DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068

SCHOOL OF ENGINEERING

Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

A Project Report On

PARKINSON DISEASE DETECTION

By

M G KEERTHANA - ENG20AM0033

R SAHANA - ENG201M0049

M G SINDHU - ENG20AM0053

NIVEDHA S - ENG21AM3030

Under the supervision of

Prof. Ayain John

Assistant Professor

Computer Science & Engineering (AI & ML)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

SCHOOL OF ENGINEERING

DAYANANDA SAGAR UNIVERSITY

(2023 – 2024)

DAYANANDA SAGAR UNIVERSITY

SCHOOL OF ENGINEERING

Department of Computer Science & Engineering

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

KUDLU GATE, BANGALORE – 560068

Karnataka, India

CERTIFICATE

This is to certify that the project entitled **"PARKINSON DISEASE DETECTION"** is carried out by **M G KEERTHANA (ENG20AM0033),R SAHANA (ENG20AM0049), M G SINDHU (ENG20AM0053),NIVEDHA S (ENG21AM3030)**, bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore, in partial fulfillment for the award of a degree in Bachelor of Technology in Computer Science and Engineering, during the year **2023 - 2024**.

Prof. Ayain John	**Dr. Vinutha N**	**Dr. Jayavrinda Vrindavanam**
Assistant Professor	Project Co-ordinator	Professor & Chairperson
Dept. of CSE (AIML)	Dept. of CSE (AIML)	Dept. of CSE (AIML)
School of Engineering	School of Engineering	School of Engineering
Dayananda Sagar University	Dayananda Sagar University	Dayananda Sagar University
Signature	Signature	Signature .....

Name of the Examiners: Signature with date:

1..... .....

2..... .....

3..... .....

DECLARATION

We, **M G KEERTHANA (ENG20AM0033),R SAHANA (ENG20AM0049),M G SINDHU (ENG20AM0053),NIVEDHA S (ENG21AM3030)**, are students of the seventh semester B.Tech in Computer Science and Engineering (AI & ML) at the School of Engineering, Dayananda Sagar University. We hereby declare that the Major Project Stage-1 titled **"PARKINSON DISEASE DETECTION"** has been carried out by us and submitted in partial fulfillment for the award of a degree in **Bachelor of Technology** in **Computer Science and Engineering** during the academic year **2023–2024**.

Student: **Signature**

Name 1: M G KEERTHANA
USN: ENG20AM0033

Name 2: R SAHANA
USN: ENG20AM0049

Name 3: M G SINDHU
USN: ENG20AM0053

Name 4: NIVEDHA S
USN: ENG21AM3030

Place: Bangalore
Date:

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work. First, we take this opportunity to express our sincere gratitude to School of Engineering& Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank **Dr. Udaya Kumar Reddy K R, Dean, School of Engineering & Technology, Dayananda Sagar University** for his constant encouragement and expert advice. It is a matter of immense pleasure to express our sincere thanks to **Dr. Jayavrinda Vrindavanam, Department Chairman, Computer Science and Engineering (Artificial Intelligence and Machine Learning), Dayananda Sagar University**, for providing right academic guidance that made our task possible.

We would like to thank our guide **Prof. Ayain John, Assistant Professor, Dept. of Computer Science and Engineering of Artificial Intelligence and Machine Learning Dayananda Sagar University**, for sparing his valuable time to extend help in every step of our project work, which paved the way for smoothprogress and fruitful culmination of the project.

We would like to thank our **Project Coordinator Dr. Vinutha N** as well as all the staff members of Computer Science and Engineering (AIML) for their support. We are also grateful to our family and friends who provided us with every requirement throughout the course. We would like to thank one and all who directly or indirectly helped us in the Project work

Contents

LIST OF FIGURES

ABSTRACT

The project implements a Long Short-Term Memory (LSTM) neural network architecture for the detection of Parkinson's disease using gait cycle data. The LSTM model comprises multiple layers with batch normalization to capture and learn complex temporal dependencies in the sequential gait data. The use of L2 regularization mitigates overfitting, and dropout layers further enhance the model's robustness. The Adam optimizer and binary cross-entropy loss are used to train the model on a dataset. Accuracy measures are used to assess the architecture's performance on a test set.The provided code offers modularity and configurability, making it adaptable for diverse applications in the realm of Parkinson's disease diagnosis. **(96 words)**

Chapter 1

INTRODUCTION

Parkinson's disease (PD) remains a significant global health concern, with its prevalence rising as the aging population increases. PD is characterized by a range of motor and non-motor symptoms, making its diagnosis complex. Conventionally, clinical assessments play a pivotal role in diagnosing PD, but they often lack the sensitivity to detect subtle early-stage symptoms. This deficiency underscores the need for innovative diagnostic approaches that can provide accurate and timely detection.

