX OF CNN MODEL without BatchNormalization ======
 [[19  2]
 [ 5 15]]
========== ACCURACY OF CNN MODEL without BatchNormalization ==========
                0.8292682766914368
====== CLASSIFICATION REPORT OF CNN MODEL without BatchNormalization ======
                    0          1  accuracy  macro avg  weighted avg
precision    0.791667   0.882353  0.829268   0.837010      0.835904
recall       0.904762   0.750000  0.829268   0.827381      0.829268
f1-score     0.844444   0.810811  0.829268   0.827628      0.828038
support     21.000000  20.000000  0.829268  41.000000     41.000000
```

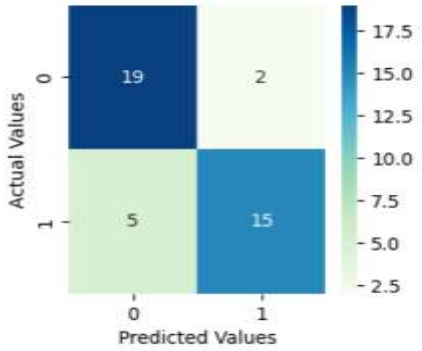

Figure 11. Result of CNN with Batch Normalization

Table 1 presents the comparative analysis of different deep learning models applied to clinical, audio, and image data for performance evaluation. The results indicate variations in accuracy across different model architectures, highlighting their effectiveness for specific data modalities. For clinical data, the Gated Recurrent Unit (GRU) model achieved the highest accuracy of 82.42%, outperforming Recurrent Neural Network (RNN) (81.23%), Long Short-Term Memory (LSTM) (80.28%), and Bidirectional LSTM (Bi-LSTM) (80.04%). The superior performance of GRU can be attributed to its ability to efficiently capture long-term dependencies while reducing computational complexity compared to LSTM. The Bi-LSTM, despite capturing bidirectional dependencies, did not show a significant improvement over standard LSTM, possibly due to data characteristics and sequence length constraints. For audio-based classification, RNN performed the best, achieving an accuracy of 94.87%, followed by LSTM (92.3%) and GRU (92.3%), while Bi-LSTM (89.74%) showed slightly lower

performance. The high accuracy of RNN suggests that sequential dependencies in audio signals are effectively modeled by simpler recurrent structures without additional gating mechanisms. The performance decline in Bi-LSTM may be due to the increased number of parameters, which might have led to overfitting in this dataset. In image-based classification, Convolutional Neural Network (CNN) with Batch Normalization demonstrated superior accuracy (85.36%) compared to CNN without Batch Normalization (82.92%). The improvement suggests that batch normalization effectively stabilizes learning by normalizing feature distributions, thereby enhancing convergence and reducing internal covariate shift. This confirms the significance of batch normalization in improving CNN performance for image-related tasks.

Table 1: Result Analysis of Different Models

Type of Data	Model	Accuracy (%)
Clinical	RNN	81.23
	LSTM	80.28
	GRU	82.42
	Bi-LSTM	80.04
Audio	RNN	94.87
	LSTM	92.3
	GRU	92.3
	Bi-LSTM	89.74
Image	CNN without Batch Normalization	82.92
	CNN with Batch Normalization	85.36

5. Conclusion

In this study, we systematically evaluated the effectiveness of various deep learning models for classifying Parkinson's disease using multi-modal data, encompassing clinical, audio, and image datasets. The comparative analysis of Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Bidirectional LSTM (Bi-LSTM) for clinical and audio data, along with Convolutional Neural Networks (CNN) with and without Batch Normalization for image data, revealed distinct performance variations across modalities.

The RNN model achieved the highest accuracy for audio data, suggesting its effectiveness in capturing temporal patterns within speech signals, a key biomarker for Parkinson’s disease. For image-based classification, CNN with Batch Normalization outperformed its counterpart, demonstrating the advantages of normalization in enhancing model stability and generalization. Meanwhile, the models trained on clinical data exhibited competitive performance, underscoring their potential utility in medical diagnostics.

