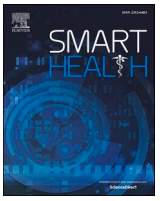
Smart Health 26 (2022) 100351

Contents lists available at ScienceDirect

Smart Health

journal homepage: www.elsevier.com/locate/smhl



Towards a telehealth infrastructure supported by machine learning on edge/fog for Parkinson's movement screening

Shehjar Sadhu [a,*,1], Dhaval Solanki [a,1], Nicholas Constant [a,1], Vignesh Ravichandran [a], Gozde Cay [a], Manob Jyoti Saikia [b], Umer Akbar [c], Kunal Mankodiya [a]

[a] Dept. of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, RI, USA
[b] Department of Electrical Engineering, University of North Florida, Jacksonville, FL, 32224, USA
[c] Department of Neurology, Brown University, Rhode Island Hospital, Providence, RI, USA

ARTICLE INFO

Keywords:
Edge computing
Fog computing
Machine learning
Internet of things
Wearable technology
Telehealth
Parkinson's disease

ABSTRACT

Approximately 10 million people worldwide live with Parkinson's disease (PD), a progressive and incurable neurological movement disorder. Symptomatic treatment is available for PD but requires patients to make periodic clinic visits (2–3 times per year) for symptom assessment. Advanced telehealth technologies can enhance clinical care for PD but warrant an Internet-of-Things-based (IoT) infrastructure that can enable symptom monitoring in out-of-clinic settings, such as homes. In this paper, we present an edge and Fog device-based IoT framework and a machine learning-based telehealth infrastructure that can detect and classify hand movement tasks based on a clinical test (UPDRS) for remote symptom assessment. We used a pair of smart gloves integrated with finger flex sensors, an inertial measurement unit (IMU), and a wireless embedded system (i.e., an edge device) to record the hand movements. The edge device (ESP32) detected the activity on the edge node and transmitted the data to the Fog node for classification. The Fog node (Raspberry pi) hosted the Machine Learning (ML) based activity classification models to classify UPDRS-based hand movement tasks. In this paper, we present the development of edge-fog-supported ML infrastructure. To develop ML models, we utilized the data of 9 participants (five healthy and four people with PD) who performed hand movements tasks while wearing the smart gloves. We developed and tested different classification models such as K-nearest neighbors (KNN), Support Vector Machine (SVM), and Decision Tree (DT) to identify which one fits best for such an edge-fog infrastructure for remote symptom assessment. Results showed that the SVM model outperformed others by giving 94% training, 94% testing, and 93% validation accuracy with a mean inference time of 560 μs on the Fog node. Our preliminary results are promising to further our research in deploying the edge-fog-driven ML-based telehealth infrastructure in real-world settings.

* Corresponding author.
E-mail address: shehjar_sadhu@uri.edu (S. Sadhu).
[1] Authors contributed equally.

https://doi.org/10.1016/j.smhl.2022.100351
Received 29 September 2022; Accepted 30 September 2022
Available online 8 October 2022
2352-6483/© 2022 Elsevier Inc. All rights reserved.

1. Introduction

Parkinson's Disease (PD) affects approximately 1 million people in the US and 10 million people globally (Naqvi, 2018). In the United States, every year, 60,000 new patients are diagnosed with PD (Naqvi, 2018). PD is a chronic, incurable, and progressive neurodegenerative disorder which can be treated symptomatically but requires close monitoring of a patient by a clinician, specifically a movement disorder specialist. Clinicians assess the patient's disease severity in their clinic approximately 3–4 times per year during patient's clinic visits (Tysnes & Storstein, 2017). These assessments are often limited to 15–20 min.

Clinicians often use standard clinical instruments such as the Movement Disorder Society's (MDS) Unified Parkinson's Disease Rating Scale (UPDRS) for such assessments. Clinicians often employ different therapies to manage the patient's symptoms, including medications, exercises, and brain stimulation. However, the administration and titration of these therapies depend on the patient's self-report(s) and the clinician's brief in-clinic assessment (Goetz et al., 2008). Such methods suffer from translational errors and reporting biases. Importantly, administration of the MDS-UPDRS is constrained by the availability of trained clinicians, while the progressive nature of this condition necessitates regular and frequent assessment.

Since individuals with PD are counted among the vulnerable demographic under the COVID-19 circumstances, telehealth video-based consultations have been used by clinicians for remote symptom assessment (Larson et al., 2021). However, during such telehealth consultation, fine details related to movements can be missed due to poor network connectivity and poor image quality from a web camera (Ferreira et al., 2021). In-home assessment systems can be used to enable individuals with PD to objectively monitor their movement symptom progression from the comfort of their homes (Salarian et al., 2007; Kim et al., 2011; Rigas et al., 2012; Sano et al., 2016; van den Noort et al., 2017). Such systems require real-time feedback when UPDRS movement tasks are performed in-home settings to ensure the right task is being performed in the absence of guidance and input from a clinician. Also, clinical test battery such as UPDRS often involves multiple movement-based exercises, which needs to be classified in-home settings. In this work, we present an edge-fog-based IoT framework consisting of a Machine Learning classifier to perform the detection and classification of hand movement tasks (selected from UPDRS). The contributions of the presented research are as follows:

- Development of Edge-Fog-supported telehealth IoT infrastructure.
- Development and assessment of Machine Learning classifiers for hand movement activity classification.
- Deployment and evaluation of the ML classification models on the Fog infrastructure.
- Demonstration of how an edge-fog computing architecture can be useful for telehealth applications, including movement assessment, symptom tracking, and symptom management.

