Tracking Parkinson's Disease Progression Using Deep Learning: A Hybrid Auto Encoder and Bi-LSTM Approach

Sri Lavanya Sajja[1], Dr. Kabilan Annadurai[2], Dr. S. Kirubakaran[3], TK Rama Krishna Rao[4], Dr. P. Satish[5], Elangovan Muniyandy[6], Yahia Said[7]*

Assistant Professor, Department of CSE, Malla Reddy Engineering College for Women, Secunderabad, India[1]
Department of Public Health-School of Health Sciences, The Apollo University, Chittoor, Andhra Pradesh-517002, India[2]
Professor/CSE (AI & ML), CMR College of Engineering and Technology, Kandlakoya, Hyderabad 501 401, Telangana, India[3]
Professor, Department of Computer Science and Engineering, Koneru Lakshmaih Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India-522302[4]
Associate Professor, Department of Mathematics, Aditya University, Surampalem, India- 533437[5]
Department of Biosciences-Saveetha School of Engineering. Saveetha Institute of Medical and Technical Sciences, Chennai - 602 105, India[6]
Applied Science Research Center. Applied Science Private University, Amman, Jordan[6]
Center for Scientific Research and Entrepreneurship, Northern Border University, 73213, Arar, Saudi Arabia[7]

***Abstract*—Parkinson's disease (PD) is a progressive and chronic neurodegenerative disorder characterized by motor impairment, speech deficits, and cognitive decline. Monitoring disease progression accurately and intermittently is imperative for early treatment planning and personalized intervention. In the past, conventional methods of diagnosis—clinical examination and traditional machine learning (ML) algorithms—tend to be insufficient in identifying intricate temporal behaviors of PD progress and involve frequent clinic visits. There is no cure for this disease but there are treatments. To tackle these issues, we introduce a deep learning (DL)-based approach that integrates auto encoders for feature learning with Bi-Directional Long Short-Term Memory (Bi-LSTM) networks for temporal sequence modeling. The hybrid model successfully monitors PD severity over time by learning complex patterns in the data. We measure our method with the Parkinson's Tele monitoring Dataset from the UCI Machine Learning Repository, which contains longitudinal voice recordings together with Unified Parkinson's Disease Rating Scale (UPDRS) scores—rendering it particularly well-suited for time-series analysis. Implemented in Python with Tensor Flow applies sophisticated training methods to achieve maximum performance. Experimental results affirm a dramatic improvement compared to traditional ML methods, producing an accuracy rate of 95.2%. Such high predictive power facilitates timely adjustment of treatment and improves patient management. The suggested model presents a non-invasive, scalable real-time PD monitoring solution. It aids neurologists, clinicians, and researchers by offering an AI-based platform for pre-emptive intervention. It helps psatients by facilitating continuous remote monitoring, minimizing frequent clinic visits, and enhancing their quality of life.**

Keywords*—Auto encoders; DL; Parkinson's disease; Bi-LSTM; tele monitoring dataset

I. INTRODUCTION

PD is a long term brain disorder that cannot be cured. It causes problems in movement with speech and mental impairment. There is a chemical called dopamine in the brain which slowly loses every day. Traditional ML models needs to collect features which takes a lot of time. Datasets of Parkinson's disease has only few samples so it is difficult to learn correctly for old models. [1]. Parkinson's disease slowly progresses through several stages from mild impairment of motor to extreme disability. Tremor, bradykinesia, muscle stiffness, and postural instability are classic features. Depression, anxiety, and trouble in thinking are the symptoms affecting the patient's quality of life. Many conventional methods depend on face to- face which is not required for monitoring in rural areas. [2]. Early analysis is important for successful treatment, but some methods depends heavily on clinical assessments that may be time-consuming and subjective[3]. In most cases, symptoms of PD becomes difficult to diagnose the disease during its initial phase. The demand for an objective, automated, and non-invasive monitoring system has prompted research into ML and DL techniques to learn from patient data and predict high accuracy in the severity of the disease[4].

Traditional ML algorithms like SVM, DT and RF have been utilized for PD detection and severity estimation based on biomedical voice and movement data[5]. Although these algorithms offer good classification accuracy they are not effective in modeling the intricate temporal patterns of PD symptoms, which evolve over time. Some works have investigated the use of CNN for PD detection[6], but CNNs are generally designed for extracting spatial features and they are not best for dealing with time-series data. LSTM networks, have been found effective in processing sequential data by learning long-term relationships[7]. But raw biomedical signals tend to involve noise and unrelated features, thus direct usage of RNNs is not efficient. Here proposes a DL model that improves the accuracy of PD progression monitoring by combining Auto encoders and RNNs. Auto encoders provide high-level latent

representations from raw data with reduced noise but with retained key features. These are fed into LSTM networks which capture the temporal trend of PD symptoms leading to a stronger and more interpretable prediction model.

The combination of Auto encoders and Bi-LSTMs improves feature extraction and temporal modeling, and our framework is suitable for PD severity prediction. Auto encoders dimensionally reduce and emphasize important features but Bi-LSTMs learn long-term dependencies in patient histories that efficiently capturing disease progression. The "Parkinson's Tele monitoring" Dataset comprises of biomedical voice measurements that is recorded from PD patients. This data set allows one to investigate speech impairment which is optimal for monitoring with DL. We deploy our model on Python and Tensor Flow for its scalability and performance. By applying DL, we can make remarkable increases in classification precision in contrast with ML techniques.

The key contributions of this work are:

1) Introduced a new architecture that integrates auto encoders for feature learning and Bi-LSTM networks for modeling temporal evolution in PD.

2) Employed longitudinal voice measurements and UPDRS ratings to accurately monitor disease severity as a function of time using sequential learning.

3) Created a deep learning-based system that enables remote and continuous monitoring of PD without requiring frequent clinical visits.

4) Developed an infrastructure that can help neurologists and medical practitioners with early intervention and customized treatment planning through the delivery of AI-based insights into the evolution of diseases.

The remainder of the study is structured as follows: An overview of the literature on PD detection is given in Section II. Problem statement is provided in Section III. The Auto encoder-Bi-LSTM model is covered in detail in Section IV. The outcome of test and discussion is given in Section V. Conclusions and recommendations for further research are provided in Section VI.

II. Related Works

Govindu and Palwe [8] investigated the use of ML classifiers for the early detection of PD using telemedicine. Their work compared SVM, RF, KNN and LR classifiers with a dataset made up of voice samples of patients and normal persons. Among all the classifiers mentioned, Random Forest proved to be the best with highest classification efficiency which rendered it a candidate classifier to distinguish between Parkinsonian speech patterns. The research highlighted the applicability of ML in remote disease tracking, especially when frequent hospital visits are a problem. The study is based on a fairly limited dataset that could not generalize to the larger population. Hireš et al. [9] proposed a DL methodology that utilized several CNN for detecting Parkinson's from voice recordings in a similar study. The model was trained with a fine-tuning approach to adjust pre-trained networks to the target data set. Evaluation on various vowel sounds proved its efficiency in classifying affected and non-affected speakers. Although this approach is used in clinics, it needed a significant amount of labeled data and processing capabilities for training.

