Reinforcement Learning-Based Adaptive Classification for Medication State Monitoring in Parkinson's Disease

Mustafa Shuqair, *Graduate Student Member, IEEE*, Joohi Jimenez-Shahed, and Behnaz Ghoraani, *Senior Member, IEEE*

Abstract*—Parkinson's Disease (PD) patients frequently transition between the 'ON' state, where medication is effective, and the 'OFF' state, affecting their quality of life. Monitoring these transitions is vital for personalized therapy. We introduced a framework based on Reinforcement Learning (RL) to detect transitions between medication states by learning from continuous movement data. Unlike traditional approaches that typically identify each state based on static data patterns, our approach focuses on understanding the dynamic patterns of change throughout the transitions, providing a more generalizable medication state monitoring method. We integrated a deep Long Short-Term Memory (LSTM) neural network and three one-class unsupervised classifiers to implement an RL-based adaptive classifier. We tested on two PD datasets: Dataset PD1 with 12 subjects (14-minute average recording) and Dataset PD2 with seven subjects (120-minute average recording). Data from wrist and ankle wearables captured transitions during 2 to 4-hour daily activities. The algorithm demonstrated its effectiveness in detecting medication states, achieving an average weighted F1-score of 82.94% when trained and tested on Dataset PD1. It performed well when trained on Dataset PD1 and tested on Dataset PD2, with a weighted F1-score of 76.67%. It surpassed other models, was resilient to severe PD symptoms, and performed well with imbalanced data. Notably, prior work has not addressed the generalizability from one dataset to another, essential for real-world applications with varied sensors. Our innovative framework revolutionizes PD monitoring, setting the stage for advanced therapeutic methods and greatly enhancing the life quality of PD patients.

***Index Terms*—Parkinson's disease, reinforcement learning, machine learning, deep q-learning, wearable health monitoring, wearable sensors.**

Received 6 November 2023; revised 16 April 2024 and 10 June 2024; accepted 29 June 2024. Date of publication 5 July 2024; date of current version 4 October 2024. The work of B. Ghoraani, PI was supported by NSF under Grant 1936586 and Grant 1942669. This work was supported by NIH under Grant 1R43NS071882-01A1 (to Cleveland Medical Devices) and Grant 5R44AG044293 (to Great Lakes NeuroTechnologies Inc). *(Corresponding author: Behnaz Ghoraani.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the institutional review boards of Great Lakes NeuroTechnologies, the University of Rochester, and Johns Hopkins University, and performed in line with the Helsinki.

Mustafa Shuqair and Behnaz Ghoraani are with the Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431 USA (e-mail: bghoraani@fau.edu).

Joohi Jimenez-Shahed is with the Department of Neurology, Icahn School of Medicine at Mount Sinai, New York, NY 10029 USA.

This article has supplementary downloadable material available at https://doi.org/10.1109/JBHI.2024.3423708, provided by the authors.

Digital Object Identifier 10.1109/JBHI.2024.3423708

I. Introduction

PARKINSON'S disease (PD) is a debilitating neurodegenerative condition characterized by pronounced motor symptoms such as tremors and gait difficulties [1]. The frequently prescribed medication, Levodopa, alleviates these symptoms but leads to motor fluctuations, causing shifts between the 'medication ON' state, where the drug's effects are optimal, and the 'medication OFF' state, where they are minimal [2]. Addressing these fluctuations remains a pivotal challenge in PD treatment [3]. Current management relies on therapy adjustments based on patient self-reports, which, due to recall biases, can be unreliable [4]. The emergence of wearable sensors combined with machine learning advancements offers a promising avenue for detecting these medication state transitions [5], [6], [7], [8], [9].

Prior research in this domain has laid important groundwork. Pérez-López et al. [10] utilized waist accelerometer data to identify dyskinesia and bradykinesia during walking as a proxy for detecting medication states. Similarly, Ossig et al. [11] analyzed spectral power from wrist accelerometer data through the Parkinson's KinetiGraph (PKG) for state monitoring. Rodríguez-Molinero et al. [12] used waist sensor data to identify bradykinesia and detect medication states during walking. Fisher et al. [13] employed an artificial neural network (ANN) with a waist sensor to detect medication OFF states. Notably, Hssayeni et al. [14] developed an individualized SVM approach using data from wrist and ankle sensors. Deep learning models, such as the Convolutional Neural Networks (CNNs) used by Um et al. [15] and Pfister et al. [16], have been employed for monitoring motor state fluctuations with wrist-acquired data.