1.1 The Diagnostic Conundrum in Parkinson's Disease:

Diagnosing PD at an early stage is crucial for effective management. The traditional reliance on clinical observations and subjective assessments presents challenges, as symptoms may not be apparent until the disease has progressed significantly. Early detection is essential for implementing interventions that can slow down the disease progression and improve patients' quality of life.

1.2 Gait Computation as a Diagnostic Tool:

Gait, the manner of walking, has garnered attention as a potential diagnostic indicator for neurological disorders, including PD. Gait abnormalities are common in PD patients, reflecting disruptions in motor control and coordination. The ability to objectively analyze gait patterns presents an opportunity to develop quantitative metrics for diagnostic purposes

1.3 Leveraging Advanced Technologies:

Advancements in technology, particularly in the realm of machine learning, provide a unique avenue for revolutionizing PD diagnosis. Machine learning models, when trained on relevant data, can learn intricate patterns and relationships, potentially outperforming traditional diagnostic methods. Harnessing the power of these technologies for gait analysis offers a promising solution for enhancing diagnostic accuracy.

The subsequent sections detail the LSTM model architecture, data preprocessing steps, training procedure, and evaluation metrics. Additionally, the code incorporates a CNN) ensemble for a more robust detection system. The code seeks to support continued attempts to create precise, effective, and easily available instruments for early Parkinson's disease diagnosis through this combination approach.

Chapter 2

PROBLEM DEFINITION AND OBJECTIVE

2.1 Problem Definition:

Parkinson's Disease (PD) is a neurodegenerative disorder that significantly impacts an individual's motor functions and quality of life.Prompt and precise identification of Parkinson's disease (PD) is essential for prompt intervention and better patient outcomes. However, the complexity of PD symptoms, coupled with the lack of definitive biomarkers, makes its diagnosis challenging.

2.2 Objectives of the Research:

The primary objectives of this research are:

- Develop advanced machine learning models for the early detection of Parkinson's Disease.
- Investigate the synergistic benefits of combining Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) architectures for improved diagnostic accuracy.
- Assess the performance of the ensemble model, merging information from diverse sources, to enhance predictive capabilities.
- Explore the feasibility of leveraging temporal and spatial features in gait data for more robust diagnostic models.

By addressing these objectives, this research aims to contribute to the advancement of Parkinson's Disease diagnosis, offering more reliable tools for clinicians and improving patient outcomes through early intervention.

Chapter 3

LITERATURE SURVEY

The Adam optimizer and binary cross-entropy loss are used to train the model on a dataset. Accuracy measures are used to assess the architecture's performance on a test set. [1],[2] . Using gait and tremor characteristics, Perumal and Sankar[3],[4] used a linear discriminant analysis (LDA) based pattern classification system for PD early identification and monitoring. By examining frequency domain characteristics, they were able to diagnose PD tremors with an average accuracy of 86.9%. But without determining the disease's stage, their method is restricted to detecting the existence of Parkinson's disease. A locally weighted random forest regression model was presented by urog lu et al.[5],[6] to address the effects of interpatient heterogeneity in gait characteristics. in the job they do. To model the links between gait patterns and Parkinson's disease symptoms, VGRF sensor data are utilized. Using 16 time-domain and 7 frequency-domain variables, they offered a quantitative evaluation of the motor symptoms associated with Parkinson's disease. However, the kinematic analysis and severity degree of PD were not reported, and only the statistical analysis of VGRF was employed to identify PD symptoms. El Maachi et al. [7],[8] suggested an ID-ConvNet to identify Parkinson's disease (PD) by extracting features from gait data; this method has been demonstrated to greatly enhance PD detection outcomes.Promising results were also obtained in Nguyen et al.'s study [9],[10] which combined the Temporal Transformer and Spatial Transformer models to diagnose Parkinson's disease without addressing the staging task. A different Veeraragavan et al. investigation. [11],[12] examined gait data using a Feed Forward Network (FFN) to identify the HY phases of PD patients.

