2023 15th International Conference on COMmunication Systems & NETworkS (COMSNETS) | 978-1-6654-7706-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/COMSNETS56262.2023.10041413

A Smartphone-based Application to Detect Parkinson's Disease Using Audio

Saumya Mathkar Prakhar Karsh Udit Baluja Surjya Ghosh Sougata Sen Vinayak Naik
BITS Pilani, Goa, INDIA
{p20210017, h20210043, h20210034, surjyag, sougatas, vinayak}@goa.bits-pilani.ac.in

***Abstract*—Parkinson's disease (PD), a commonly occurring neurodegenerative disease, affects millions worldwide. One approach to detecting PD is observing variations in an individual's speech patterns, such as tone, jitter, shimmer, and pitch. In this demo, we present PidiBuddy, a smartphone-based system that detects PD based on the user's voice data. To reduce privacy concerns and dependency on background infrastructure and facilitate usage by naive users, PidiBuddy runs end-to-end on the smartphone. It collects short speech segments, extracts features, and infers PD (each step happens in situ) using a Random Forest-based machine learning model. Before deploying the model on the device, we trained the model offline using a publicly available speech dataset comprising a set of MFCC (Mel-Frequency Cepstral Coefficients) related speech features. The initial findings from the system are promising in terms of PD detection performance, system parameters, and system usability, all of which we aim to improve further in our ongoing work.**

I. Introduction

Parkinson's disease (PD) is a brain disorder that leads to stiffness, shaking, and difficulty with walking, balance, and coordination [1]. PD symptoms gradually worsen with time leading to difficulty in walking and talking and overall poor quality of life. Therefore, detecting PD in its initial stages is crucial [2].

In the existing literature, manifestations of PD symptoms in the *speech signals* have been leveraged widely to design PD detection systems [3]. This has been primarily motivated by the following reasons: First, speech-based detection is non-invasive, cost-effective, and time-efficient as it reduces frequent visits to the practitioner [1], [4]. Second, the commonness of speech disorders in diseased patients has been reported to be as high as 89% at the prodromal stages, thus underscoring a common speech pattern among PD patients [2]. Speech or voice quality may become soft or slurred in the early stages of PD, along with other symptoms like tremors and slowed movement.

Finally, speech symptoms, such as reduced tongue flexibility, voice intensity, reduced pitch range, reduction in vocal tract volume, and weakening of voice quality, vary significantly between PD patients and normal users, which have been leveraged by machine learning models for screening PD in the existing systems [4]. However, one major limitation of existing systems are that for inferring the PD, they often rely on backend servers or the cloud; the systems send the speech segment to these servers, which also raises privacy concerns [5].

We, in this paper, design and develop PidiBuddy, a smartphone-based system that performs PD screening from a short speech segment on the device itself. PidiBuddy collects a 5-second speech sample from the user, extracts a set of speech-related features, and infers the PD using a Random Forest-based machine learning model on the device. As a result, it reduces the dependency on background infrastructure and privacy concerns. The machine learning model deployed on the device is trained offline with a large publicly available dataset for superior performance [2]. The model is then optimized to make it suitable for smartphone execution.

II. System Architecture and Implementation

The system consists of four primary components, as shown in Figure 1 – (a) speech acquisition module, (b) data processing module, (c) feature extracting module, and (d) inference engine.

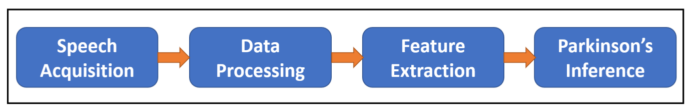


Fig. 1: PidiBuddy architecture. The speech signals are acquired using the in-built smartphone microphone, which we process for feature extraction. A set of MFCC-related features are extracted and fed to a Random Forest model for inferring PD (on-device).

Speech Acquisition This module captures the voice from the user. It records a 5-seconds audio clip and stores the amplitude values of the speech signal. We sample the speech signal at 16 kHz. We used this sampling rate as recommended in the earlier works for efficient feature extraction [6].