2. Background & related works

Literature shows that several researchers have explored different technology-based solutions for remote movement assessment, as discussed below. Galna et al. and Chen et al. proposed non-contact imaging-based movement monitoring approaches to offer an evaluation modality for PD which does not require any sensors to be attached to the patients (Chen et al., 2021; Galna et al., 2014). Such an approach allows patients to perform free movements without any hindrance. However, such a method has many challenges and limitations in real-world settings, such as lighting, positional variability, privacy concerns, usability, and cost. To address these limitations, researchers have also explored Inertial Measurement Unit (IMU) based solutions along with machine-learning techniques to capture tremors, bradykinesia, motor fluctuations, and dyskinesia in patients with PD (Kim et al., 2011; Rigas et al., 2012; Salarian et al., 2007). Mera et al. proposed an in-home monitoring system (named Kinesia) using flex sensors to quantitatively monitor tremor and bradykinesia in PD patients (Mera et al., 2012). Besides IMUs, there are many sensors used by researchers to monitor the patient's movements and motor symptoms. Table 1 offers a comparison of different sensors used in various research studies for monitoring PD motor symptoms. Patel et al. developed a wearable body sensor network (BSN) and a web-based application that can be useful for gathering clinically relevant information from in-home settings (Patel et al., 2009). An SVM-based model was used to classify clinical scores for dyskinesia, bradykinesia, and tremor.

Further, Sano et al. investigated the use of magnetic field sensors to capture movement during the finger-tapping task and extracted several characteristic features to estimate symptom severity (Sano et al., 2016). These sensor data were used to estimate the UPDRS ratings. The accuracy of UPDRS rating classification achieved using this method ranged between 70 and 86% based on different classifiers. These research efforts are steppingstones to advance the development of modalities for remote PD symptom assessment and

Table 1
Related technologies targeted for PD.

Study/Approach	IMU	Magnetometer	Optical
Kim (Kim et al., 2011)	X	N/A	N/A
Sano (Sano et al., 2016)	N/A	X	N/A
Salarian (Salarian et al., 2007)	X	N/A	N/A
Rigas (Rigas et al., 2012)	X	N/A	N/A
Van den Noort (van den Noort et al., 2017)	X	N/A	N/A
Galana (Galna et al., 2014)	N/A	N/A	X
Chen (Chen et al., 2021)	N/A	N/A	X

NOTE: Not Available (N/A), Not Used (X).

management.

The increasing adoption of telehealth in patient assessment and care delivery in recent times provides an avenue for patients to share symptom data with clinicians from the comfort of their homes. Such a telehealth IoT infrastructure has seen increasing usage for remote patient monitoring while maintaining social distancing imposed by the COVID-19 pandemic (Jahmunah et al., 2021), (Kelly et al., 2020). These IoT infrastructures are commonly composed of sensor-to-cloud architectures often limited to edge devices; Such infrastructures are not scalable for sophisticated healthcare setups due to increased data transmission costs and long response time (Hartmann et al., 2019). In such cases, integrating IoT infrastructure with a Fog computing architecture can bridge sensor and cloud nodes and improve the IoT infrastructure's efficiency and performance while reducing network traffic (Farahani et al., 2020). Edge-fog architectures can also increase participant privacy compared to cloud computing approaches (Pallas et al., 2020). Researchers have explored the role of Fog devices in patient health monitoring (Verma & Sood, 2018). They motivate the use of Fog computing in monitoring symptoms and progression of PD in remote or home settings.

In our literature survey, we also found various machine learning techniques to estimate the UPDRS scoring (Patel et al., 2009; Sano et al., 2016). However, we could not find any hand movement task classification system for the UPDRS scoring activities for PD, which is directly related to PD symptoms. Also, research involving the combinational architecture of edge-fog computing combined with ML approaches offering a clinical representation of PD movement symptoms is scarce. Exploring such edge-fog-supported machine learning-based telehealth infrastructure can provide meaningful insights related to remote UPDRS assessment and reduce the clinical data translation barriers. We propose an edge-fog-enabled IoT framework that uses an edge node to record the movement and a Fog node to serve as a reliable local intermediator for classifying UPDRS-based motor assessment tasks.

3. Materials and Methods

The IoT infrastructure consisted of a pair of wearable smart gloves that collected data from sensors (five flex sensors and one IMU sensor) and transmitted the sensor data to a local Fog node wirelessly, as shown in Fig. 2 (see Fig. 1). The following section discusses (A) the IoT infrastructure for data collection, (B) the data collection study, and (C) the simulation environment for edge-based activity detection and fog-based activity classification experiments.

3.1. IoT infrastructure for data collection

The IoT infrastructure comprised of (i) an Edge node of smart gloves and (ii) a Fog node of Raspberry Pi as discussed below.

3.1.1. Smart gloves as edge device

We designed a pair of e-textile Smart Gloves, which consisted of a microprocessor with a 10-bit in-built Analog-to-Digital Converter (ADC). Flex sensors were attached to the microcontroller through ADC for recording finger movements (Constant et al., 2021). For this, five flex sensors were placed on the five fingers of the gloves. We also integrated an Inertial Measurement Unit (IMU), having a triaxial accelerometer and gyroscope, in the gloves. We recorded the finger movement data through flex sensors and hand orientation data through the IMU sensor. The data were sampled from these sensors at an effective sampling frequency of 40Hz upsampled to 128Hz. The microprocessor was also equipped with wireless data transmission/reception capabilities. We transmitted the data recorded from the sensors to a remote Fog node discussed below. The responsibility of the edge node was to detect activities and compute the features used to classify those activities.