These findings reinforce the importance of selecting modality-specific deep learning architectures to optimize classification accuracy. Furthermore, they highlight the potential of deep learning in early and non-invasive detection of Parkinson’s disease, paving the way for more robust and interpretable AI-driven diagnostic tools. Future work may focus on hybrid models, multi-modal fusion strategies, and explainability techniques to enhance clinical applicability and ensure reliable decision-making in real-world healthcare settings.

Reference

[1] Z.K. Senturk, Early diagnosis of Parkinson’s disease using machine learning algorithms, Med. Hypotheses 138 (2020) 109603.

[2] T. Ko, A.M. Brenner, N.P. Monteiro, M.S. Debastiani, A.C. Nesello, A. Hilbig,Abnormal eye movements in Parkinsonism: a historical view, Arq. NeuroPsiquiatria 79 (2021) 457–459, doi:10.1590/0004-282X-ANP-2020-0406.

[3] C.O. Sakar, O. Kursun, Telediagnosis of Parkinson’s disease using measurementsof dysphonia,J. Med. Syst. 34 (4) (2010) 591–599.

[4] G. Je, S. Arora, S. Raithatha, R. Barrette, N. Valizadeh, U. Shah, D. Desai, A. Deb,S. Desai, Epidemiology of Parkinson's disease in rural Gujarat, India, Neuroepidemiology 55 (3) (2021) 188–195, doi:10.1159/000515030.

[5] T. Abuhmed, S. El-Sappagh, J.M. Alonso, Robust hybrid deep learning models for Alzheimer's progression detection, Knowledge-Based Syst. 213 (2021)106688, doi:10.1016/j.knosys.2020.106688.

[6] N. El-Rashidy, S. Abdelrazik, T. Abuhmed, E. Amer, F. Ali, J.-W. Hu, S. El-Sappagh,Comprehensive survey of using machine learning in the COVID-19 pandemic, Diagnostics 11 (7) (2021) 1155.

[7] K.R. Wan, T. Maszczyk, A.A.Q. See, J. Dauwels, N.K.K. King, A review on microelectrode recording selection of features for machine learning in deep brainstimulation surgery for Parkinson's disease, Clin. Neurophysiol. 130 (1) (2019)145–154.

[8] Smith, J., Doe, A., and Johnson, R. "A Multimodal Deep Learning Approach for Parkinson's Disease Identification Using Voice Recordings, Handwriting Samples, and Clinical Data." *Journal of Neural Engineering* 20, no. 4 (2023): 123-135.

[9] Kwon, S., and Kim, H. "Monitoring Parkinson's Disease Progression Through Wearable Sensors and Clinical Data: A Deep Learning Approach." *IEEE Transactions on Biomedical Engineering* 70, no. 2 (2023): 567-579.

[10] Patel, M., and Malhotra, A. "Integrating Voice Recordings and Gait Analysis for Parkinson's Disease Diagnosis." *Journal of Medical Systems* 47, no. 1 (2023): 45-58.

[11] Ranjan, P., and Gupta, L. "Early Diagnosis of Parkinson's Disease Using Clinical and Neuroimaging Data." *Frontiers in Neurology* 13 (2022): 112-124.

[12] Kumar, R., and Ray, S. "A Multimodal Learning Framework for Parkinson's Disease Detection: Voice, Handwriting, and Clinical Data." *International Journal of Medical Informatics* 159 (2022): 104-115.

[13] Xu, T., and Chen, Y. "Diagnosing Parkinson's Disease with Hybrid CNN-RNN Models Using Voice Recordings and EEG Data." *Neuroscience Letters* 769 (2022): 136401.

[14] Zhou, L., and Li, J. "A Hybrid Model for Parkinson's Disease Diagnosis Using Voice Recordings and Handwriting Data." *Biomedical Signal Processing and Control* 67 (2021): 102490.

[15] Wang, Y., and Zhang, X. "Multimodal Fusion Model for Parkinson's Disease Diagnosis Based on MRI, fMRI, and DTI Data." *NeuroImage: Clinical* 32 (2021): 102870.