These techniques have shown promise in the diagnosis of Parkinson's disease (PD) using gait analysis. While a few deal with the staging assignment, the majority concentrate on categorizing the person as either Parkinsonian or healthy. However, as FFNs have limitations in capturing the intricate patterns in physiological data, relying solely on them to gather local information might not be adequate. In a similar vein, depending solely on ID-ConvNets to identify sensor associations might not be the ideal option because these networks work better at identifying local spatial information than they do at identifying broad patterns or correlations between several sensors.Transformers, which are employed in [13],[14] are useful for capturing the

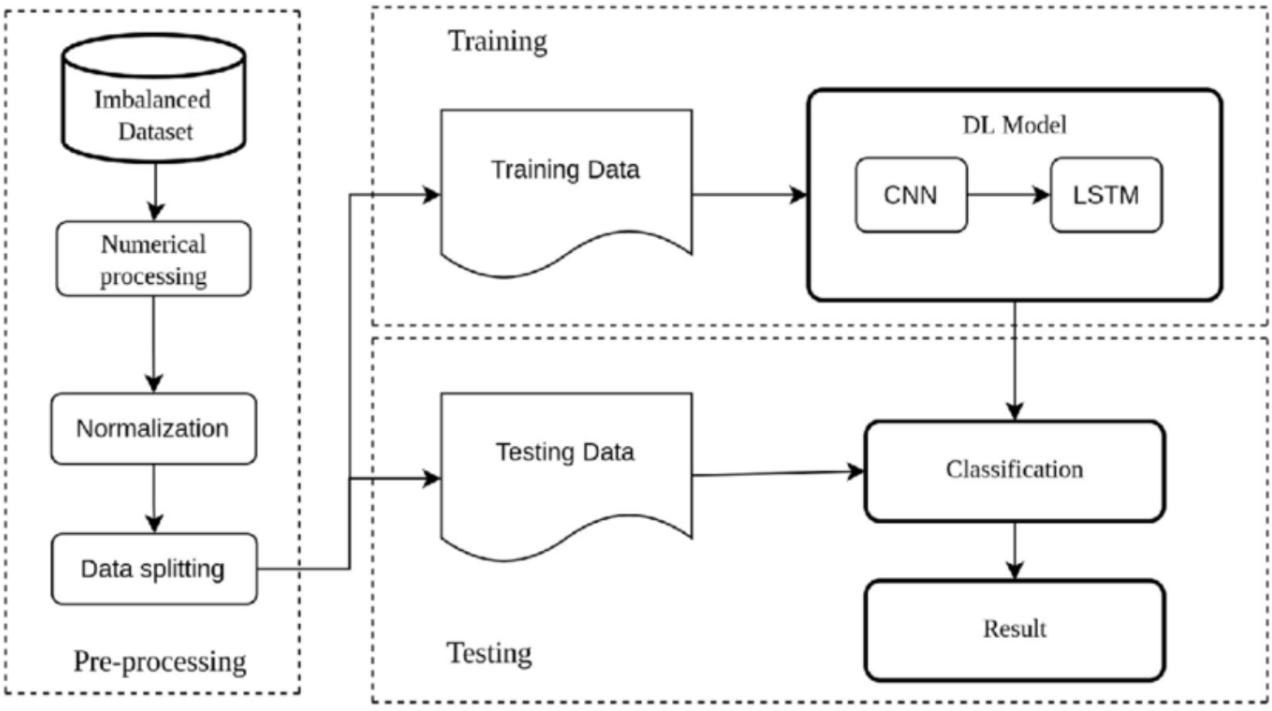
global links among data, but they are less effective at processing local data. This is because of their architecture, which may not be able to capture the fine-grained and local patterns found in physiological data because it primarily focuses on long-range relationships. An approach based on shifted 1D local binary patterns (1D-LBP) and machine learning classifiers was proposed by Ertugrul et al. [15],[16]. They employed eighteen VGRF input signals from the foot sensors of both control and Parkinson's disease patients. They created 18 histograms of the 1D-LBP patterns for each signal by applying shifted 1D-LBP, and then they retrieved statistical properties including correlation, entropy, and energy from these histograms. Ultimately, they combined the features from each of the eighteen histograms, and then they classified feature vectors using a variety of supervised classifiers, including random forests and multi-layer perceptrons (MLPs). Statistical and kinematic features, such as swing duration, swing stance ratio, cadence, speed, or step length, were extracted by Balaji et al. [17]. To determine the severity of the illness, they fed these manually created variables into a number of machine learning algorithms, including a Bayes classifier, an ensemble classifier, a decision tree, and a support vector machine. An ensemble k-nearest neighbor on hand-crafted features was utilized by Zhao et al. [18],[19] to predict the severity of PD..