3.1.2. Fog node architecture

We designed a Fog node infrastructure using a single-board computer. The Fog node was able to connect with Smart Gloves wirelessly for collecting sensor data and offer a Graphical User Interface (GUI) to guide through UPDRS-based hand exercise tasks. The GUI presented a small demo video about the exercise followed by the timed exercise duration (i.e., 10 s), during which the data was collected from gloves. Once the user completes the given exercise task (using the gloves), the user can click the 'Next' button on the GUI

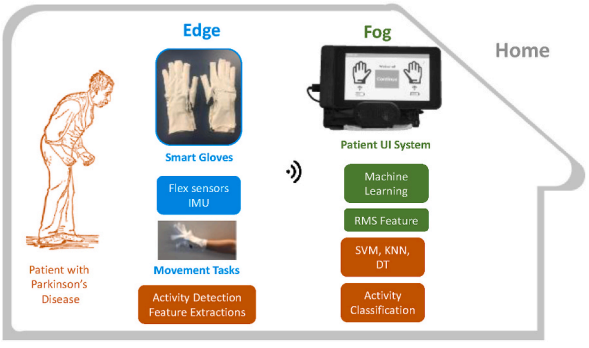


Fig. 1. An overview of the telehealth architecture supported by edge and Fog computing.

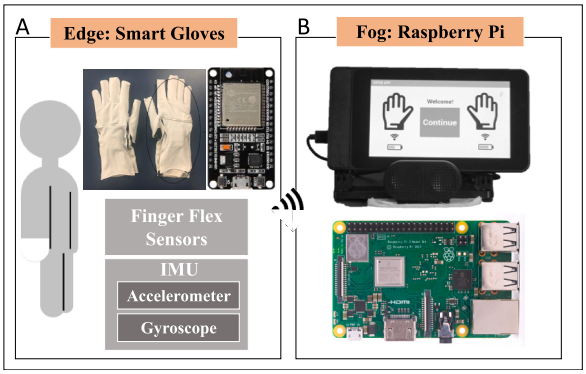


Fig. 2. The IoT infrastructure for data collection.

to go to the next exercise or click the 'Complete' button to finish the exercise. The data is logged into a comma-separated-values (CSV) file for each exercise separately. Each participant was given a unique ID number, and all participant data files were stored in their respective folders. Additionally, the responsibility of the Fog node was to classify movement exercise tasks performed by the patients using the ML algorithm.

3.2. Data collection study

The hand movement data were collected using smart gloves (discussed above). Exercise tasks according to a sub-section of the MDS-UPDRS protocol (i.e., MDS-UPDRS PART III) were followed for this study. We collected data from participants with PD (n = 4, two males, and two females) and healthy controls (n = 5, two males and three females), with the mean age for PD participants being 73.25 years and healthy participants being 70.8 years. Participants performed the data collection in the Wearable Biosensing Lab at the University of Rhode Island (URI). This study was approved by the Institutional Review Board (IRB) at the URI (IRB No: 1020863–8).

After the consent process, participants were asked to wear smart gloves and sit comfortably with their back straight in the chair. Fig. 3 shows a typical study setup with the participant wearing the smart glove. They were sequentially guided through the UPDRS exercise tasks using a Graphical User Interface (GUI) which demonstrated the exercise tasks using animated images. This GUI was also connected to the smart gloves and used to log the data while the user performed exercise tasks. We selected four hand exercise tasks related to motor examination from MDS-UPDRS Part III. Each exercise was performed for 10 s. Fig. 4 displays a picture of four upper limb UPDRS exercises. Participants completed the following hand motor exercises for both the left and right hands (see Fig. 5).

- **Finger to Nose:** With the arm starting from an outstretched position, with the index finger pointing forward, the patient was asked to place their index finger on their nose, then put their hand back to the initial position. The maneuver was performed repeatedly. This exercise was meant to observe the ***amplitude of kinetic tremors in their hands.***
- **Finger Tapping:** The participant was asked to repeatedly tap their index finger on the thumb as quickly and as widely as possible. This exercise was meant to observe the ***speed, amplitude, and decrementing amplitude of tapping, as well as hesitations and halts in tapping.***

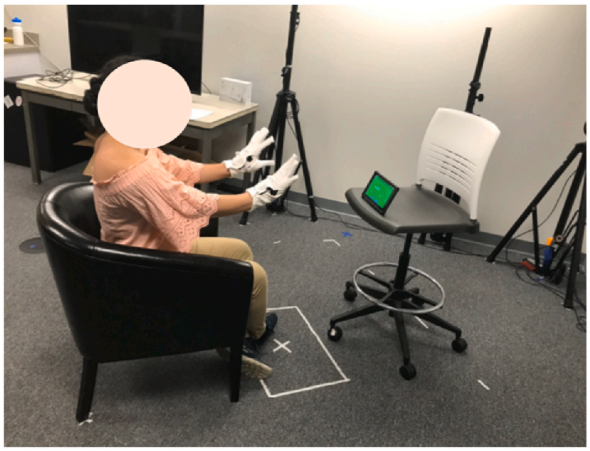


Fig. 3. Data collection setup.

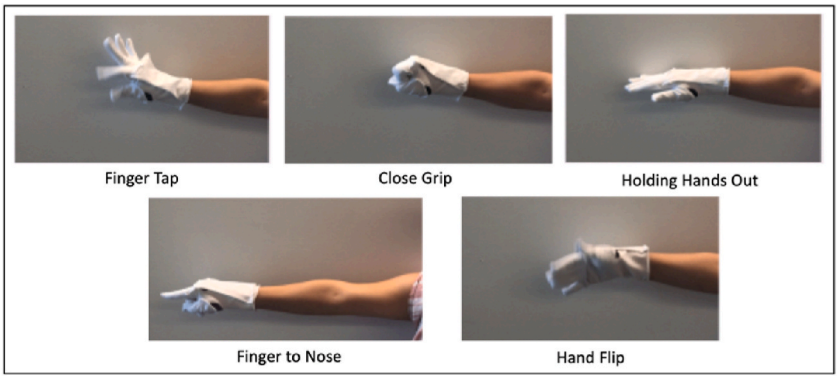


Fig. 4. Visual showing upper limb UPDRS exercises.