Data Processing The collected audio signal is passed through a band-pass filter of 16 kHz to remove the noise artifact, especially the ambient noise. We decided to use this filter as earlier works suggested the use of this filter for noise removal [7].

Feature Extraction We extract a set of MFCC (Mel-frequency cepstral coefficients) features from the speech signal [8]. MFCC feature extraction procedure applies the DFT (Discrete Fourier Transform) on windowed signal, performs the log of the magnitude, and then warps the frequencies on a Mel scale, followed by the IDFT (Inverse DFT) [9]. We perform feature extraction on-device so that no data should move outside the phone, thus reducing user privacy concerns.

Inference Engine We use this set of features as input to run the inference engine that determines whether the user is exhibiting PD signs. The inference engine operates on a Random Forest classifier. Notably, we have tried different classification models (e.g., kNN, Decision Tree, SVM, and a Feed-forward Neural Network), but used the Random Forest model as it performed better than the other models in terms of accuracy and F-score. We trained the model offline (outside the device) using a publicly available Parkinson's Disease dataset based on speech features [10]. This dataset includes 252 participants (188 PD patients and 64 healthy controls) aged 33 to 87 years. Various speech signal processing algorithms, including time-frequency features and MFCCs were applied to find a total of 756 features, which we reduced to 168 using Principal Component Analysis (PCA) algorithm.

A. System Implementation

We implemented PidiBuddy as an Android application. The in-built microphone of the smartphone records the voice, which is stored on-device and fed to the data processing module. Post-data processing, we use the MFCC library for feature extraction. We compare the accuracy and performance of multiple machine learning classifiers. The classifiers used for model evaluation are KNN, Decision Trees, Naive Bayes, and Neural Networks. We use a random forest model within the Android app as we get high accuracy for Decision Trees among all the other Machine learning models.

We provide extracted features to the Random Forest-based inference engine. However, we obtained the model by training it offline using the scikit-learn library on the given dataset [10]. The model consists of 100 trees. To avoid overfitting the model, we have set the max_features parameter to 84.

III. Evaluation

We evaluate PidiBuddy in terms of PD inference performance both offline and on-device. We also measure the system overhead for the latter. We train the model offline on a PC having a configuration of 11th Gen Intel Core i5-1135G7 with a clock frequency of 2.42 GHz and 16 GB RAM using the publicly available Parkinson's Disease dataset based on speech features. The model is extracted and integrated into the Android app as a .jar file. We deploy the Android device having Android version 9.

A. Offline Performance of the Model

We use accuracy and AUCROC as the performance metric to evaluate the model performance offline. We use AUCROC also as the metric as the dataset is imbalanced in terms of the distribution of PD subjects and healthy controls. We note the values of these metrics as obtained from different models in Table I. We observe that the Decision Tree-based model returns the best performance among all the models.

Model	Accuracy (%)	AUCROC (%)
Naïve Bayes	75.53	75.0
k-NN	81.35	79.0
MLP	82.41	83.0
Decision Tree	83.73	86.0

TABLE I: Comparison of different PD inference models

B. On-device Performance of the Model

Motivated by the earlier findings, we deploy the model on the device and measure the performance of the PD inference. However, as the Decision Tree-based model returns the best performance in offline mode, we deploy the Random Forest model on the device as a .jar file and measured the inference performance.

-	Predicted not PD	Predicted PD
Actual not PD	TN = 24	FP = 37
Actual PD	FN = 7	TP = 159

TABLE II: Confusion matrix corresponding to the on-device inference using Random Forest model. TP, FP, FN, TN indicate True Positive, False Positive, False Negative, and True Negative, respectively.