However, these existing methods have notable limitations. Many excel in controlled settings or specific activities but do not address the real-world variability of PD symptoms across different datasets, an aspect critical for robust daily monitoring [17]. The majority are also tailored for a single dataset, overlooking the potential benefits of cross-dataset validation, which is essential for generalizability in unconstrained daily activities (cross-domain testing). In this study, we introduce a

novel methodology designed for wearable-based motor fluctuation monitoring in PD patients within the uncontrolled settings of daily life. Unlike traditional classification methods, which may struggle with inter-subject variability due to the assumption of similar data distribution across training and testing datasets [18], our model focuses on capturing the patterns of medication state transitions. Instead of modeling individual medication states like 'ON' or 'OFF,' our approach emphasizes capturing transition patterns between these states, addressing the challenges of distinct data distributions.

We present a unique approach using Reinforcement Learning (RL) to discern shifts in data stream distributions during state transitions. By training an RL agent to recognize these transitions, it becomes adept at identifying transitions in new data and dynamically represents a patient's medication state. Instead of static models, our agent interacts with the data's dynamics and makes optimal health state decisions. This RL-driven framework prioritizes understanding changes in data distribution, making it robust against inter-subject and intra-subject variations. Building on a preliminary version in [19], this paper delves deeper with theoretical formulations, sensitivity analyses, and real-world health monitoring validations.

Our study utilizes accelerometer and gyroscope data streams acquired from two wearable sensors affixed to the subjects' most affected wrists and ankles to monitor individuals across various Activities of Daily Living (ADLs). The study employs two PD datasets: Dataset PD1 [20] and Dataset PD2 [21]. Our methodology began with evaluating our model on Dataset PD1 using a leave-one-subject-out approach, termed the *Within-Domain* scenario. Subsequently, we trained the model using Dataset PD1 and tested its efficacy on Dataset PD2, designating this as the *Cross-Domain* scenario. Our model's performance was juxtaposed against leading methods using SVM classifiers and deep CNNs. The results showcased superior efficacy, offering healthcare professionals a refined tool for managing PD motor fluctuations, thus improving patient life quality. This work highlights the promise of RL in healthcare, emphasizing its ability to craft tailored and adaptive solutions to intricate health dilemmas.

II. Background

A. Reinforcement Learning

The RL problem is formalized based on the theory of dynamical systems to achieve optimal control on partially known Markov Decision Processes (MDPs). Optimal control is characterized by designing a controller that maximizes or minimizes a measure of a dynamical system's behavior over time. In RL, a learning agent interacts with an environment over time to achieve a set goal related to the environment's state. The agent observes the state of the environment and takes actions affecting this state. The three elements of observations, actions, and goals are enclosed within the MDPs framework. RL utilizes the MDPs framework to describe how the learning agent interacts with the environment regarding these three elements. [22]. The agent interacts with the environment at discrete time steps $t = 0, 1, 2, 3, \ldots$. At every step t, the learning agent acquires a representation of the environment's state $s_t \in \mathcal{S}$ and takes action $a_t \in \mathcal{A}$. The agent then receives a reward $r_{t+1} \in \mathcal{R} \subset \mathbb{R}$ and observes the new state s_{t+1} as an outcome of its action a_t. These states, actions, and rewards can be described as an MDP sequence:

$$s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, r_3, \ldots \tag{1}$$

For a finite MDP, all three elements $\mathcal{S}, \mathcal{A}$, and $\mathcal{R}$ have a finite number of elements. Therefore, the variables s_t and r_t have defined discrete probability distributions which only depend on s_{t-1} and a_{t-1}. The MDP's dynamics are defined by p for all $s', s \in \mathcal{S}, r \in \mathcal{R}$, and $a \in \mathcal{A}(s)$ as:

$$p(s', r|s, a) \doteq \Pr\{s_t = s', r_t = r|s_{t-1} = s, a_{t-1} = a\} \tag{2}$$

The reward received after each time step t is $r_{t+1}, r_{t+2}, r_{t+3}, \ldots$. The learning agent aims to maximize the total amount of received rewards or the expected return by taking a series of optimal actions. The expected return G_t, in (3), is defined as the sum of all the reward values, where $\gamma \in [0, 1]$ is a discount factor to assign a weight to future vs. immediate rewards.

$$G_t \doteq r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{3}$$

B. Q-Learning

The agent's performance when taking action a in a state s is assessed using value functions expressed by a series of actions called policies. When the agent follows a policy π at a time t, it maps the probabilities of selecting the possible action a from the current state s with policy $\pi(a|s)$ being the probability of $a_t = a$ when $s_t = s$. The expected return when taking action a in a given state s and following a policy π is defined as the action-value function for policy π:

$$q_\pi(s, a) \doteq \mathbb{E}_\pi[G_t|s_t = s, a_t = a] \tag{4}$$

The RL problem is solved by maximizing the expected return over a policy π to find the optimal action-value function q_*:

$$q_*(s, a) \doteq \max_\pi q_\pi(s, a) \tag{5}$$

Q-learning [23] is used to directly approximate q_*. The state-action pairs are updated using the Q-function to reach convergence as in (6). The $Q(s, a)$ at the convergence point is an estimation of q_*.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right], \tag{6}$$

where $\gamma \in [0, 1]$ and $\alpha \in (0, 1]$ are the discount and learning factors, respectively.