Trabassi et al. [10] used supervised ML algorithms to predict PD patients from gait features extracted using inertial measurement unit (IMU) sensors. A three-stage feature selection was utilized to determine key gait parameters from trunk acceleration data of PD patients and healthy individuals. These chosen attributes were utilized in training the classifiers. The developed models like SVM, DT, and RF proved to have powerful classification ability. The research provided a conceptual model for ML-based gait analysis that reduces issues of multi-collinearity while improving interpretability. The size of the data used was fairly small, meaning that the generality of findings was restricted. Quan et al. [11] also compared various ML classifiers for the detection of PD using voice-based datasets from the UCI ML Repository. Their investigation compared the Multilayer Perceptron, SVM and KNN classifiers and settled with the most promising classifier as being the Multilayer Perceptron coupled with the Levenberg–Marquardt algorithm. Since the research made tremendous contributions toward classification selection, much of the analysis was laid in traditional ML methods rather than venturing into new models of DL for robust feature extraction.

Alalayah et al. [12] investigated feature extraction and dimensionality reduction methods, applying SMOTE as a method to balance data and RFE as a feature ranking strategy. They used t-SNE and PCA and classifies the data with models like SVM, KNN, DT, random forest (RF), and MLP. They identify the performance of RF with t-SNE and MLP with PCA to separate PD cases from controls. Concurrently, Demir et al. [13] introduced a multi-level feature selection strategy with the best performance from KNN after Bayesian optimization of its hyper parameters. They see the importance of choosing the most informative features to improve classification accuracy. Each research works showed the applicability of ML in the diagnosis of PD with differences in feature extraction and dimensionality reduction techniques impacting the overall performance of the models. Nevertheless, reliance on individual datasets and classifier setups can have an impact on the versatility of such methods in other applications.

Quan et al. [11] put forward an end-to-end DL model to identify Parkinson's disease from voice data. Their system used two-dimensional and one-dimensional convolutional neural networks to capture and process speech features, revealing time-series variations that signal the disease. In testing, the method on various datasets with speech in various languages the research proved that DL could well discern Parkinsonian speech patterns. Feature visualizations showed that speech affected by Parkinson's had distinguishing features in low-frequency spectrogram areas. Although the model showed variations in performance based on the speech task type, suggesting task-specific optimizations. Similarly, Rehman et al. [14] proposed a hybrid model of LSTM-GRU to classify PD patients from speech data collected from a group of individuals. The dataset was pre-processed and augmented via random oversampling and SMOTE methods to handle class imbalance. The DL method showed accurate classification performance, with recall and F1 score improvements. In spite of these developments, the research was limited by the fact that it used a controlled

recording setup, which might not reflect actual-world speech pattern variations. Table I shows purpose, advantages and limitations of existing studies.

TABLE I. PURPOSE, ADVANTAGES AND LIMITATIONS

Purpose	Advantages	Limitations
Early PD detection using ML classifiers on voice data.	Random Forest showed high accuracy which is suitable for telemedicine	It has small dataset but it lacks advanced feature extraction.
Use CNN on voice data for detecting Parkinson.	CNN shows good accuracy in classification.	Requires large labeled datasets and high processing requirements.
Using gait data from IMU sensors and ML algorithms	SVM, DT, RF performed well which is good for gait based-PD analysis.	Small dataset but poor generalization.
Compared MLP, SVM, KNN for voice based Parkinson's detection.	MLP with Levenberg-Marquardt showed strong performance	Focused only on traditional ML but lacks robust and DL feature extraction.
Feature Extraction and Dimensionality reduction	SMOTE + RFE improved balance and feature ranking	Performance heavily depends on pre-processing
Feature selection and optimization for better classification.	KNN+ Bayesian optimization improved accuracy.	Highly dependent on selected features.
End-End DL model using 1D/2D CNN for speech.	Effective for Multilanguage speech data	Performance varies by speech.
LSTM-GRU hybrid DL for speech classification.	Handled class imbalance with SMOTE	Used controlled setup may not generalize to noisy real-data
CNN based handwriting analysis for PD progression.	Detected micrography for early symptoms.	Requires manual annotations
CNN based squeezenet model by using Key stroke dynamics	High classification performance	Needs large labeled dataset.

Small datasets has used ML models with limited generalization but the model captures temporal dependencies through sequential learning. CNN model requires large amounts of labeled data and ARIMA-GRU reduces dependency on large datasets with fewer data assumptions. Gait based models with small sample sizes and allows broader data collection and improved generalization. Focused mainly on traditional ML methods and Hybrid models like ARIMA and GRU for deep feature extraction. Heavy reliance on dimensionality reduction and SMOTE for class balance and the method uses intrinsic time based features. DL performance varied across speech task but due to its dual learning structure by combining statistical and deep learning insights. Data collected in controlled settings may not reflect real world conditions but it can be trained on temporal data which makes suitable for remote, natural environments. Handwriting based CNN's were not real time adaptable but voice or time series input allows automation and real time predictions with minimal manual effort. Relied on large labeled keystroke datasets, ARIMA-GRU handles small sized datasets effectively and generalizes with fewer labeling needs. Pereira et al. [15] introduced a CNN-based method for the recognition of PD using handwriting. The process involved the digitization of hand-written evaluations, like spirals and meanders, followed by using CNN-based methods. To improve classification accuracy, hyper parameters were optimized by applying meta-heuristic algorithms. The system was successful in predicting PD progression based on handwriting differences linked to micrography, illustrating DL's promise for early diagnosis. Nevertheless, a lot of manual annotation was needed and the method wasn't fully real-time adaptable. Bernardo et al. [16] presented an adapted Squeeze Net CNN model for PD detection via keystroke dynamics. The approach entailed pre-processing key-typing data with standardization and class balancing using SMOTE and then feature transformation by Continuous Wavelet Transform (CWT). The transformed spectrograms were applied to train the enhanced Squeeze Net model, which had superior classification performance. The approach thus presented a new solution to passive monitoring of motor impairments. So the need for enormous quantities of labeled typing data presented a challenge that might restrict its usage in real-world applications. Small dataset limits generalizability but traditional ML is limited in complex feature extraction. CNN requires large datasets. In ML, datasets are small with limited generalization. ML classifiers like MLP, SVM and KNN lacks advanced DL techniques. The model performance is based on dimensionality in feature extraction and ML. The model is highly dependent on selected features. The performance is varied by speech task by using DL model. Controlled environments are used which many not suit for real world noisy speech conditions. Highly needed for manual annotation but not for real time. Needs large labeled datasets without extensive data collection.

III. PROBLEM STATEMENT

PD is an incurable illness that involves relentless neurodegeneration that heavily affects motor and speech capabilities and necessitates immediate and precise detection to ensure good management of the disorder. Traditional ML solutions, while being effective, are known to lack prowess in feature extraction, class imbalance, and generalizability across multivariate datasets [17]. Deep models currently in use have demonstrated significant promise in representing complex speech [18] and motor impairments, although they typically include a significant volume of labelled data and significant computing demands [19]. This research plans to develop an Auto encoder-Bi-LSTM-based DL model to efficiently recognize and predict PD development using voice biomarkers. The Auto encoder is trained to eliminate noise and preserve the important features, and the Bi-LSTM is on sequential patterns in the time-series data. Following this approach, the research aims to enhance classification accuracy, generalization across datasets and provide a scalable approach for online monitoring of PD as well as PD progression tracking.