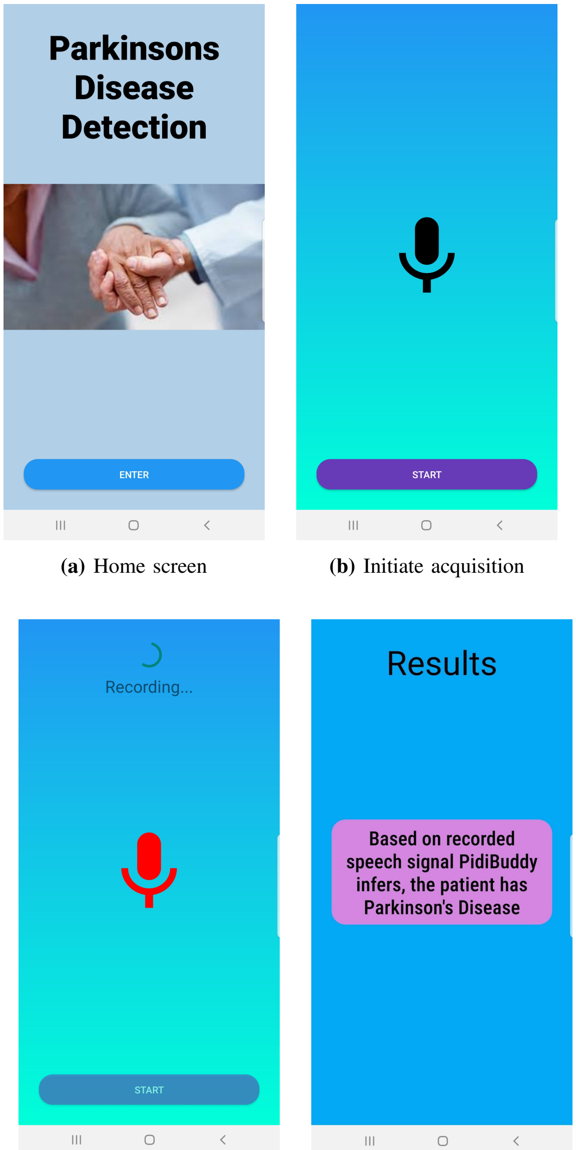
To evaluate the model performance, we randomly select a subset of the dataset and run the on-device inference. We obtained an accuracy of 80.6% and a high TPR (True positive rate) of 95.8%. We present the confusion matrix related to this on-device model based on the randomly selected dataset in Table II. These findings underscore the superior performance of the on-device model.

C. System Overhead on the Device

We measure different system parameters as PidiBuddy runs on the device. Specifically, we measure the *inference time* for a given set of feature values. We run the inferences 50 times and note an average inference time is $\approx$0.1482 milliseconds. We measure the *model size*, which is $\approx$712 KB signifying the model does not require a lot of space. We measured the *CPU utilization* and *main memory size* to find the model's efficiency for on-device execution. The CPU utilization ranges from 1% to 40%, and the main memory occupation varies from 80 MB to 190 MB. We also measure the *latency* while capturing the speech signal and the feature extraction. We estimate the average latency to be 627 milliseconds. In summary, all these system parameters suggest no significant resource consumption by PidiBuddy system, thus making it suitable for on-device execution.

IV. Description of the Demo

We demonstrate interaction with the PidiBuddy system in Fig. 2. On the home screen (Fig. 2a), the participant proceeds and initiates the speech acquisition module (Fig. 2b). The system records a speech for (5 sec). After recording the speech (Fig. 2c), we perform processing and feature extraction and the features are fed to the inference module. In the demonstration, to detect the positive results, we will replay a recording of the speech of patients. The inference module outputs the final result on screen (Fig. 2d), indicating whether the voice clip is from a Parkinson's patient.



(a) Home screen **(b)** Initiate acquisition

(c) Speech acquisition **(d)** Inference result

Fig. 2: End-to-end navigation for inferring Parkison's disease using the PidiBuddy system

V. Conclusion

Parkinson's disease is a slowly progressive disease. The average life expectancy of a patient with Parkinson's Disease is the same as that of a normal individual. However, it will differ from patient to patient. In the later stages, Parkinson's patients may not respond to medications. In some patients, it might lead to serious complications such as difficulty in walking, talking, pneumonia, choking, and overall poor quality of life. Therefore, the detection of PD in its initial stages is crucial.