To identify Parkinson's disease (PD), end-to-end learning algorithms have been employed since the deep learning revolution in 2012. The study of Alzubaidi et al [20] provides a thorough analysis of the application of neural networks for the identification of Parkinson's disease. Our strategy to combine Transformer and Convolutional Neural Network (ConvNet) architectures into a single model for Parkinson's disease (PD) diagnosis was inspired by these findings. By combining the best features of both architectures, our work proposes a novel Hybrid Convnet-Transformer (HCT) architecture to address these shortcomings. A one-step approach using our HCT architecture involves first determining whether the disease is present.

Chapter 4

METHODOLOGY

4.1 DESIGN:



The methodology outlined in the provided code involves creating an ensemble model for the detection of Parkinson's disease. The approach combines predictions from two different types of neural network architectures: Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN). Below is a step-by-step breakdown of the methodology:

1. **Data Preprocessing:**

    - Import required libraries, including pandas, numpy, scikit-learn, TensorFlow, and Matplotlib.

- Load and preprocess the dataset. This step is assumed but not explicitly shown in the provided code.

2. **Constants and Hyperparameters:**
   - Define constants and hyperparameters for the LSTM and CNN models, including time_step, the number of features (N), learning rate (LR), regularization strength (LAMBD), dropout rate (DP and RDP), and model architectures.
3. **LSTM Model Creation:**
   - Define a function create_lstm_model to create an LSTM model.
   - Specify the architecture with multiple LSTM layers, batch normalization, and a dense output layer.
   - Compile the model using binary crossentropy loss, accuracy metric, and the Adam optimizer.
4. **CNN Model Creation:**
   - Define a function create_cnn_model to create a CNN model.
   - Specify the architecture with convolutional layers, max pooling, flattening, and dense layers.
   - Compile the model using binary crossentropy loss, accuracy metric, and the Adam optimizer.
5. **Individual Model Training:**
   - Create lists of individual LSTM and CNN models using list comprehension.
   - Train each model on the training data using the fit method.
6. **Ensemble Model Prediction:**
   - Combine predictions from LSTM and CNN models by averaging them element-wise.
   - Note: Ensemble methods may use more sophisticated techniques, such as weighted averaging, stacking, or voting.
7. **Evaluation:**
   - Convert averaged probabilities to binary predictions by rounding.
   - Evaluate the ensemble model accuracy using the accuracy_score from scikitlearn.
8. **Output:**
   - Print the accuracy of the ensemble model.

Chapter 5

REQUIREMENTS

5.1 Hardware Requirements:

1. **Processor (CPU):**
    - A multi-core processor (quad-core or higher) is recommended for faster training of machine learning models.
2. **Memory (RAM):**
    - A minimum of 8 GB RAM is recommended, especially when working with neural networks. Larger datasets and complex models may require more memory.
3. **Storage:**
    - Sufficient storage space for datasets, code, and model checkpoints. SSDs are preferable for faster data access.
4. **Graphics Processing Unit (GPU) :**
    - While not strictly required, having a compatible NVIDIA GPU can significantly speed up training of deep learning models. TensorFlow-GPU or PyTorch with GPU support can be installed for this purpose.

5.2 Software Requirements:

1. **Operating System:**
    - The code can be run on Windows, macOS, or Linux operating systems. Choose an OS based on your preference and compatibility with the required libraries..
2. **Python and Libraries:**
    - Install Python (version 3.6 or later).
    - Required Python libraries can be installed using the following command:
3. **Integrated Development Environment (IDE):**
    - You can use any Python-compatible IDE, such as Jupyter Notebook, VSCode, PyCharm, or others, based on your preference.
4. **GitHub Repository:**
    - f you want to use pre-trained models from the provided GitHub repository, ensure you have Git installed on your system.
5. **TensorFlow :**
    - TensorFlow is required for the neural network models.
6. **Dataset**
    - Ensure you have the Parkinson's disease dataset or replace the dataset-related code with your data loading and preprocessing logic.