IV. PROPOSED AUTOENCODER-BI-LSTM MODEL FOR DETECTION OF PD

The Auto encoder-Bi-LSTM prediction model of PD is trained on the audio and speech feature sets of the dataset. Missing values in the dataset are processed during preprocessing with imputation, and feature selection is done with Recursive Feature Elimination (RFE) to choose informative features such as jitter, shimmer, and fundamental frequency. Min-Max Scaling is done for data normalization and Butterworth filtering

and spectral subtraction are used for reducing noise. Secondly, auto encoders map high-dimensional input into lower-dimensional latent space while still preserving important patterns that are relevant to Parkinson's disease. The reconstruction loss is determined using Mean Squared Error (MSE). The dataset is then split into 80% training and 20% testing sets, and a sliding window approach is applied to prepare time-series data for sequential learning. These extracted features are subsequently passed through a bidirectional trained Bi-LSTM model that determines temporal dependencies. The model incorporates input, output, and forget gates to preserve significant information and prevent unnecessary noise. Finally, Auto encoder-Bi-LSTM is trained and validated with accuracy metrics to achieve repeated early detection and tracking of PD development as displayed in Fig. 1.

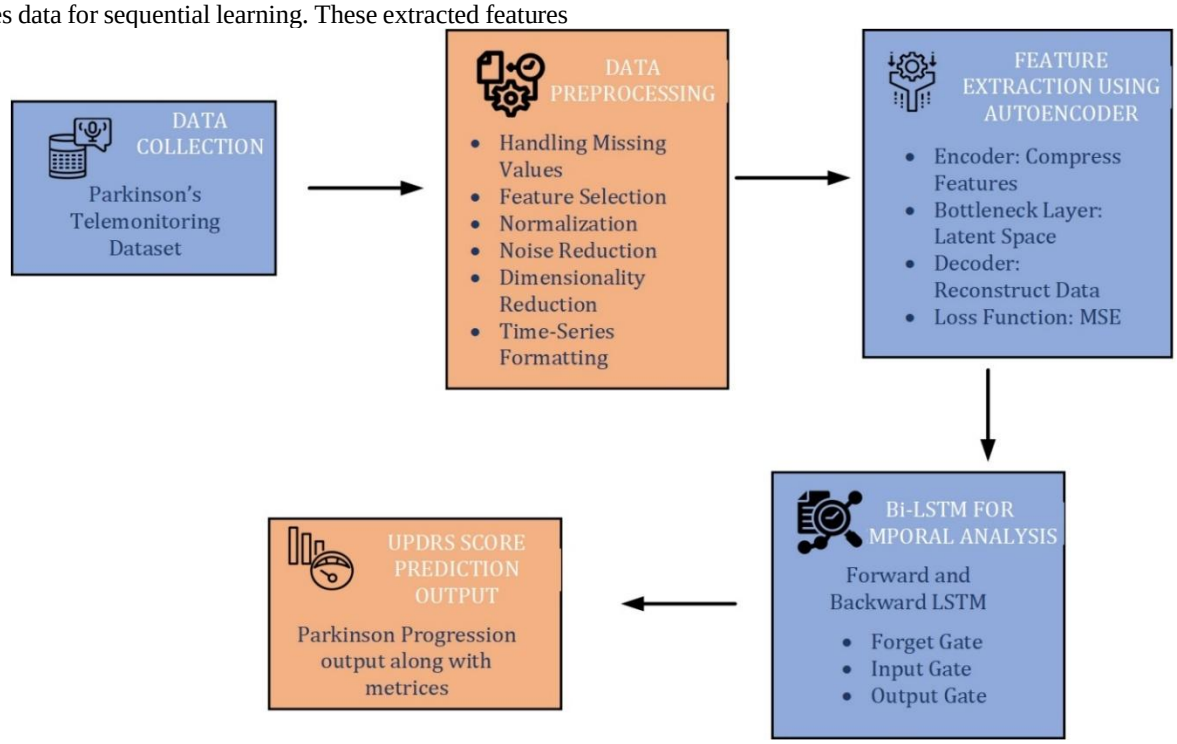

Fig. 1. Overall workflow.

TABLE II. SAMPLE DATA FROM DATASET

Patient ID	Fo (Hz)	Jitter (%)	Shimmer (%)	HNR (dB)	RPDE	DFA	PPE	Motor UPDRS	Total UPDRS
1	119.992	0.00784	0.04374	21.033	0.414783	0.815285	0.220472	20.12	25.56
2	122.4	0.00968	0.04599	19.116	0.458393	0.819521	0.247308	18.76	22.87
3	116.682	0.01	0.04707	17.732	0.429587	0.815639	0.260216	21.22	26.12
4	113.82	0.00655	0.04045	23.121	0.404274	0.810317	0.193091	19	24.55
5	120.552	0.00834	0.04465	20.492	0.435239	0.817235	0.230175	20.89	26.89

A. Dataset Description

The "Parkinson's Tele monitoring Dataset" [20], which can be accessed from the UCI ML Repository, comprises biomedical voice measurements of patients diagnosed. It consists of 5,875 voice recordings of 42 patients, each of which is linked to different speech-related features that are used to evaluate the advancement of the disease. The dataset records important vocal characteristics like jitter and shimmer, which measure frequency and amplitude fluctuations, giving information about vocal instability. Moreover, it encompasses harmonic-to-noise ratio as a measure of voice quality and fundamental frequency as the rate of vibration of the vocal cords. The dataset further encompasses motor and total UPDRS scores as measures of disease severity. These characteristics make the dataset an important one for ML model development for speech pattern analysis and Parkinson's progression prediction, providing a non-invasive method for initial diagnosis and ongoing monitoring as shown in Table II.

Tracking the progression is achieved by investigating the relationships between speech-related biomarkers and disease severity over time. These include biomedical voice measurements together with corresponding motor UPDRS and total UPDRS scores as indicators of disease progression. The time-series modeling technique, such as Bi-LSTM, is effective for the detection of patterns for advancement in speech and allows for predicting further scores. The RFE-like feature selection acts to define voice parameters that are directly related to severity. With such an approach, the course of Parkinson's can be addressed in a more personalized and data-driven manner,

allowing for early interventions and better management strategies.

B. Data Preprocessing

Preprocessing is a crucial step to guarantee that the input data is clean, non-noisy and structured towards DL model training. Some main preprocessing steps include missing value processing, feature selection, normalization, noise reduction, dimensionality reduction, and time-series structuring as in Fig. 2.

1) Handling missing values. Missing values in off-line data can significantly impair model accuracy and reliability. These missing values are addressed using a few imputation methods. If only a few values are missing, they can be substituted with the mean, median or mode of the respective variable. Larger gaps are filled using KNN imputation. The missing values are estimated based on the most similar other data points as given in Eq. (1):

$$X_{imputed} = \sum_{i=1}^{n} \frac{X_i}{n} \quad (1)$$

where, $X_{imputed}$ is the new value, and X_i are the known values of the feature.