C. Deep Q-Network

Deep Q-network (DQN) [24] was introduced to approximate the optimal action-value function $q_*(s, a)$ within the Q-learning framework using deep neural networks. The Q-learning equation in (6) is reformulated as $Q(s, a, \theta)$ with θ being the neural

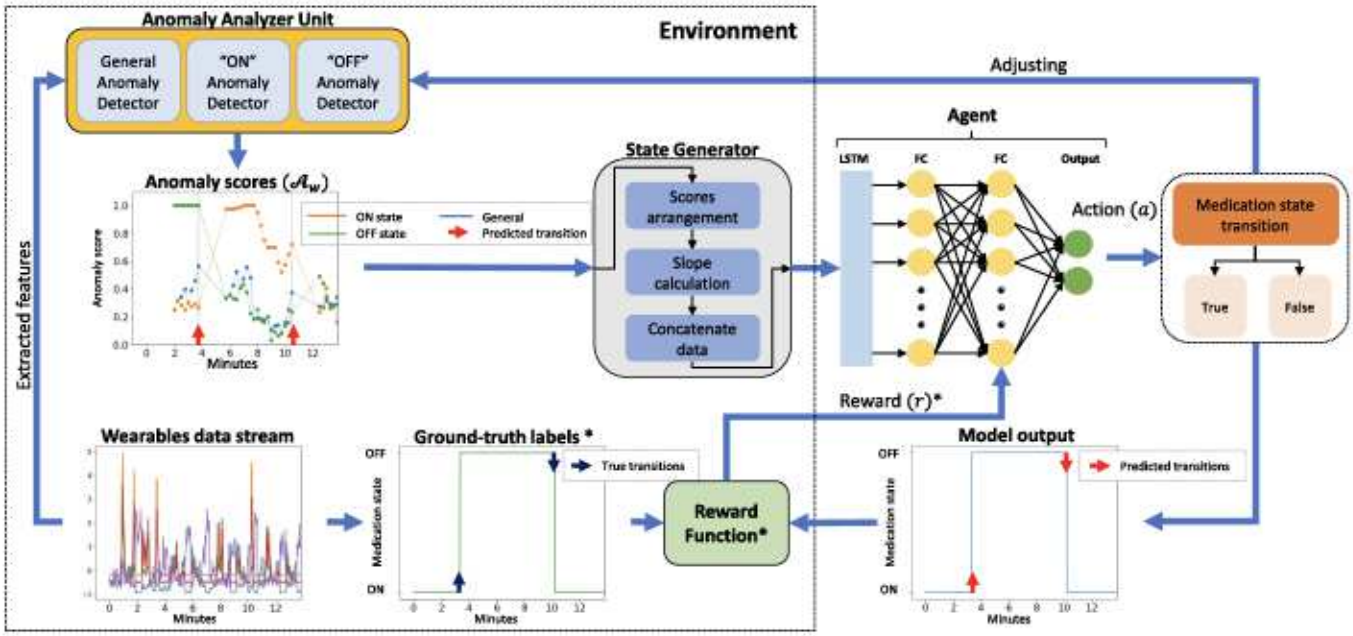

Fig. 1. The proposed reinforcement learning-based adaptive classification framework. The agent (i.e., the deep Q-network) learns the wearable sensors' data stream dynamics to optimize classification decisions for maximum reward r. The trained agent then observes the dynamics and classifies the incoming testing data stream. * The marked components are utilized only for the model's training and are eliminated during testing.

network's weights:

$$Q(s,a,\theta) \leftarrow r_{t+1} + \gamma \max_a Q(s_{t+1},a) \tag{7}$$

The neural network's output is the action a for a given state s. Therefore, training the neural network will result in updating the state-action pair and, at the convergence point, providing the optimal action-value function:

$$Q_*(s,a,\theta) \doteq \mathbb{E}_\pi \left[r_{t+1} + \gamma \max_a Q_*(s_{t+1},a) \right] \tag{8}$$

The relation in (9) describes the DQN loss function. Optimizing this loss function will minimize the difference between the $Q_*(s,a,\theta)$ (i.e., the first two terms in (9) and the current action-value function.