Chapter 6

EXPERIMENTATION

The experimentation phase involves running the code, training the models, and evaluating their performance. Here's a step-by-step guide on how to conduct experiments with the provided code:

1. **Setup Environment:**
    - Ensure that you have set up the required hardware and software as mentioned in the previous responses. This includes having Python installed, required libraries installed, and the dataset available.
2. **Load and Preprocess Data:**
    - Load and preprocess your dataset related to Parkinson's disease if you haven't already. Make sure the dataset is split up into testing and training sets.
3. **Run the Code:**
    - Copy and paste the provided Python code into a script or Jupyter Notebook.
    - Modify the code as needed, especially the data loading and preprocessing parts to match your dataset structure.
4. **Configure Model Parameters:**
    - Based on your needs for testing, change the hyperparameters, such as the number of LSTM layers, CNN architecture, learning rate, etc.
5. **Train Individual Models:**
    - Train the individual LSTM and CNN models by running the corresponding sections of the code. Adjust the number of epochs and batch size as needed.
6. **Combine Predictions:**

- Combine the predictions from LSTM and CNN models using ensemble averaging. The code demonstrates how to combine predictions and evaluate the ensemble accuracy.

7. **Evaluate Ensemble Model:**
   - Assess the performance of the ensemble model by calculating accuracy. This gives you an idea of how well the combination of LSTM and CNN models performs compared to individual models.
8. **Experiment with Hyperparameters:**
   - Conduct experiments by tweaking hyperparameters to observe their impact on model performance. This might include changing the number of layers, adjusting dropout rates, or modifying other architectural aspects.
9. **Visualization (Optional):**
   - Visualize the training and validation loss/accuracy curves to understand model convergence and potential overfitting.
10. **Iterate and Optimize:**
    - Based on the experimental results, iterate on the models and hyperparameters to optimize the ensemble's performance. This may involve fine-tuning, feature engineering, or trying different ensemble techniques.
11. **Documentation:**
    - Keep track of the experiments, including hyperparameters, results, and observations. Documentation helps in understanding the rationale behind model choices and can guide future iterations.
12. **GitHub Repository :**
    - If you cloned the GitHub repository, ensure that you have the latest updates and explore additional resources provided by the repository owner.