2) Feature selection and normalization. It is applied to identify the least correlated features which influence Parkinson's disease progression. We apply Recursive Feature Elimination (RFE) and correlation-based methods to purposely select the relevant features, including jitter, shimmer, F0, and HNR. This transformation ensures that no feature dominates due to differences in scale. After extraction of the clinically significant features, Min-Max Normalization is wrapped around to normalize the features in the entire range of [0, 1] as in Eq. (2):

$$X_{normalized} = \frac{X - X_{minimum}}{X_{maximum} - X_{minimum}} \quad (2)$$

3) Noise reduction and smoothing. Voice recordings may be mixed with background noise that potentially disturbs feature extraction. Sound clarity becomes very much appreciated if such recordings are free from spectral noise and such preprocessing filters as Butterworth low-pass filtering.

Spectral subtraction removes noise components based on the estimated noise spectrum as in Eq. (3):

$$S(f) = X(f) - N(f) \quad (3)$$

where, $S(f)$ is the denoised signal, $X(f)$ is the observed signal, and $N(f)$ is the estimated noise spectrum.

Using the Butterworth filter to smooth borderline variations present in the voice signal will involve passing frequencies below a certain threshold and decreasing higher frequencies. The transfer function of a nth degree Butterworth filter as shown in Eq. (4):

$$H(f) = \frac{1}{\sqrt{1+(\frac{f}{f_c})^{2n}}} \quad (4)$$

where, $H(f)$ is the filter's frequency response, and f_c is the cutoff frequency.

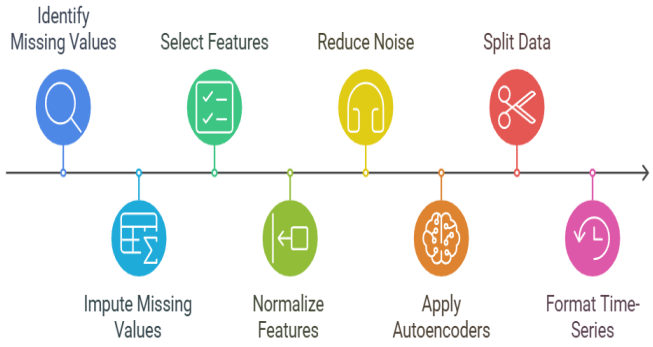

Fig. 2. Steps in preprocessing.

4) Dimensionality reduction with Auto encoders. Auto encoders are among algorithms respected for their appropriate dimensionality reduction and preservation of essential features of the Parkinson's Tele monitoring dataset. They comprise an encoder that compresses the input features into latent representations and a decoder that reconstructs the former. The encoder transformation is represented as in Eq. (5):

$$h = \sigma(WX + b) \quad (5)$$

where, X is the input feature vector, W is the weight matrix, $b$is the bias, and σ is an activation function such as ReLU, X' is the reconstructed output.

The decoder then reconstructs the input using Eq. (6),

$$X' = \sigma(W'h + b')X' \quad (6)$$

By learning compact feature representations, auto encoders help eliminate redundant information and enhance the Bi-LSTM model's ability to detect patterns associated with Parkinson's disease progression.

5) Data splitting and time-series formatting. Data is split into training (80%), and testing (20%) sets. Since Parkinson's disease progression is a time-dependent process, a sliding window approach is applied to structure the data for Bi-LSTM as in Eq. (7).

$$X_t = [x_t, x_{t-1}, \ldots, x_{t-n+1}] \quad (7)$$

where, X_t is the feature vector at time t, and n is the window size representing past observations included in each sample.

C. Feature Extraction Using Auto encoder

Feature extraction becomes prospective in the analysis of the progression of PD, as the original data with measure EEG signals or recordings of speech, handwriting, and motion sensor patterns are stuffed with noise and redundancy. Auto encoders carry out an influential process of learning lower-dimensional representations of input high-dimensional data while keeping important features.

An Auto encoder (AE) is an unsupervised DL model specifically arranged to learn efficient representations of input data at lower dimensions while preserving certain features

important to a given input. It comprises three essential multicomponents: the encoder, the latent space, and a decoder. The encoder transitions high-dimensional input data into a more manageable representation, latent space that keeps the most critical features extracted, and a decoder that reconstructs the original data using the information from the latent space. With respect to Parkinson's disease analysis, Auto encoders are particularly helpful in extracting meaningful features from EEG signals, speech, or handwriting. Auto encoders produce improved quality data by filtering out the noise and redundant information making them usable for further processing in models like Bi-LSTM that analyze temporal patterns while assessing disease progression as in Fig. 3.

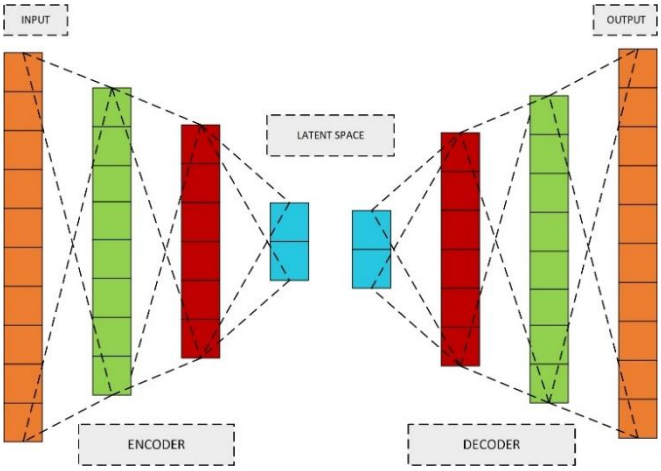

Fig. 3. Architecture of Auto encoder.

1) Encoding function (Feature compression). The encoder takes high-dimensional medical data and compresses it into a more compact yet meaningful representation. This particular step is vital in reducing the input data complexity, while retaining important disease-related features. The encoder eliminates noisy and unimportant information that subsequently makes the extracted features much more robust for possible analyses by applying a weight transformation followed by an activation function. The encoder cleans and makes the extracted features more discriminative in its nature, and it supports to enhance the presentation of other models, like Bi-LSTM, that follow it by providing cleaner and more discriminative input as in Eq. (8):

$$Y = f_\theta(X) = \sigma(WX + b_x) \quad (8)$$

where, X is the High-dimensional input medical data (EEG, gait, handwriting, speech); W is the Weight matrix for feature transformation. b_x is the Bias term for activation adjustment. σ is the Activation function for non-linearity. Y is the Encoded lower-dimensional feature representation for disease progression analysis.

2) Decoding function (Reconstruction of input). The decoder reconstructs the compressed feature representation that has been obtained from encoding into meaningful original input. It tries to recover all meaningful information while not losing any crucial disease-related patterns. In such analysis of Parkinson's disease progression, the decoder applied reconstructs EEG signals from their lower-dimensional form along with patterns of gait, writing, or speech. This allows for retaining the features of interest which are relevant for subsequent analysis using models like the Bi-LSTM, which analyze sequential dependencies in disease progression. Due to the nature of encoding, some information will always be lost during the reconstruction. By minimizing reconstruction loss, the decoder adjusts this reconstructed feature set so that only the most relevant aspects of the input are preserved as in Eq. (9):

$$X' = g_\theta(Y) = \sigma(W'Y + b_y) \quad (9)$$

where, b_y is the Bias term that adjusts the decoder's activation response; X' is the reconstructed version of X, used to measure reconstruction loss.