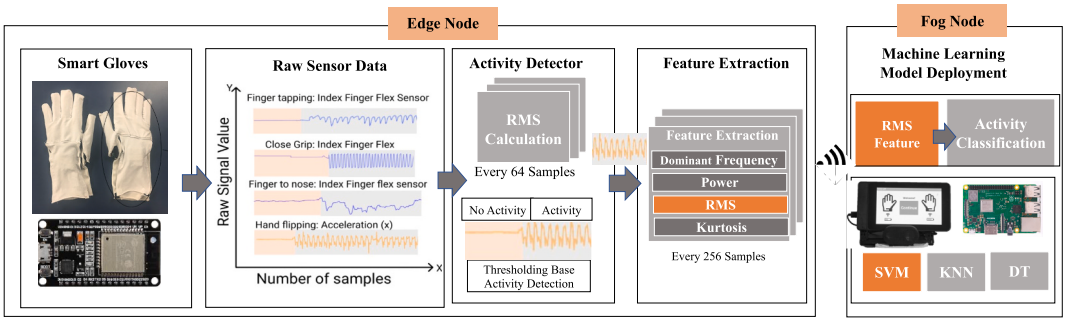


Fig. 5. Data processing pipeline Implemented on the Edge (ESP32) device and Fog device (Raspberry Pi).

- **Close Grip:** The participant was asked to make a fist with their palms facing downward. The participant then repeatedly alternates between having their hands closed and opened as fast and as widely as possible. This exercise was meant to observe, again, ***speed, amplitude, hesitation, halts, and decrementing amplitude of hand movements.***
- **Pronation-Supination Movements of Hands (Hand flip):** With arms stretched forward, and fists facing down, the participant was asked to turn their palm up and down alternately, as fast and as widely as possible. This exercise was meant to observe the ***speed, amplitude, hesitation, halts, and decrementing amplitude of hand flipping.***

3.3. IoT environment for real-time movement exercise detection and classification

This section discusses the tools that we developed to process the data collected using gloves and the Fog setup. The constraints on accessing the patients placed by the COVID-19 pandemic restricted us from evaluating the performance of the proposed detection and classification model. Therefore, we designed an IoT testing environment that used the data collected from patients during our pre-COVID experiment and processed it using an (IoT testing environment) in-home edge-fog infrastructure (Constant et al., 2021). This IoT testing environment involved an ESP32 device (edge device that acts as the simulated smart glove data source) and a Raspberry Pi (fog device that collects data from the edge device and runs classification models on collected data.

We used an ESP32 board set up as a simulated edge device which is a low-cost, low-power microprocessor with wireless communication capabilities. We interfaced an SD card containing all the raw data collected from our pre-COVID experiment with the ESP32 device. This data was fetched by the ESP32 from the onboard SD card at 128 Hz to match the sampling rate used for pre-COVID data collection. The data consisted of timestamp information, raw data from three flex sensors, a 3-axes accelerometer, and a 3-axes gyroscope. The data of 9 patients were placed on the SD card for this simulation experiment. The responsibility of the Edge (ESP32) device was (to simulate) the collection of data, pre-process the data to detect whether there is an activity present or not (using a thresholding-based activity detection method discussed in Section IIIC), and transmit raw sensor data to the Fog node if an activity is detected. Subsequently, the Fog node uses (UPDRS) exercise classification models to instantaneously classify the activity being performed by the patient, as discussed below.

3.3.1. Edge-based architecture for activity detection

The data from the gloves would be sent to the Fog node for activity classification. However, the transmission of the raw data to the Fog node would result in increased power consumption at the edge node. To account for the power constraint related to the edge node, we decided to send the data only when the activity is detected. To detect an activity, a thresholding-based activity detection algorithm

was designed by computing the Root Mean Square (RMS) value of the Euclidean Distance of the triaxial accelerometer data. The computed RMS value was compared against the pre-defined RMS threshold value to identify the presence of any activity. A computed RMS signal value greater than the threshold value will indicate the presence of activity and vice versa. The data was transmitted from the edge node to the Fog node only if the activity was detected using the activity detection method.

$$\mathrm{RMS} = \sqrt{1 \Big/ n\left(\sum_{i}^{n} x_i^2\right)} \tag{1}$$

where x is one of the input signals, n is the total number of samples in the window, and 'i' is an index going from 1 to n. The gloves' intended purpose is to capture the fine-grain hand movements of a person performing repetitive tasks. The tasks themselves were short in time, such as tapping a finger. We approximated that a person can tap their finger 2–3 times per second. We detect whether there is a signal activity for every half a second, i.e., 64-sample buffer, and consider an activity to be underway once it is continued for 2 s, i.e., 256 samples. We calculate the RMS feature for every 2-s window for all the channels, i.e., three flex sensors and 6 degrees of freedom IMU. If a consistent activity was detected for 2 s, the glove published data to the Fog node. We used the Message Queue Telemetry Transport (MQTT) communication protocol to transfer this data from the edge to the Fog node.

3.3.2. Fog-based architecture for activity classification

A Raspberry Pi Model 4B was used as the Fog node to deploy the activity classification model. Once the edge node detects the activity, the RMS feature dataset is sent to the Fog node. This feature dataset received by the Fog node consisted of average RMS features for the window of 256 samples associated with the Index, Ring, Thumb, Acc(x), Acc(y), Acc(z), Gry(x), Gry(y) and Gry(z) sensors. These features are passed to the classification model deployed on the Fog node. We deployed two classification models (KNN and SVM) on the Fog node for their respective performance comparisons. The classification model was developed using an M1 MAC BOOK PRO computer. The development procedure for these models is discussed in section 3 below. The Fog node then classified the UPDRS exercise based on an ML algorithm using the RMS feature of different sensors.