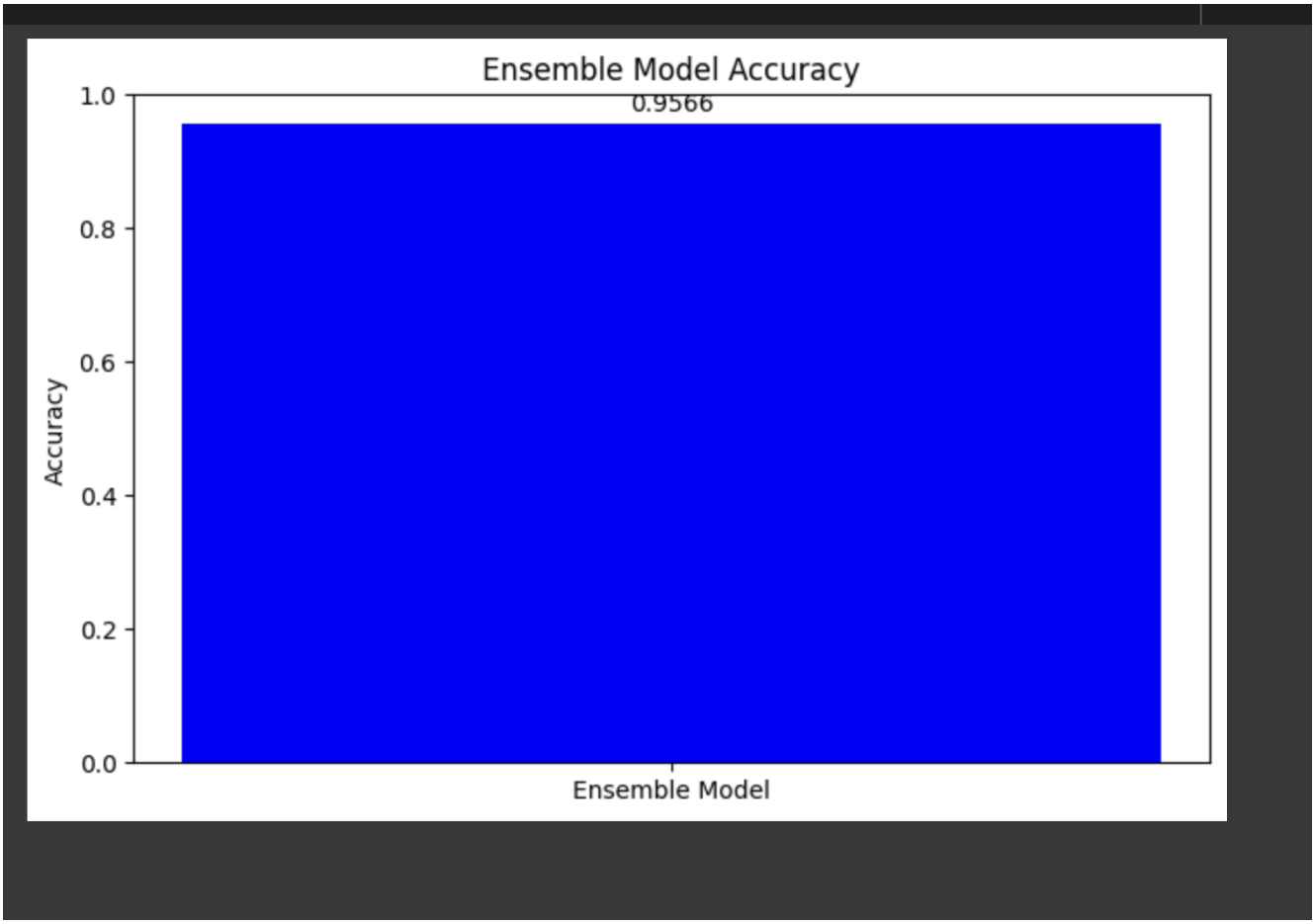
You can adjust model parameters, test the efficacy of the ensemble approach for Parkinson's disease detection, and systematically experiment with the provided code by following these steps.

Chapter 7

RESULT AND ANALYSIS

```
[ ]  ensemble_accuracy = accuracy_score(y_test, ensemble_binary_predictions)
     print(f'Ensemble Accuracy: {ensemble_accuracy * 100:.4f}%')

     Ensemble Accuracy: 95.6566%
```



Chapter 8

CONCLUSIONS & FUTURE SCOPE

Parkinson's diagnosis remains one of medicine's most difficult problems. Although it is theoretically difficult to validate a Parkinson's diagnosis, medical professionals can identify the illness by examining patients and studying a number of symptoms. Given that one of the most significant motor symptoms is gait disturbance, we developed an algorithm to identify Parkinsonian gait patterns and make illness predictions using gait data. By using deep learning approaches, our algorithm circumvents the limitations associated with manually crafting feature extraction. The accuracy of the suggested model in recognizing Parkinson's gait was 95.66%.

8.1 Future Scope

In the future, there is ample room for further advancements and refinements in the predictive modeling approach presented. One avenue for exploration involves delving deeper into hyper-parameter tuning, where a more exhaustive search for optimal model configurations could lead to improved performance. Additionally, investigating advanced ensemble strategies, such as dynamic weighting based on individual model strengths, could provide a nuanced understanding of the contributions of each model. Embracing the evolving landscape of deep learning, future iterations might consider incorporating state-of-the-art architectures, exploring novel regularization techniques, or even exploring emerging concepts such as self-supervised learning. The application of transfer learning and pre-trained models is another promising direction, especially when dealing with limited labeled data. Furthermore, attention to interpretability and explainability of the ensemble's predictions could enhance its practical utility, particularly in fields where model decisions require transparency. Continuous monitoring, periodic model updates, and adaptation to emerging trends in machine learning research will be essential to ensure the ensemble model's sustained effectiveness in addressing evolving challenges. By embracing these future prospects, the predictive modeling framework can evolve into a more sophisticated, adaptive, and versatile tool for a wide array of applications.