$$L_i(\theta_i) = \mathbb{E}\left[(r_{t+1} + \gamma \max_a Q(s_{t+1},a;\theta_{i-1}) - Q(s,a;\theta_i))^2 \right] \tag{9}$$

At every iteration i, the stochastic gradient descent method minimizes the loss function $L_i(\theta_i)$. Subsequently, (10) with $\eta \in (0,1]$ learning rate updates the DQN weights.

$$\theta_{i+1} \leftarrow \theta_i - \eta \nabla_\theta L_i(\theta_i) \tag{10}$$

D. Unsupervised Anomaly Detection

Anomaly detection focuses on pinpointing unexpected deviations from a system's typical behavior. Algorithms for anomaly detection define the standard distribution of the system's data and seek to recognize deviations when the incoming data diverges from this norm. When these data points lack labels, the challenge is unsupervised [25]. One-class classification techniques have been employed to understand the current data distribution, marking it as the baseline, and any divergence from this norm is classified as an anomaly [26]. This is achieved by establishing a decision boundary around the baseline data to differentiate it from any new incoming data, which might be anomalous [27]. SVMs have been suggested to form this decision boundary, using a hyperplane crafted to encapsulate the data domain. This is done by gauging the support vectors of the data's high-dimensional distribution without any specific class details [28]. Considering training as $x_1, x_2, .., x_l \in \mathcal{X}$, where l is the number of samples and $\mathcal{X} \in \mathbb{R}$, SVM defines the hyperplane parameters $\alpha_i > 0; i \in l$. These parameters determine the decision function for any new data point x:

$$f(x) = \text{sgn}\left(\sum_{i=1}^{l} \alpha_i k(x_i,x) - \rho \right), \tag{11}$$

where α_i are the support vectors describing the hyperplane, the parameter $\rho \in \mathbb{R}$ is optimized during training. The kernel function is computed from the dot product of the features map ϕ as $k(x_i,x_j) = (\phi(x_i)^T \cdot \phi(x_j))$. If a data sample x lies inside the hyperplane, the decision function $f(x)$ produces $+1$, indicating the baseline category. Conversely, if it does not, the result will be -1, marking it as an anomaly.

III. Methodology

A. Problem Definition

We reconceptualized the classification challenge by leveraging the MDP framework. Rather than crafting a conventional classifier that recognizes data patterns linked to particular medication states, our approach trains an RL agent to comprehend the continuously evolving distributions of data streams during transitions between these states. This method

positions the classification challenge as the environment, with the agent operating as the classifier. The agent's mission is to execute the most appropriate actions, thus reducing the frequency of misclassification. Consider the input sequence as $\mathcal{X} = x_1, x_2, \ldots, x_n, x_{n+1}, \ldots, x_{2n}, x_{2n+1}, \ldots$, where n stands for the number of data samples in a single time window segment and $\mathcal{X} \in \mathbb{R}$. Time steps are seen as consistent intervals. All the data points $X_w = x_1, x_2, \ldots, x_n$ fall within one segmented time window. Here, $w = 1, 2, 3, \ldots \in \mathbb{R}^+$ signifies the sequential order of these time windows in the input series.

In our MDP framework, each value of $u \in \mathbb{R}^+$ stands for a sequence of successive time windows, which equates to an individual discrete time point t within an event series consisting of a_t, r_{t+1}, and s_{t+1}. The action a_t is a two-fold decision, labeled as either $true$ or $false$. This decision hinges on whether the agent decides in favor of a transition between medication states. Following the agent's decision of a_t, the environment counteracts by delivering a reward denoted as r_{t+1} and presents a subsequent state, s_{t+1}. This new state influences the agent's next action, a_{t+1}. The reward r_{t+1} provided by the environment is derived from how accurately the agent predicts transitions between medication states. The upcoming state, s_{t+1}, is crafted to reflect the differences between the newly received data in time windows X_w and the data patterns inherent to the prevailing medication state.

In our designed environment, we introduce a component termed the Anomaly Analyzer Unit (AAU), detailed further in Section III-B-1. The AAU employs an ensemble of one-class classifiers, which are responsible for capturing the intricate data distribution of the prevailing medication state, thus establishing it as the reference class. Leveraging these classifiers, the AAU calculates anomaly values, denoted as $\mathcal{A}_w \in [0, 1]$, for every incoming time window labeled as w in relation to the reference class. An anomaly 0 indicates that the incoming window lacks any deviant data. Conversely, 1 suggests all data points within the window are outliers. Post the agent's decision of a_t, the AAU periodically refreshes the one-class classifiers after every u interval. Such periodic updates ensure that the anomaly values remain aligned with any evolving changes in the data distribution. The state s_t is expressed as:

$$s_t = \mathcal{A}_{tu+1}, \mathcal{A}_{tu+2}, \ldots, \mathcal{A}_{(t+1)u} \tag{12}$$