3.3.3. Development of the ML models for IoT experimental environment

The ML models were deployed at the Fog node using the features such as RMS sent by the edge node. For this, we developed an ML classification model based on a 32-bit processor architecture supported by the Raspberry Pi 4 B. We developed ML models using KNN, SVM, and DT using the M1 MAC BOOK PRO computer (i.e., the computer). We used our previously recorded data (Section 3 A) for model development. The data contained different hand movement tasks with labels. These labels included various tasks such as Finger to Nose (FN), Finger Tapping (FT), Close Grip (CG), and Hand Flipping (HF).

The data of 9 participants were compiled to create a master data set containing all the data with label information. This master data set was split into three sub-sets, i.e., training, testing, and validation. The training set included 80% of the data to develop the ML models. The testing and validation sets had 10% of data in each and were used to evaluate the accuracy of the models (Section IV). Care was taken to ensure a stratified split such that the data sub-sets contained information from all exercise tasks (FN, FT, CG, HF) based on their respective distribution. To understand how the variation in the training, testing, and validation data sets affects the performance of the models, we shuffled the master data set and created 11 pairs (T1-T11) of training, testing, and validation sub-sets having mixed data. The sum of all data in training, testing, and validation sets from any of the T1-T11 pairs will resemble the master dataset. The training, testing, and validation of the models were conducted on the computer. We used different hyperparameters related to each ML model to identify the best fit. The best hyperparameters were selected after performing a grid search. The hyperparameters for each ML model are described below:

- **KNN Model**: *Number of Neighbors* ranging from 3 to 23 (odd numbers only); *Distance Metric* involving Euclidean, Manhattan, Minkowski, Chebyshev; *Weight Metric* as uniform and distance weight matrix.
- **SVM Model:** Kernel Radial Basis Function (RBF) with Gamma values of 1, 0.1, 0.01, 0.001, 0.0001.
- **DT Model:** Maximum Depth ranging from 1 to 10; Minimum number of splits ranging from 1 to 10; Minimum samples leaf ranging from 1 to 5; The DT was developed based on two criteria, i.e., "gini" for the Gini impurity and "entropy" for the information gain.

Different ML models were trained and saved using Python programming language and the Sklearn library. In addition, we also experimented with varying features of raw data to assess the performance of each feature on the model accuracy. For this, we computed different features (namely Power, RMS, Kurtosis, Dominant Frequency, and Area under the curve) of the raw data. We assessed the performance of each classifier model with each feature separately and with a set of all the features together. Further, we also tested our models with 2-fold, 5-fold, and 10-fold cross-validation. After analyzing the outcomes of these experiments on the validation accuracy, the best model was selected for deployment on the Fog node, i.e., Raspberry Pi.

4. Results and discussion

In our proposed work, we wanted to identify a machine learning technique that can be deployed on the Fog node for instantaneous motor exercise task classification. For this, we conducted experiments with KNN, SVM, and DT classifiers with different data features and cross-validation techniques. We present our findings and discussion about these experiments in this section.

4.1. Analyzing and identifying data features for activity classification

We collected raw data (3x flex sensor data and 6x IMU sensor data) using the sensor-based gloves. We computed different data features from our raw data. These features included Root-Mean-Square (RMS), Kurtosis (KURT), Area Under the Curve (AUC), Power (P), and Dominant Frequency (DF). We computed these features for each of the nine data channels using Eqs. (2)–(5). In Equations (2), (3) and (5), N is the total number of samples, n is the index of the Nth sample, and k is a list of values from 0 to N-1.

$$\text{Discrete Fourier Transform } x[k] = \sum_{n=0}^{N-1} x[n] e^{(-j2\pi kn)/N} \tag{2}$$

$$\text{RMS} = \sqrt{1 \Big/ N \left(\sum_{n=0}^{N} x_n \hat{2} \right)} \tag{3}$$

$$\text{AUC} = \int_a^b f(x)dx \sim (b-a) \,.\, (f(a)-f(b))/2 \tag{4}$$

$$\text{Kurtosis} = \frac{\sum_{n=0}^{N} (x_n - x_n)^4}{\sum_{n=0}^{N} (x_n - x_n^2)^2} \tag{5}$$

We applied this feature data set to each ML classifier (KNN, SVM, DT) based on the stratified split (T1-T11, Section III·C.3) for evaluating the accuracy of classifiers. Tables 2–4 show the average classification accuracy and standard deviation for testing, training, and validation data set for KNN, SVM, and DT models, respectively, for different features.

As can be seen from Tables 2–4, KURT and DF features were not useful for achieving classification accuracy above 90% through any of the classification models and cross-validation techniques (Tables 2–4, Highlighted in Red). Thus, we decided not to select KURT and DF features for our classification model to be deployed on the Fog node.

The AUC feature showed promise to be a good classification feature when used with the KNN model. However, the outcomes using the SVM and DT classifiers showed unsatisfactory performance of the AUC feature for activity classification (Tables 3 and 4, Highlighted in Red).

The remaining feature, Power and RMS, showed competitive results to be good candidates for activity classification. Our detailed analysis showed that, though Power was offering higher training and testing accuracy for the KNN classifier, it was performing poorly for validation cases (Table 2, Highlighted in Red). Again, based on the outcomes of the DT classifier, we found that the Power feature was performing poorly, with a classification accuracy of less than 90% (Table 4, Highlighted in Red). Thus, we did not consider the Power feature for our Fog classifier model.

Looking at the RMS feature, we found that the RMS was able to offer classification accuracy higher than 90% using KNN and SVM classifiers (Tables 2 and 3). Though being one of the top performers using the DT classifier, accuracy for RMS was below 90%, suggesting that, overall, the DT classifier could not learn the classification in an acceptable manner (Table 4).