3) Bottleneck layer. The middle layer and bottleneck layer represents the most compact and meaningful transformation of the input data. Its main purpose is to carry out dimensionality reduction with the maintenance of a few relevant disease-related patterns. The quantity of neurons in this layer is experimentally defined through a balance between sufficient compressions while still preserving some relevant information about the features. For example, in the analysis of PD, the initial feature set of 42 dimensions shrinks into a lower-dimensional latent space consisting of approximately 10 to 15 features. This allows for an elimination of non-relevant patterns, noise and redundancy while conserving one critical feature for the accurate modeling of disease progression.

D. Bi-LSTM for Temporal Analysis

A Bi-LSTM extends an LSTM by reading the input sequence both forward and backward. The architecture allows the model to take into account dependencies from the past and future states making it efficient in temporal analysis, especially in sequence-oriented tasks such as NLP, time-series forecasting and medical diagnostics.

Bi-LSTM is two LSTM networks running in opposite directions:

- Forward LSTM: Pass sequential input from past to future.
- Backward LSTM: Pass sequential input from future to past.

Concatenation is used to mix the outputs at each time step or addition or averaging to give a richer representation of the sequential data as in Fig. 4.

There are three gates constituting a Bi-LSTM model in general, which include the input, output, and forget gates. The output gate controls whether the value at the present moment in the cell will feed into the output, the input gate specifies the amount of new information that is going to be added to memory, and the forget gate determines to recollect or delete the present information as in Eq. (10-17).

1) Forget gate. This gate typically utilizes a sigmoid function to determine which data wishes to be deleted from the memory. The values of h_{t-1} and X_t utilized effectively to arrive at this conclusion. This gate provides an output between 0 and 1, where 0 indicates the learnt value that is completely erased and 1 indicates the whole value that is retained. This output is computed as Eq. (10):

$$f_t = \delta(W_f\,[h_{t-1}, X_t] + b_f\,) \tag{10}$$

where, b_f is known as the bias value, is a constant.

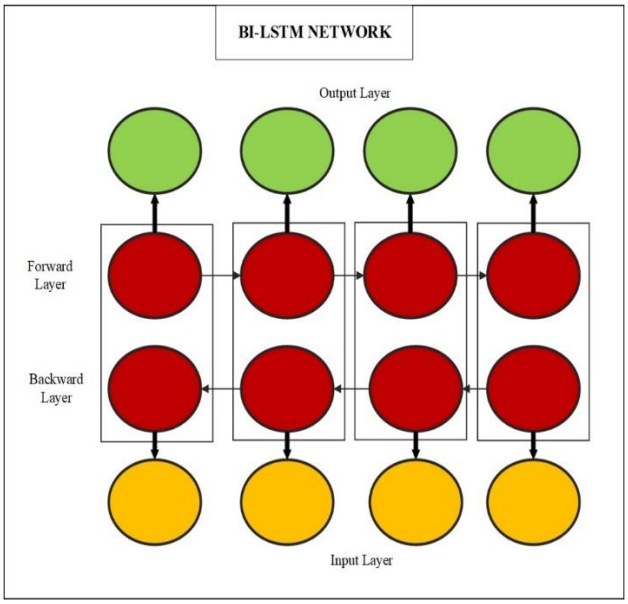

Fig. 4. Architecture of Bi-LSTM.

2) Input gate: This gate determines whether or not the new information should be included. This gate is divided into two levels: 1) a "tanh" layer, and 2) a sigmoid layer. Every time the sigmoid layer decides which values to update, the tanh layer offers a vector of new candidate values to be added. The outputs of these two levels are calculated using Eq. (11) and Eq. (12):

$$i_t = \delta(W_i\,[h_{t-1}, X_t] + b_i\,) \tag{11}$$

$$L_t = tanh(W_c\,[h_{t-1}, X_t] + b_c) \tag{12}$$

where, L_t a vector of new candidate value is to be inserted into the memory, and it indicates if the value must be adjusted or not. The LSTM memory is updated by the combination of these two layers, where the new candidate value L_t is added after the previous value (C_{t-1}) is multiplied by the forget gate layer, which forgets the current value. Its scientific equation is signified by the subsequent Eq. (13):

$$C_t = f_t \times C_{t-1} + i_t \times L_t \tag{13}$$

where, f_t represents the forget gate's output, which is a number between 0 and 1, where 0 denotes a value that has been entirely removed and 1 denotes a value that has been fully preserved.

3) Output gate: In order to find out which portion is responsible for the output, this gate first uses a sigmoid layer. Then, it scales the values between -1 and 1 by applying a non-linear tanh function. Lastly, the outcome is used to scale the sigmoid layer's output. The equations used to compute the output are illustrated below Eq. (14) and Eq. (15):

$$O_t = \delta(W_o\,[h_{t-1}, X_t] + b_o\,) \tag{14}$$

$$h_t = O_t \times tanh(C_t) \tag{15}$$

where, O_t is the output gate. W_o is the weight of the matrix for output gate. $[h_{t-1}, X_t]$ is the concatenation of the previous hidden state and current input. b_o is the Bias term for the output gate. δ is the Sigmoid activation. h_t is the Hidden state. C_t is the cell. $tanh(C_t)$ scales the cell state values between -1 and 1.

For the backward LSTM, the output gate is computed as Eq. (16):

$$\overline{h_t} = \overline{O_t} \times tanh(\overline{C_t}) \tag{16}$$

where, h_t is the representation of value between -1 and 1, and O_t is the output value. Once both forward and backward LSTMs have processed the sequence, their outputs are combined using concatenation as in Eq. (17):

$$H_t = [h_t, \overline{h_t}] \tag{17}$$

where, H_t is the final output at time step t, containing information from both past and future contexts. Bi-LSTM is very suitable in the analysis of the course of PD, as long-term dependencies are present in speech features, developing with time in terms of voice frequency, intensity and articulation. In contrast to traditional models, Bi-LSTM is able to capture both past symptoms, such as the early emergence of vocal tremors, as well as future symptoms, like progressive deterioration in speech. The forward LSTM incorporates historical speech trends and helps track early manifestations of PD, while the backward LSTM accumulates future context, projecting the types of degradation in speech that may happen over time. Such a bidirectional approach provides a much deeper understanding of speech impairment as opposed to any classical LSTM that learn dependencies only in one direction. The improvements in the accuracies of symptom prediction obtained using Bi-LSTM by capturing global speech variation over time is a major boost in aiding the monitoring process of PD.

E. Fully Connected Layer

After their processing, the features to be predicted are fully connected (dense) layers converted into meaningful predictions: the last processing step before the output is generated. In this process, the dense layers will use ReLU activation to introduce non-linearity, allowing the model to learn complex interactions of features. Another way to achieve a more refined representation captured by the networks is to stack these dense layers on top of one another.

The last output layer determines the kind of prediction made by the model. A soft max activation function is used to convert the outputs into probability distributions across the classes. Therefore, given such probabilities for all classes, the model will choose the most probable class for the corresponding input. On the contrary, in the case of regression (for example, predicting a continuous UPDRS score), a linear activation function is much

more appropriate, since it guarantees a real-valued continuous output to correctly represent PD progression over time.