The design blueprint of our model is graphically represented in Fig. 1. We tackle the challenge by interpreting the sequence of state, action, and reward within the framework of an MDP, as illustrated in (2). Our primary objective is to amplify the expected return, as delineated in (3). The augmentation of this expected return is intrinsically linked to the precision of the agent's decision-making process. Through the fine-tuning of this MDP, the agent gradually learns to initiate actions, indicating a shift between medication states solely when authentic transitions are detected. Yet, for the agent to proficiently discern these authentic state shifts, it is essential that it learns to recognize fluctuations in the overall data stream distribution rather than merely concentrating on the individual distribution of each medication state. Consequently, we hypothesize that our adaptive methodology is better suited to unseen data with different baseline distributions. The agent's inherent capacity to assimilate from historical data and consequently make judicious decisions in a given environment [22] further supports our expectation. Thus, we predict a heightened performance when exposed to novel datasets.

B. Environment Design

The environment (Fig. 1) consists of an Anomaly Analyzer Unit (AAU), a State Generator, and a Reward Function.

1) Anomaly Analyzer Unit (AAU): The AAU is constructed by incorporating three one-class classifiers, each serving as an anomaly detector with a specific focus. The "General anomaly detector" captures the overall dynamics of the data as they evolve. The "Medication state ON anomaly detector" specializes in monitoring variations in data dynamics within the context of ON state data. The "Medication state OFF anomaly detector" specializes in tracking these variations within state OFF data. For every data sample x within a time window X_w, all three anomaly detectors yield an anomaly decision denoted as $f(x)$, determined according to (13). These individual anomaly decisions are then aggregated to calculate an anomaly score Aw for Xw as the ratio of the number of anomaly samples for which $f(x) = -1$ to the window length.

$$f(x) = \begin{cases} +1, & \text{if x is not anomaly,} \\ -1, & \text{if x is anomaly.} \end{cases} \tag{13}$$

Upon the agent's prediction of a transition event ($a_t = true$), the AAU trains the anomaly detectors for the General and the currently predicted medication state (e.g., OFF). This training uses the data from the following consecutive u time windows. Following that, the anomaly detectors for the current state and the anomaly detector for the opposite state (e.g., ON), which remains unaltered, are employed to evaluate incoming time windows. This testing generates an anomaly score A_w for each time window X_w. These anomaly scores, reflecting the data assessment within each window, contribute to the state s_t, as in (12). The current medication state anomaly detector is updated with each new time window w, and the General anomaly detector is updated at u time window intervals. After u time windows, the agent employs the state s_t to make its next prediction, resulting in a reward r_{t+1}. This process repeats until the agent identifies the next transition event or until the data stream concludes. Fig. 2 illustrates this workflow considering the discrete time steps $t = 0, 1, 2, 3$. In this example, there are two predicted transition events. The data stream initially starts in the medication state ON, prompting the training of both the General and ON anomaly detectors. As no OFF anomaly detector is trained yet and the incoming data pertains to medication state ON, an OFF anomaly score of 1.0 is assigned to these time windows. This visualization depicts the dynamic interplay between the agent, the AAU, and the evolving time-series data. The solid-filled, circle-shaped markers in Fig. 1 show samples of the anomaly scores.

2) State Generator: The anomaly scores Aw generated by the AAU are forwarded to the State Generator, as in Fig. 1. For

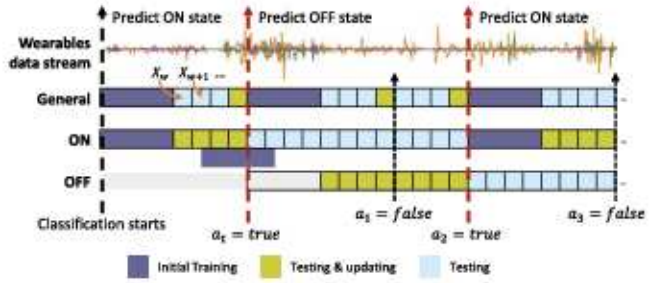

Fig. 2. Anomaly Analyzer Unit workflow as the incoming data stream is processed, and two transition events are detected. The segments indicate the duration the anomaly detectors (General, ON, OFF) were trained, applied to test the incoming data, or first applied and updated. a_t; agent actions at reinforcement learning discrete time step $t = 0, 1, 2, 3$.