4.2. Identifying ML model and cross-validation technique for activity classification

- Identification of cross-validation technique

Now that we have selected RMS as the feature of choice, we focused our attention on comparing 2-5-10 fold cross-validation folds across KNN and SVM. In the KNN model, we identify 5-fold cross-validation as the ideal fold as its training and testing accuracy are the same (94%) which indicates that the model would generalize better than 2- or 10-fold cross-validation. Even though the training

Table 2
KNN Classification accuracies for different features.

KNN Accuracy (Training; Testing; Validation)					
AVG(T1-11) %	Power	RMS	Kurt	DF	AUC
2 F –Training	92(1)	92(1)	62(2)	68(2)	90(1)
2 F –Testing	94(1)	94(3)	66(6)	69(4)	93(5)
2 F- Validation	88(2)	93(1)	70(3)	76(24)	94(2)
5 F –Training	94(1)	94(1)	66(2)	69(2)	92(1)
5 F –Testing	94(2)	94(3)	67(6)	69(5)	94(4)
5 F- Validation	88(2)	93(1)	69(4)	82(20)	95(1)
10 F –Training	94(1)	95(1)	67(1)	69(2)	93(1)
10 F –Testing	94(2)	94(4)	67(5)	69(5)	94(4)
10 F-Validation	88(2)	93(1)	69(4)	75(26)	94(2)

NOTE: Red Highlight-Rejected Model & <90% accuracy; Green- Best performing model & >90% accuracy; No Highlight >90% validation accuracy but Rejected model.

Table 3
SVM Classification accuracies for different features.

SVM Accuracy (Training; Testing; Validation)					
AVG(T1-11) %	Power	RMS	Kurt	DF	AUC
2 F –Training	92(1)	91(1)	62(2)	70(1)	82(2)
2 F –Testing	91(4)	93(3)	67(7)	70(6)	86(4)
2 F- Validation	93(2)	93(1)	73(3)	69(7)	85(2)
5 F –Training	93(1)	94(1)	65(1)	71(1)	84(1)
5 F –Testing	91(4)	93(3)	66(6)	71(5)	86(4)
5 F- Validation	92(2)	93(1)	70(4)	70(7)	85(2)
10 F –Training	91(1)	94(1)	66(1)	72(1)	85(1)
10 F –Testing	91(4)	94(3)	66(8)	71(5)	86(5)
10 F-Validation	92(2)	93(1)	72(3)	67(9)	85(2)

NOTE: Red Highlight-Rejected Model & <90% accuracy; Green- Best performing model & >90% accuracy; No Highlight >90% validation accuracy but Rejected model.

Table 4
DT Classification accuracies for different features.

DT Accuracy (Training; Testing; Validation)					
AVG(T1-11) %	Power	RMS	Kurt	DF	AUC
2 F –Training	86(2)	85(2)	58(2)	70(1)	86(2)
2 F –Testing	88(7)	86(6)	60(6)	71(7)	87(6)
2 F- Validation	87(3)	86(5)	67(6)	70(5)	86(3)
5 F –Training	89(1)	88(1)	59(2)	72(1)	88(1)
5 F –Testing	88(6)	97(8)	60(8)	72(8)	86(8)
5 F- Validation	87(2)	86(5)	68(3)	70(4)	84(4)
10 F –Training	90(1)	89(1)	61(2)	72(1)	89(2)
10 F –Testing	86(7)	86(7)	62(8)	70(7)	86(6)
10 F-Validation	87(3)	85(5)	65(5)	70(5)	84(3)

NOTE: Red Highlight-Rejected Model & <90% accuracy; Green- Best performing model & >90% accuracy; No Highlight >90% validation accuracy but Rejected model.

accuracy of 10-fold cross-validation is higher than the training accuracy of 5-fold cross-validation, the testing accuracy in the 5-fold cross-validation experiment is less than the training accuracy (94%) itself. Also, the standard deviation in testing accuracy measurement (using our stratified split) was higher for 10-fold cross-validation compared to 5-fold. This might indicate that the KNN model is marginally overfitting in the case of 10-fold cross-validation. Therefore, we selected a 5-fold cross-validation for the KNN model.

For the SVM model, we compared the accuracy results across different cross-validation techniques, and we found that the 10-fold cross-validation had the highest training accuracy (94%) among all cross-validation techniques. Thus, we selected 10-fold cross-validation for the SVM model for activity classification on the Fog node.

- Identification of ML model

Having identified the feature and cross-validation techniques for each model, we further our investigations to compare the outcomes of both our selected models with their respective cross-validation techniques (i.e., KNN with 5-fold cross-validation and SVM with 10-fold cross-validation) through a confusion matrix. Using the confusion matrix, we were able to take a detailed look at how different activities (FT, FN, CG, HF) were classified by these two different models.

The Confusion Matrix presented in Tables 5–7 presents an average of the 11 randomly split validation experiments. We present the confusion matrix (Table 5) for the 5-fold cross-validation KNN model, and Table 6 shows the confusion matrix for the 10-fold cross-validation SVM model. Both tables were derived using the RMS feature. Our findings suggested that both models could classify different activities with more than 90% accuracy.

We observed a 9% error rate between Finger Tapping and Close Grip exercises, which accounts for the most errors in the KNN

Table 5
KNN confusion matrix.

Predicted Class	True Class					
	FT	FN	CG	HF	SN	SP
FT	100%	0%	0%	0%	100%	97%
FN	0%	91%	0%	9%	91%	96%
CG	0%	9%	91%	0%	90%	96%
HF	0%	0%	0%	91%	90%	100%

NOTE: Sensitivity (SN), Specificity (SP).

Table 6
SVM confusion matrix.

Predicted Class	True Class					
	FT	FN	CG	HF	SN	SP
FT	100%	0%	0%	0%	100%	100%
FN	0%	100%	9%	0%	100%	96%
CG	0%	0%	91%	0%	100%	100%
HF	0%	0%	0%	100%	100%	100%

NOTE: Sensitivity (SN), Specificity (SP).