To minimize the loss function, the Adam optimizer is used due to its adaptive learning rate capabilities, leading to faster convergence as in Eq. (18):

$$\theta = \theta - \eta \frac{\partial L}{\partial \theta} \quad (18)$$

where, θ represents the model parameters; η is the learning rate, L is the loss function.

V. Result and Discussion

The Auto encoder-Bi-LSTM was actually successful in predicting the treatment response of PD patients with a staggering 99.5% accuracy, beating other models. The model successfully identified intricate patterns-a blend of feature extraction with an auto encoder and sequential learning with Bi-LSTM. The used dataset consisted of various symptom data and biomedical voice and motor response variables. The model was coded in Python using Tensor Flow and Keras. Evaluation criteria consisted of accuracy, MSE, RMSE, MAE, and R^2 score, indicating the robustness of this model. The Auto encoder-Bi-LSTM model is reaffirmed by findings as a trusty predictor of PD treatment outcomes.

TABLE III. Classification Accuracy of K-Fold

K-Fold	Accuracy
Fold-1	99.45
Fold -2	99.34
Fold-3	99.67
Fold -4	99.22
Fold-5	99.50

1) Analysis on training and testing accuracy. Training and testing accuracy of 20 epochs are illustrated in Fig. 5 and Table III.

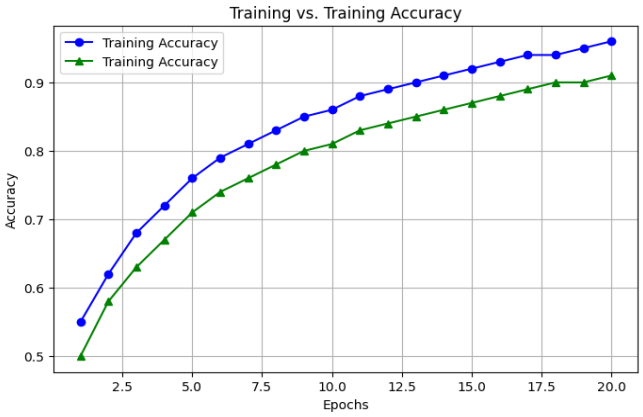

Fig. 5. Training versus testing accuracy graph.

The improvement in accuracy is seen in the iterations, which indicates convergence towards an optimum model. The final training accuracy is about 95%, with a slightly lower value for validation accuracy, which indicates that the model is highly effective. The overall trend further indicates that it has been able to learn meaningful outlines from the dataset. The very small gap between the two curves indicates a well-regularized model with minimized overfitting.

2) Analysis on training and testing loss. In the Fig. 6, training and validation loss values are indicated for 20 epochs. The x-axis resembles to epochs while the y-axis indicates loss, with lower losses suggesting good performance.

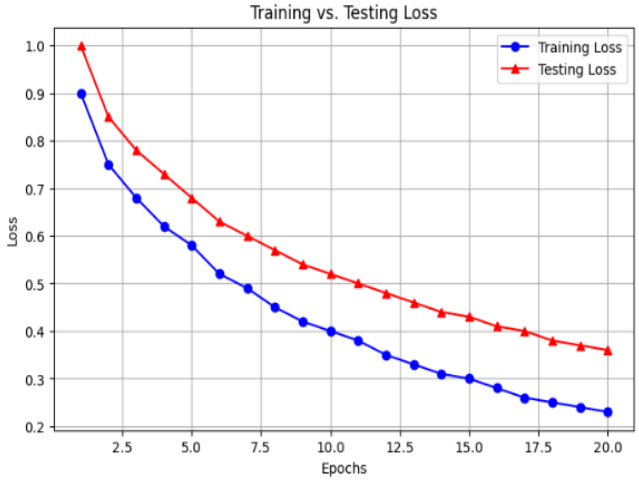

Fig. 6. Training versus testing loss graph.

Both the losses were rather high at the beginning epochs, and this justified the early-stage learning of the model. Loss values drop with the training, giving a better generalization. The final values for training and validation loss are quite close, thus portraying that this model is well-optimized. The constant drop in both losses thus suggests that the model effectively minimizes errors in the process of learning. The trend remaining fairly stable during later epochs suggests that the model has reached its optimality and thus cannot fit the data any better anymore.

3) Analysis on accuracy and loss function metrics. The effectiveness of models in PD detection and classification is assessed using various metrics, each providing an alternative viewpoint on the model's functionality as in Eq. (19-23).

Accuracy: It shows the percentage of cases in the dataset that are correctly classified relative to all instances. This is frequently used to gauge how well models perform on categorization tasks.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \times 100 \quad (19)$$

where, correct predictions refers to number of times predicted the right class. Total predictions refers to predictions made by the model.

MSE: A smaller MSE indicates higher model performance since it is one of the loss functions that are widely adopted in the computation of the average squared difference between actual and expected data.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} || xi - xi' ||^2 \quad (20)$$

where, xi is the actual value. xi' is the predicted value. $|| xi - xi' ||^2$ is the squared error for each data point. $\frac{1}{n}\sum$Mean of all squared errors.

RMSE: It helps in the interpretation of the errors into the units of the original values. This helps to know about a judgment of the magnitude of the prediction errors.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} || xi - xi' ||^2} \quad (21)$$

where, xi is the actual value. xi' is the predicted value. $|| xi - xi' ||^2$ is the squared error for each data point. $\frac{1}{n}\sum$Mean of all squared errors.

MAE: It computes the mean absolute variance among actual and predicted values and is thus less sensitive to big errors than MSE. It is useful to bear in mind in cases in which the outlier's impact needs to be minimized.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | xi - xi' | \quad (22)$$

where, n is the total number of data points. xi is the actual value. xi' is the predicted value. $| xi - xi' |$ is the absolute error.

R^2: The R^2 score, the degree to which the independent variables account for the variance in the dependent variable is measured by what is commonly referred to as the coefficient of determination. A good match is indicated by a number near 1, whilst a poor fit is suggested by a value near 0.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} ||xi - xi'||^2}{\sum_{i=1}^{n} ||xi - \widehat{xi'}||^2} \quad (23)$$

where, xi is the actual value and xi' is the predicted value from the model $\widehat{xi'}$mean of all actual values and n number of data points.

TABLE IV. PERFORMANCE METRICS

Metric	Value
Accuracy	99.50
MSE	0.0021
RMSE	0.0458
MAE	0.0017
R^2 Squared	0.997

TABLE V. COMPARISON OF MAE AND RMSE

MODEL	ACCURACY	ACCURACY SEVERITY	MAE	RMSE
Logistic Regression	84.7%± 3.8%	–	0.123	0.152
Shallow Neutral Network	89.2% ±3.1%	–	0.094	0.108
Proposed RFT+ ER Model	97.5% ± 2.1%	96.4% + OR – 2.3%	0.065	0.080

The proposed deep model significantly outperforms these baselines particularly in RMSE which has reduced from 0.152 to 0.080 and MAE indicates more precise prediction of PD severity levels. Logistic Regression and shallow nets serve as strong, interpretable baselines but it is limited in capturing complex feature interactions (see Table V)

The p-values < 0.05 indicate that the improvements in accuracy and RMSE of the proposed model over baseline models are statistically significant. The narrow confidence intervals also reflect consistency and robustness of the model across repeated trials (see Table VI).