We, in this paper, present a smartphone-based system PidiBuddy that leverages speech signals for inferring Parkinson's disease. The system records a user's short (5 sec) audio clip, extracts a set of MFCC-related features, and feeds the features to a Random Forest-based model, which classifies the patient as a PD patient (or not). The complete data processing, feature extraction, and inference module are performed on the device to deal with network issues and privacy concerns. Firstly, we implement machine learning models, such as Naive Bayes, KNN, MLP, and Decision Trees to check the accuracy in predicting whether a particular person has Parkinson's disease. We find that the accuracy is 75.53 %, 81.35%, 82.41%, and 83.73%, respectively. Here, the decision tree model gives the best accuracy compared with other models. In the next phase, we use a random forest-based model as the decision tree model yields high accuracy.

We aim to improve the model performance by considering additional modalities and overall system performance (e.g., latency, memory, and CPU utilization) in our future work.

VI. Acknowledgment

We thank National Mission on Interdisciplinary Cyber Physical Systems (NM-ICPS) implementation mechanism - Technology Innovation Hubs (TIHs) for their funding.

References

[1] N. Narendra, B. Schuller, and P. Alku, "The detection of parkinson's disease from speech using voice source information," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1925–1936, 2021.

[2] Z. K. Senturk, "Early diagnosis of parkinson's disease using machine learning algorithms," *Medical hypotheses*, vol. 138, p. 109603, 2020.

[3] S. S. S. Aarushi Agarwal, Spriha Chandrayan, "Prediction of parkinson's disease using speech signal with extreme learning machine," *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, vol. 16, pp. 3776–3779, 2016.

[4] T. Sriram, M. V. Rao, G. Narayana, D. Kaladhar, and T. P. R. Vital, "Intelligent parkinson disease prediction using machine learning algorithms," *Int. J. Eng. Innov. Technol*, vol. 3, pp. 212–215, 2013.

[5] Y. Zhang, "Can a smartphone diagnose parkinson disease? a deep neural network method and telediagnosis system implementation," *Parkinson's disease*, vol. 2017, 2017.

[6] S. Arora and A. Tsanas, "Assessing parkinson's disease at scale using telephone-recorded speech: insights from the parkinson's voice initiative," *Diagnostics*, vol. 11, no. 10, p. 1892, 2021.

[7] D. V. Rao, Y. Sucharitha, D. Venkatesh, K. Mahamth, and S. M. Yasin, "Diagnosis of parkinson's disease using principal component analysis and machine learning algorithms with vocal features," *International Conference on Sustainable Computing and Data Communication Systems*, pp. 200–206, 2022.

[8] S. Kalimuthukumar, G. Vagesvari, P. Thenmozhi, T. GirijaPragathi, and N. Vigneshwari, "Early- detection of parkinson's disease by patient voice modulation analysis through mfcc feature extraction technique," *3rd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, vol. 21, pp. 1045–1049, 2021.

[9] A. D. Pedroza Ramirez, J. I. de la Rosa Vargas, R. R. Valdez, and A. Becerra, "A comparative between mel frequency cepstral coefficients (mfcc) and inverse mel frequency cepstral coefficients (imfcc) features for an automatic bird species recognition system," *Latin American Conference on Computational Intelligence (LA-CCI) Guadalajara, Jalisco, Mexico*, vol. 18, pp. 7–9, 2018.

[10] C. O. Sakar, G. Serbes, A. Gunduz, H. C. Tunc, H. Nizam, B. E. Sakar, M. Tutuncu, T. Aydin, M. E. Isenkul, and H. Apaydin, "A comparative analysis of speech signal processing algorithms for parkinson's disease classification and the use of the tunable q-factor wavelet transform," *Applied Soft Computing*, vol. 74, pp. 255–263, 2019.