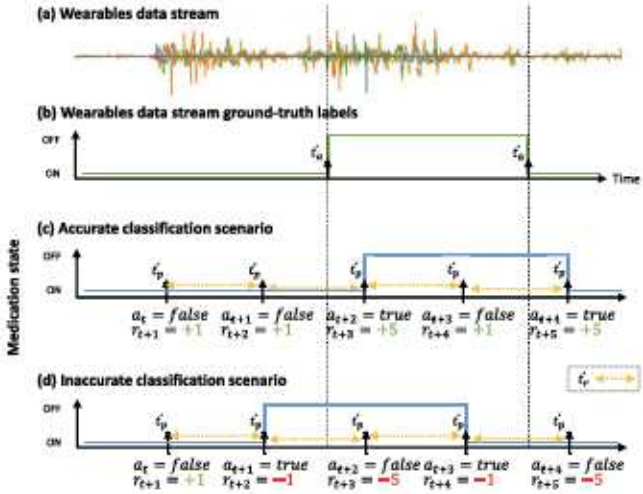

Fig. 3. (a) The wearables data stream forms the input for the model. (b) The ground-truth labels of the data stream with the actual transition time stamps at $t_a^{'}$. (c) An accurate classification scenario, and (d) an inaccurate classification case with the received rewards r_t. $t = 0, 1, 2, \ldots$: RL discrete time steps; a_t: actions at $t_p^{'}$; $t_r^{'}$: rewarding window.

each window X_w, the State Generator receives Aw values from all three anomaly detectors within the AAU. It then calculates the slope of $Aw + 1$ to the previous Aw. These slope calculations estimate both the direction (increasing or decreasing) and the rate of change in the A_{w+1} values. These variables are then concatenated and presented to the agent as the state s_t at each time step t. This approach enables the agent to learn the variations in data dynamics and the rate and direction of these variations in the patient's current state.

3) Reward Function: The agent, at t, takes an action a_t and receives a reward r_{t+1}, indicating the agent's precision in detecting transitions between medication states. Designing the reward mechanism is a crucial step in RL, as it is a primary feedback loop directing the agent's behavior and learning trajectory. In our design, action $a_t = true$ signifies that the agent detected a change in the medication state, whereas $a_t = false$ implies no such transition. The reward, r_t, is derived considering two temporal variables: $t_a^{'}$, the actual transition time in the data stream, and $t_p^{'}$, the RL time step t in the same stream. The reward r_t is mathematically:

$$r_{t+1} = \begin{cases} +5, & a_t = true \text{ and } t_p^{'} - t_a^{'} \leq t_r^{'} \\ -1, & a_t = true \text{ and } t_p^{'} - t_a^{'} > t_r^{'} \\ +1, & a_t = false \text{ and } t_p^{'} - t_a^{'} > t_r^{'} \\ -5, & a_t = false \text{ and } t_p^{'} - t_a^{'} \leq t_r^{'} \end{cases} \quad (14)$$

This equation offers a numerical assessment of the agent's capability in identifying medication state transitions. The parameter $t_r^{'}$ is instrumental in the reward strategy, defining the desired accuracy margin between the agent's prediction time $t_p^{'}$ and the actual transition time $t_a^{'}$. Fig. 3(b) and (c) provide insights into two distinct monitoring situations using our reward design, with their actual medication state labels depicted in Fig. 3(a). For instance, in Fig. 3(b), the agent accurately predicts medication state shifts at $t + 2$ and $t + 4$ and accordingly receives rewards of $+5$ and $+1$ based on equation (14). Conversely, Fig. 3(c) illustrates a scenario where the agent incorrectly detected a transition at $t + 1$ and $t + 3$ and was penalized by the reward of -1, and missed a transition at $t + 2$ and $t + 5$, which was punished a reward of -5.

C. Reinforcement Learning Design

1) Agent: The agent interacts with the environment by receiving the state s_t and taking an action a_t based on assessing whether there has been a significant change in the data stream distribution, indicating a shift in the medication state. The Q-network architecture proposed for this task comprises four consecutive layers (Fig. 1). The first layer is an input Long Short-Term Memory (LSTM) Layer to handle time-series data [29], utilizing 32 units and the hyperbolic tangent (tanh) activation function. Following, two fully connected hidden layers containing 64 neurons are attached, employing the Rectified Linear Unit (ReLU) activation function. The output layer of the Q-network is a fully connected layer with two neurons and a linear activation function. These two neurons correspond to the possible actions a_t that the agent can take. At each time step t, the input to the Q-network is the state s_t, and the output is a vector of Q values, each corresponding to a specific action a_t that the agent can select.