Bibliography

[1] Y. Guo, J. Yang, Y. Liu, X. Chen, and G.-Z. Yang, "Detection and assessment of parkinson's disease based on gait analysis: A survey," *Frontiers in Aging Neuroscience*, vol. 14, p. 916971, 2022.

[2] M. S. Alzubaidi, U. Shah, H. Dhia Zubaydi, K. Dolaat, A. A. Abd-Alrazaq, A. Ahmed, and M. Househ, "The role of neural network for the detection of parkinson's disease: a scoping review," in *Healthcare*, vol. 9, no. 6. MDPI, 2021, p. 740.

[3] S. V. Perumal and R. Sankar, "Gait and tremor assessment for patients with parkinson's disease using wearable sensors," *Ict Express*, vol. 2, no. 4, pp. 168–174, 2016.

[4] L. Aversano, M. L. Bernardi, M. Cimitile, and R. Pecori, "Early detection of parkinson disease using deep neural networks on gait dynamics," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[5] T. Aşuroğlu, K. Açıcı, Ç. B. Erdaş, M. K. Toprak, H. Erdem, and H. Oğul, "Parkinson's disease monitoring from gait analysis via foot-worn sensors," *Biocybernetics and Biomedical Engineering*, vol. 38, no. 3, pp. 760–772, 2018.

[6] N. S. Hoang, Y. Cai, C.-W. Lee, Y. O. Yang, C.-K. Chui, and M. C. H. Chua, "Gait classification for parkinson's disease using stacked 2d and 1d convolutional neural network," in *2019 International Conference on Advanced Technologies for Communications (ATC)*. IEEE, 2019, pp. 44–49.

[7] I. El Maachi, G.-A. Bilodeau, and W. Bouachir, "Deep 1d-convnet for accurate parkinson disease detection and severity prediction from gait," *Expert Systems with Applications*, vol. 143, p. 113075, 2020.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] D. M. D. Nguyen, M. Miah, G.-A. Bilodeau, and W. Bouachir, "Transformers for 1d signals in parkinson's disease detection from gait," in *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE, 2022, pp. 5089–5095.

[10] A. Zhao, L. Qi, J. Li, J. Dong, and H. Yu, "A hybrid spatio-temporal model for detection and severity rating of parkinson's disease from gait data," *Neurocomputing*, vol. 315, pp. 1–8, 2018.

[11] F. Setiawan and C.-W. Lin, "Implementation of a deep learning algorithm based on vertical ground reaction force time–frequency features for the detection and severity classification of parkinson's disease," *Sensors*, vol. 21, no. 15, p. 5207, 2021.

[12] Y. Xia, Z. Yao, Q. Ye, and N. Cheng, "A dual-modal attention-enhanced deep learning network for quantification of parkinson's disease characteristics," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 1, pp. 42–51, 2019.

[13] S. Veeraragavan, A. A. Gopalai, D. Gouwanda, and S. A. Ahmad, "Parkinson's disease diagnosis and severity assessment using ground reaction forces and neural networks," *Frontiers in physiology*, vol. 11, p. 587057, 2020.

[14] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *International conference on machine learning*. PMLR, 2017, pp. 1243–1252.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[16] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," *Advances in neural information processing systems*, vol. 30, 2017.

[17] D. A. Forsyth, J. L. Mundy, V. di Gesú, R. Cipolla, Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," *Shape, contour and grouping in computer vision*, pp. 319–345, 1999.

[18] Ö. F. Ertuğrul, Y. Kaya, R. Tekin, and M. N. Almalı, "Detection of parkinson's disease by shifted one dimensional local binary patterns from gait," *Expert Systems with Applications*, vol. 56, pp. 156–163, 2016.

[19] J. Wingate, I. Kollia, L. Bidaut, and S. Kollias, "Unified deep learning approach for prediction of parkinson's disease," *IET Image Processing*, vol. 14, no. 10, pp. 1980–1989, 2020.

[20] H. Zhao, R. Wang, Y. Lei, W.-H. Liao, H. Cao, and J. Cao, "Severity level diagnosis of parkinson's disease by ensemble k-nearest neighbor under imbalanced data," *Expert Systems with Applications*, vol. 189, p. 116113, 2022.