Table 7
DT confusion matrix.

Predicted Class	True Class					
	FT	FN	CG	HF	SN	SP
FT	83%	8%	0%	9%	83%	94%
FN	17%	84%	10%	9%	83%	89%
CG	0%	8%	90%	0%	90%	96%
HF	0%	0%	0%	82%	100%	100%

NOTE: Sensitivity (SN), Specificity (SP).

model (Table 5). This can be attributed to the fact that during both the exercises (FT and CG), the two fingers (i.e., Index and Thumb) are almost moving in the same way, resulting in similar feature outcomes. In Table 6, we present the confusion matrix of the SVM classifier for the RMS 10-fold feature experiment. In this confusion matrix, we can observe that the classifier is classifying Finger to Nose as Close Grip 9% error. Again, this can be attributed to the fact that 4 (out of 5) fingers remain folded during the exercise for a certain period leading to this error in classification. These results suggested that the SVM classifier was performing better than the KNN classifier. An SVM classifier would be a more appropriate choice for our Fog node-based activity classification.

4.3. Evaluating the timing performance of the selected ML model on the fog device

To further investigate how both classifiers perform in terms of time for activity classification, we did a timing analysis experiment on our Fog node, where the models were deployed. In this experiment, we measured the classification time taken by each classifier model on the Fog device (i.e., raspberry pi). In our analysis, we found that the SVM classifier took 560 ($\pm$103) us, and the KNN classifier took 2388 ($\pm$435) us. Fig. 6 shows the histogram of the time taken to classify activities for both SVM and KNN classifiers. We found that the SVM outperformed the KNN model and required less time than the KNN model to classify an activity on the Fog node.

4.4. Timing estimation for the proposed architecture

In this experiment, we measured time across various checkpoints in our processing pipeline to estimate latency. We broke down the pipeline into two tasks, (i) processing and (ii) transmission for each component (i.e., edge and Fog). The wearable device's processing was responsible for detecting activity and extracting features. The fog's processing was responsible for task classification using the trained model. We recorded the transmission time for sending data to fog device. This timing accounts for receiving an acknowledgment from the MQTT broker. Our experiment involved running all our data through the pipeline twice for a total of 162 activities. We observed that the mean transmission time was 2790 us with a standard deviation of 334 us. Fig. 7 shows a histogram of transmission time for 162 activities being detected.

5. Conclusion

In this work, we demonstrated an edge-fog-supported machine learning-based IoT infrastructure for movement activity (related to UPDRS) classification of Parkinson's patients. We developed ML models that can be deployed on a Fog architecture (Raspberry Pi) to classify hand movements related to the UPDRS motor exam. We presented an edge architecture that can calculate data feature (i.e., RMS) on a low-power device such as ESP32 and communicate it wirelessly to a Fog device such as Raspberry Pi.

Our in-lab experiments showed that the RMS feature using SVM with 10-fold CV offered the best classification accuracy for hand activities. Overall, our findings suggested that the KNN and SVM models offered similar classification accuracy (93 ($\pm$1) %). Though the KNN model performed very similarly to SVM in terms of accuracy, the classification timing analysis demonstrated that the SVM is a better choice for classifying movement activities. In our timing experiment, we observed that the mean classification time for SVM was 560($\pm$103) us, which was approximately four times lower than the KVM. Based on these findings, we selected the SVM model for activity classification on the Fog node.

A limitation of our presented work was that the experiments were conducted in lab settings. In the future, we would like to deploy this system in real-world settings such as PD patients' homes. Further investigations are needed to examine the efficacy of the proposed

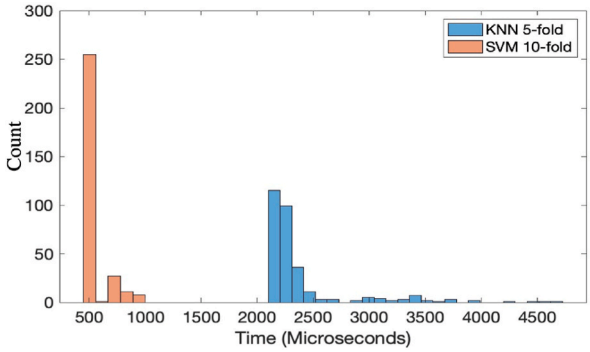


Fig. 6. Activity classification inference timing recorded on Fog node (Raspberry Pi).

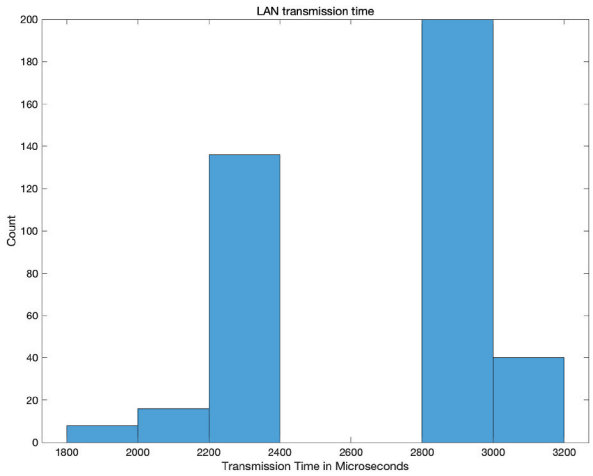


Fig. 7. Local area network transmission time from edge to Fog node.

system to classify more detailed UPDRS scores based on the movement tasks.