2) Predicted Medication State: At each time step t, the agent's action a_t determines whether the predicted medication state will remain the same or transition. If $a_t = false$, the agent indicates no change in the patient's state, and if $a_t = true$, the agent predicts a transition. In such cases, the predicted state changes from $ON \rightarrow OFF$ or $OFF \rightarrow ON$, depending on the current predicted state. Fig. 3(b) and (c) show two examples of predicted states, where the transitions between medication states occurred when $a_t = true$.

IV. Experiments

A. Datasets

Dataset PD1 [20]: This dataset comprises data from 12 individuals diagnosed with PD aged between 42 and 77 years, with disease duration ranging from 3.5 to 17 years. Data

TABLE I
A SUMMARY OF THE EMPLOYED PARKINSON'S DISEASE (PD) DATA: DATASET PD1 AND DATASET PD2

Characteristics	Dataset PD1	Dataset PD2
Number of subjects	12	7
Sex (m, f)	9, 3	5, 2
Age (year)	55.8 ± 9.9	58.7 ± 8.0
Disease duration (year)	8.4 ± 3.6	10.6 ± 2.3
UPDRS-III	22.5 ± 12.5	20.3 ± 12.6
Tremor	1.2 ± 2.7	2.8 ± 4.0
Bradykinesia	13.7 ± 7.5	10.6 ± 6.1
Dyskinesia	3.7 ± 3.9	N/A

collection utilized KinetiSense motion sensors from Great Lakes NeuroTechnologies Inc., which recorded tri-axial accelerometer and gyroscope data at a sampling rate of 128 Hz. Each participant wore two sensor units, one on the most affected wrist and another on the ankle. The participants refrained from taking antiparkinson medication the night before the experiment, starting the data collection in their OFF state. The participants performed seven ADLs: walking, resting, cutting food, dressing, drinking, unpacking groceries, and brushing hair. Afterward, participants resumed their medication and repeated the activities in their ON state. Neurologists conducted clinical examinations to assess the medication state, the Unified Parkinson Disease Rating Scale part III (UPDRS-III), tremor, bradykinesia, and the modified Abnormal Involuntary Movement Scale (mAIMS) scores to characterize dyskinesia-related complications. The average data duration for each subject was 14 minutes with 65% in the medication ON state.

Dataset PD2 [21]: This dataset includes data from seven individuals diagnosed with PD, aged between 48 and 68, with disease duration ranging from 6 to 15 years. Two Kinesia motion sensor units from Great Lakes NeuroTechnologies were placed on each participant's wrists and ankles to capture accelerometer and gyroscope data at a 64 Hz. The participants performed six activities of hygiene-related tasks (brushing hair or teeth), dressing, eating, desk work, and entertainment (watching TV or reading). Participants cycled through these stations for two hours. Like the PD1 dataset, participants began the study in their OFF state and visited each activity station at least once. They then resumed their regular medication and repeated the cycle of activities at the stations. Once they transitioned into their ON state, confirmed through clinical examination and self-reports, they revisited all the activity stations at least once more. Neurologists conducted clinical examinations to assess the medication state, UPDRS-III, tremor, and bradykinesia. 76% of the data was collected in the medication state OFF state.

Table I summarizes the patient characteristics in the two PD datasets. The data collection for both datasets was conducted with the approval of the institutional review boards of Great Lakes NeuroTechnologies, the University of Rochester, and Johns Hopkins University. All participants provided informed consent, and the studies adhered to the Declaration of Helsinki.

B. Data Preprocessing and Feature Extraction

The Dataset PD1 was down-sampled to 64 Hz to match the Dataset PD2 sampling rate. Next, the signal was passed through a finite impulse response (FIR) filter with a pass frequency between 0.5-15 Hz. After noise elimination, the data were segmented into 5 s windows with an overlap of 4 s. This window length was selected empirically (supplementary Fig. S1) and based on previous research that demonstrated its effectiveness in encapsulating the symptomatic expressions of PD [30]. We extracted hand-crafted features (supplementary Table S1), which have been established in the literature as robust indicators of PD motor symptoms [14].

C. Experiment Setup

Within-Domain Testing: In our initial experiments, we use Dataset PD1 to assess our framework within the PD patient data domain. This allows us to gauge the model's adaptability to PD-related motion signals. We employed a leave-one-out approach: iteratively using one subject's data for testing and the rest for training. This was done for each subject to obtain average results. Five-fold cross-validation optimized the model on the training data. We replicated this within-domain testing setup on Dataset PD2 for further evaluation.

Cross-Domain Testing: After experimenting with Dataset PD1, we used the model trained on its entire set to monitor individuals in Dataset PD2, maintaining the same hyperparameters. This phase embodies cross-domain testing, evaluating the model on a distinct dataset. The aim is to gauge the model's efficacy in a new and larger data domain. Significantly, Dataset PD2 is five times larger than PD1, testing the model's capacity to generalize from smaller to larger datasets. We also assessed the model's performance in generalizing from larger to smaller datasets, from Dataset PD2 to Dataset PD1.