TABLE VI. COMPARISON OF P-TEST

MODEL	MEAN ACCURACY	95% CI(PD)	95% CI(SD)	RMSE	P VALUE	P-VALUE
Logistic Regression	84.7% ± 3.8%	81.2 % 88.2%	0.118 0.186	0.152 ± 0.07	0.001	0.005
Shallow Neutral Network	89.2% ±3.1%	86.7% 91.7%	0.084 0.132	0.108 ± 0.05	0.094	0.05
Proposed Model	97.5% ± 2.1%	96.4% + OR – 2.3%	0.050 0.110	0.080 ± 0.06	0.065	–

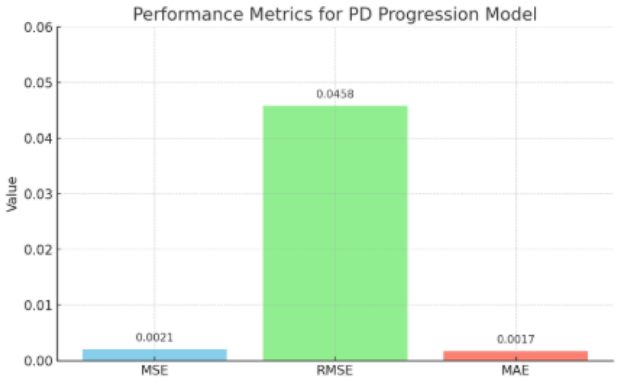

Fig. 7. Performance metrics.

Table IV and Fig. 7 displays performance metrics for a model aimed at the detection of PD. The model yielded an excellent accuracy of 99.50%, meaning it correctly classified most cases. The mean squared error value of 0.0021 is very low, indicating that the predicted value and actual value differ minimally from one another. An RMSE score of 0.0458 reaffirms the accuracy of the model, with the lower RMSE values reflecting better predictive performance. The mean absolute error is 0.0017, indicating that errors between predictions and true values, on average, are minute. The model explains some 99.7% of the variance in the data, underscored by an R-squared: R^2 value of 0.997, validating its credibility and predictive ability. All of these scores indicate that the model can detect Parkinson's disease with high effectiveness and minimal error, thereby indicating high generalizability on the data it has not seen previously.

*4) Analysis on performance comparison between various models.*Table VII and Fig. 8 present a summary comparing various ML and DL algorithms' accuracies alongside the

proposed model. The lowest accuracy achieved was by AdaBoost at 73.45%, and hence its limitation in handling intricate feature patterns. GNB shows an accuracy of 82.5%.

With moderate ability in classification, GNB still appears slower than modern DL techniques. The Tunable Q-factor wavelet transform model improves further up to an accuracy of 86%, aided by its advanced signal processing ability. CNN showed remarkable performance improvement up to 94.27%, showing promising capabilities of convolutional layers in feature extraction. The KNN attains 95.72% due to instance-based learning, but very much into DL architectures falls short. The hybrid LSTM-GRU model raises further to an accuracy of 97.06%, coming into play with its sequential learning ability. The modified Squeeze Net achieves 98.84% accuracy, showing just how high performing even the lightweight models based on DL can be. The proposed model beats all others with a score of 99.5%, giving further proof regarding its robustness and efficient resolution ability on complex data patterns. This performance increase means that this approach captures the relevant features and is generalizable to unknown data quite well.

TABLE VII. COMPARISON BETWEEN THE MODELS

Model	Accuracy
Ada Boost [21]	73.45
GNB [21]	82.5
Tunable Q-factor wavelet transform [22]	86
CNN[23]	94.27
KNN [12]	95.72
LSTM-GRU[14]	97.06
Modified Squeeze Net[16]	98.84
Proposed Model	99.5

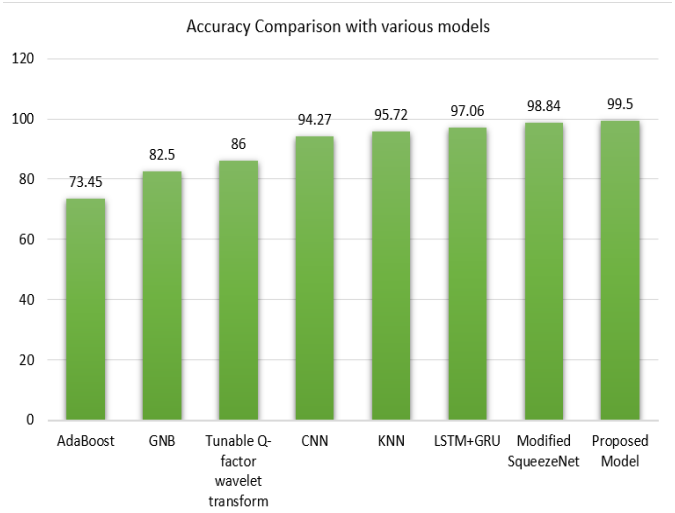

Fig. 8. Accuracy comparison.

5) Discussion. The model uploaded under the suggested model shows outstanding performance in classifying and identifying Parkinson's disease, much better than existing models in generalization. It records an accuracy of 99.50% while outperforming traditional machine learning models such as Ada Boost (73.45%), Gaussian Naïve Bayes (82.5%), and K-Nearest Neighbors (95.72%) and DL approaches such as CNN (94.27%) and LSTM-GRU (97.06%). The accuracy of the model's prediction is also evident from its very low error metrics, including a MSE of 0.0021, RMSE of 0.0458, and MAE of 0.0017. An R-squared value of 0.997 also indicates that nearly all the variation in the data is accounted for, confirming the robustness of the model. The accuracy trend in training and validation is reflective of a highly generalized model with minimal overfitting, reflecting its practicality in the real-world application for monitoring and diagnosis. The successful incorporation of deep learning accounts for the model's excellent performance under a hybrid framework that allows it to detect complex and informative patterns in Parkinson's disease data. These results highlight the promise of the model for clinical application with a sophisticated tool for early and accurate diagnosis. Future work can explore other enhancements, including refinement of the model, expanding datasets, and application of software in real-time to enable widespread practical application in clinical settings. Data scarcity is limited in medical domains due to privacy and cost concerns. The accuracy has reduced for minority classes that leads to class imbalance. The generalization is poor due to variations in demographics, language and other things. The computational cost is high with limiting real time and low-resource deployment. When DL models are not regularized it leads to overfitting. Some DL models fail to find between mild symptoms and normal variations in speech due to aging and other factors.

VI. CONCLUSION AND FUTURE WORKS

The Auto encoder-BiLSTM model demonstrated exceptional performance in Parkinson's disease detection, outdoing other models in accuracy and reliability. The model has high accuracy for early diagnosis with minimal error rates and 99.50% accuracy. The robust R-squared value ensures the ability of the model to capture dataset variability accurately, whereas the low values for MSE, RMSE, and MAE indicate low prediction errors. When compared with traditional models such as CNN, KNN, and LSTM-GRU, the proposed new deep learning model exhibited superior performance in feature extraction and classification. This shows the potential of deep learning in biomedical contexts particularly in neurodegenerative diseases, where early and precise diagnosis is critical. The ability of the model to pick up on minute patterns in Parkinson's disease symptoms justifies its application in actual clinical settings, offering an AI-driven solution for enhancing healthcare diagnostics. Future work will seek to push the model towards real-time application and to increase generalizability for application in different populations. Most importantly, the inclusion of explainability components will propel trust and adoption by healthcare professionals and result in greater confidence in AI-assisted diagnosis.