Our preliminary findings are promising, and we believe that the proposed system can be useful for providing real-time feedback to patients about the motor examination tasks, and it could be useful for longitudinal disease progression monitoring and PD symptom management. Such a system can allow a clinician to track symptom progression, optimize the treatment plan for a patient and open a new horizon for remote patient management.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:Kunal Mankodiya reports financial support was provided by National Science Foundation.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank the participants who participated in our study. This work was supported by the National Science Foundation under grant numbers #1652538 and #1919135.

References

Chen, Y., Ma, H., Wang, J., Wu, J., Wu, X., & Xie, X. (2021). PD-Net: Quantitative motor function evaluation for Parkinson's disease via automated hand gesture analysis. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 2683–2691). https://doi.org/10.1145/3447548.3467130

Constant, N., et al. (2021). Data analytics for wearable IoT-based telemedicine. In *Wearable sensors* (pp. 357–378). Academic Press.

Farahani, B., Barzegari, M., Shams Aliee, F., & Shaik, K. A. (2020). Towards collaborative intelligent IoT eHealth: From device to Fog, and cloud. *Microprocessors and Microsystems, 72*(Feb). https://doi.org/10.1016/j.micpro.2019.102938

Ferreira, D., Azevedo, E., & Araújo, R. (2021). Teleneurology in Parkinson's disease: A step-by-step video guide. *Acta Neurologica Scandinavica, 144*(2), 221–225. https://doi.org/10.1111/ane.13429. Blackwell Publishing Ltd.

Galna, B., Barry, G., Jackson, D., Mhiripiri, D., Olivier, P., & Rochester, L. (2014). Accuracy of the Microsoft Kinect sensor for measuring movement in people with Parkinson's disease. *Gait & Posture, 39*(4), 1062–1068. https://doi.org/10.1016/j.gaitpost.2014.01.008

Goetz, C. G., et al. (2008). Movement disorder society-sponsored revision of the unified Parkinson's disease rating Scale (MDS-UPDRS): Scale presentation and clinimetric testing results. *Movement Disorders, 23*(15), 2129–2170. https://doi.org/10.1002/mds.22340

Hartmann, M., Hashmi, U. S., & Imran, A. (2019). Edge computing in smart health care systems: Review, challenges, and research directions. *Transactions on Emerging Telecommunications Technologies*. https://doi.org/10.1002/ett.3710

Jahmunah, V., et al. (2021). Future IoT tools for COVID-19 contact tracing and prediction: A review of the state-of-the-science. *International Journal of Imaging Systems and Technology, 31*(2), 455–471. https://doi.org/10.1002/ima.22552

Kelly, J. T., Campbell, K. L., Gong, E., & Scuffham, P. (2020). The Internet of things: Impact and implications for health care delivery. *Journal of Medical Internet Research, 22*(11). https://doi.org/10.2196/20135. JMIR Publications Inc.

Kim, J. W., et al. (2011). Quantification of bradykinesia during clinical finger taps using a gyrosensor in patients with Parkinson's disease. *Medical, & Biological Engineering & Computing, 49*(3), 365–371. https://doi.org/10.1007/s11517-010-0697-8

Larson, D. N., Schneider, R. B., & Simuni, T. (2021). A new era: The growth of video-based visits for remote management of persons with Parkinson's disease. *Journal of Parkinson's Disease, 11*(s1), S27–S34. https://doi.org/10.3233/JPD-202381. IOS Press BV.

Mera, T. O., Heldman, D. A., Espay, A. J., Payne, M., & Giuffrida, J. P. (2012). Feasibility of home-based automated Parkinson's disease motor assessment. *Journal of Neuroscience Methods, 203*(1), 152–156. https://doi.org/10.1016/j.jneumeth.2011.09.019

Naqvi, E. (2018). *Parkinson's disease statistics*. Parkinson's News Today. https://parkinsonsnewstoday.com/parkinsons-disease-statistics/.

van den Noort, J. C., et al. (2017). Quantification of hand motor symptoms in Parkinson's disease: A proof-of-principle study using inertial and force sensors. *Annals of Biomedical Engineering, 45*(10), 2423–2436. https://doi.org/10.1007/s10439-017-1881-x

Pallas, F., Raschke, P., & Bermbach, D. (2020). Fog computing as privacy enabler. *IEEE Internet Computing, 24*(4), 15–21. https://doi.org/10.1109/MIC.2020.2979161

Patel, S., et al. (2009). Monitoring motor fluctuations in patients with Parkinsons disease using wearable sensors. *IEEE Transactions on Information Technology in Biomedicine, 13*(6), 864–873. https://doi.org/10.1109/TITB.2009.2033471

Rigas, G., et al. (2012). Assessment of tremor activity in the Parkinsons disease using a set of wearable sensors. *IEEE Transactions on Information Technology in Biomedicine, 16*(3), 478–487. https://doi.org/10.1109/TITB.2011.2182616

Salarian, A., Russmann, H., Wider, C., Burkhard, P. R., Vingerhoets, F. J. G., & Aminian, K. (2007). Quantification of tremor and bradykinesia in Parkinson's disease using a novel ambulatory monitoring system. *IEEE Transactions on Biomedical Engineering, 54*(2), 313–322. https://doi.org/10.1109/TBME.2006.886670

Sano, Y., et al. (2016). Quantifying Parkinson's disease finger-tapping severity by extracting and synthesizing finger motion properties. *Medical, & Biological Engineering & Computing, 54*(6), 953–965. https://doi.org/10.1007/s11517-016-1467-z. Jun.

Tysnes, O. B., & Storstein, A. (2017). Epidemiology of Parkinson's disease. *Journal of Neural Transmission, 124*(8), 901–905. https://doi.org/10.1007/s00702-017-1686-y. Springer-Verlag Wien.

Verma, P., & Sood, S. K. (2018). Fog assisted-IoT enabled patient health monitoring in smart homes. *IEEE Internet of Things Journal, 5*(3), 1789–1796. https://doi.org/10.1109/JIOT.2018.2803201