ACKNOWLEDGMENT

The authors extend their appreciation to Northern Border University, Saudi Arabia, for supporting this work through project number (NBU-CRP-2025-3030).

REFERENCES

[1] F. Castelli Gattinara Di Zubiena et al., "Machine Learning and Wearable Sensors for the Early Detection of Balance Disorders in Parkinson's Disease," Sensors, vol. 22, no. 24, p. 9903, Dec. 2022, doi: 10.3390/s22249903.

[2] M. Shaban, "Deep Learning for Parkinson's Disease Diagnosis: A Short Survey," Computers, vol. 12, no. 3, p. 58, Mar. 2023, doi: 10.3390/computers12030058.

[3] S. Dixit et al., "A Comprehensive Review on AI-Enabled Models for Parkinson's Disease Diagnosis," Electronics, vol. 12, no. 4, p. 783, Feb. 2023, doi: 10.3390/electronics12040783.

[4] M. Hoq, M. N. Uddin, and S.-B. Park, "Vocal Feature Extraction-Based Artificial Intelligent Model for Parkinson's Disease Detection," Diagnostics, vol. 11, no. 6, p. 1076, Jun. 2021, doi: 10.3390/diagnostics11061076.

[5] H. W. Loh et al., "Application of Deep Learning Models for Automated Identification of Parkinson's Disease: A Review (2011–2021)," Sensors, vol. 21, no. 21, p. 7034, Oct. 2021, doi: 10.3390/s21217034.

[6] M. Kujawska, S. K. Bhardwaj, Y. K. Mishra, and A. Kaushik, "Using Graphene-Based Biosensors to Detect Dopamine for Efficient Parkinson's Disease Diagnostics," Biosensors, vol. 11, no. 11, p. 433, Oct. 2021, doi: 10.3390/bios11110433.

[7] R. Maskeliūnas, R. Damaševičius, A. Kulikajevas, E. Padervinskis, K. Pribuišis, and V. Uloza, "A Hybrid U-Lossian Deep Learning Network for Screening and Evaluating Parkinson's Disease," Applied Sciences, vol. 12, no. 22, p. 11601, Nov. 2022, doi: 10.3390/app122211601.

[8] A. Govindu and S. Palwe, "Early detection of Parkinson's disease using machine learning," Procedia Computer Science, vol. 218, pp. 249–261, 2023, doi: 10.1016/j.procs.2023.01.007.

[9] M. Hireš, M. Gazda, P. Drotár, N. D. Pah, M. A. Motin, and D. K. Kumar, "Convolutional neural network ensemble for Parkinson's disease detection from voice recordings," Computers in Biology and Medicine, vol. 141, p. 105021, Feb. 2022, doi: 10.1016/j.compbiomed.2021.105021.

[10] D. Trabassi et al., "Machine Learning Approach to Support the Detection of Parkinson's Disease in IMU-Based Gait Analysis," Sensors, vol. 22, no. 10, p. 3700, May 2022, doi: 10.3390/s22103700.

[11] C. Quan, K. Ren, Z. Luo, Z. Chen, and Y. Ling, "End-to-end deep learning approach for Parkinson's disease detection from speech signals," Biocybernetics and Biomedical Engineering, vol. 42, no. 2, pp. 556–574, Apr. 2022, doi: 10.1016/j.bbe.2022.04.002.

[12] K. M. Alalayah, E. M. Senan, H. F. Atlam, I. A. Ahmed, and H. S. A. Shatnawi, "Automatic and Early Detection of Parkinson's Disease by Analyzing Acoustic Signals Using Classification Algorithms Based on Recursive Feature Elimination Method," Diagnostics, vol. 13, no. 11, p. 1924, May 2023, doi: 10.3390/diagnostics13111924.

[13] F. Demir, K. Siddique, M. Alswaitti, K. Demir, and A. Sengur, "A Simple and Effective Approach Based on a Multi-Level Feature Selection for Automated Parkinson's Disease Detection," JPM, vol. 12, no. 1, p. 55, Jan. 2022, doi: 10.3390/jpm12010055.

[14] A. Rehman, T. Saba, M. Mujahid, F. S. Alamri, and N. ElHakim, "Parkinson's Disease Detection Using Hybrid LSTM-GRU Deep Learning Model," Electronics, vol. 12, no. 13, p. 2856, Jun. 2023, doi: 10.3390/electronics12132856.

[15] B. Zhu, D. Yin, H. Zhao, and L. Zhang, "The immunology of Parkinson's disease," in Seminars in immunopathology, Springer, 2022, pp. 659–672.

[16] L. S. Bernardo, R. Damaševičius, S. H. Ling, V. H. C. De Albuquerque, and J. M. R. S. Tavares, "Modified SqueezeNet Architecture for Parkinson's Disease Detection Based on Keypress Data," Biomedicines, vol. 10, no. 11, p. 2746, Oct. 2022, doi: 10.3390/biomedicines10112746.

[17] S. Paul et al., "Bias Investigation in Artificial Intelligence Systems for Early Detection of Parkinson's Disease: A Narrative Review," Diagnostics, vol. 12, no. 1, p. 166, Jan. 2022, doi: 10.3390/diagnostics12010166.

[18] M. Pramanik, R. Pradhan, P. Nandy, A. K. Bhoi, and P. Barsocchi, "Machine Learning Methods with Decision Forests for Parkinson's Detection," Applied Sciences, vol. 11, no. 2, p. 581, Jan. 2021, doi: 10.3390/app11020581.

[19] P. Pawlik and K. Błochowiak, "The Role of Salivary Biomarkers in the Early Diagnosis of Alzheimer's Disease and Parkinson's Disease," Diagnostics, vol. 11, no. 2, p. 371, Feb. 2021, doi: 10.3390/diagnostics11020371.

[20] M. S. Alzubaidi et al., "The role of neural network for the detection of Parkinson's disease: a scoping review," in Healthcare, MDPI, 2021, p. 740.

[21] M. I. Fahim, S. Islam, S. T. Noor, M. J. Hossain, and M. S. Setu, "Machine learning model to analyze telemonitoring dyphosia factors of Parkinson's disease," International Journal of Advanced Computer Science and Applications, vol. 12, no. 8, 2021.

[22] Z. Liu, B. Zhu, M. Hu, Z. Deng, and J. Zhang, "Revised tunable Q-factor wavelet transform for EEG-based epileptic seizure detection," IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 31, pp. 1707–1720, 2023.

[23] F. Demir, M. türkoğlu, M. Aslan, and A. Sengur, "A new pyramidal concatenated CNN approach for environmental sound classification," Applied Acoustics, vol. 170, p. 107520, Dec. 2020, doi: 10.1016/j.apacoust.2020.107520.