Implementation: TThe RL agent underwent training across 1,000 episodes using the epsilon-greedy method. It started with an epsilon value of $\epsilon = 1.0$, which decreased until $\epsilon = 0.01$, a strategy proved effective [24]. The Q-learning equation had a discount factor, $\gamma = 0.5$, balancing current and future rewards. In Dataset PD1, nine participants transitioned from medication OFF to ON states, with three experiencing multiple transitions. The agent training was conducted on an extended data stream for each participant, ensuring the agent faced both transition directions (from OFF to ON and vice versa). This extended data stream replicated the original data stream.

Evaluation Metrics: We used four metrics to gauge our approach: accuracy, sensitivity, specificity, and weighted F1-score. Sensitivity and specificity measure the identification of OFF and ON medication states, respectively. The weighted F1-score averages the F1-scores for each label, adjusted by the number of true instances, addressing class imbalances.

Comparative Methods To assess our model, we compared it with leading machine learning techniques for medication state monitoring. We benchmarked against an SVM [14], a data-augmented CNN [15], and a CNN [16]. Additionally, we explored employing a Gated Recurrent Unit (GRU) layer as the input layer of the agent's Q-network to handle the time-series

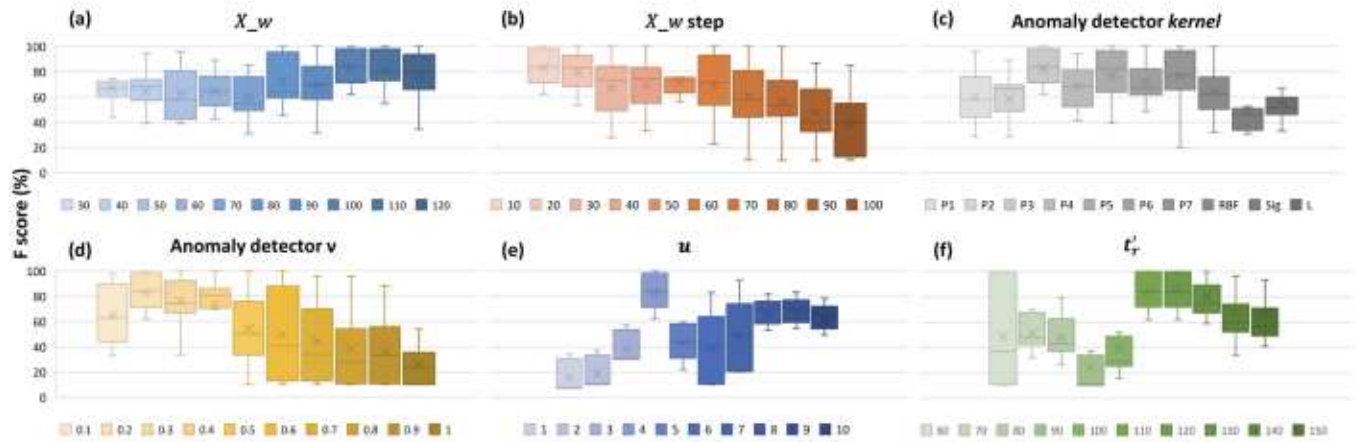

Fig. 4. Dataset PD1 cross-validation weighted F-score obtained by the proposed model for six hyperparameters: (a) sliding window X_w size, (b) sliding window X_w step size, (c) anomaly detectors kernel function (P: polynomial function with the number signifying the degree, RBF: radial bias function, Sig: sigmoid), (c) anomaly detectors ν value, (e) value of u, and (f) rewarding window size $t_r^{'}$.

data. The SVM model used the same extracted features. The CNNs used the pre-processed wrist and ankle raw accelerometer and gyroscope data.

D. Model Hyperparameters Optimization

Our methodology involves six hyperparameters: sliding window size X_w and X_w step, anomaly detector kernel type, ν, u in the RL framework, and reward window $t_r^{'}$. The ν parameter constrains SVM training errors and support vector fraction [28]. We adopted the Bayesian optimization approach [31] to optimize these hyperparameters and maximize the model's weighted F1-score output for Dataset PD1. The search space for hyperparameters was defined as follows: $X_w \in [30, 120]s$, $X_w step \in [10, 100]s$, kernel type = [linear, polynomial $(degree \in [1, 7])$, radial bias function (RBF), sigmoid], $\nu \in (0, 1.0]$, $t_r^{'} \in [60, 150]s$. After performing Bayesian optimization for 50 trials, the optimal hyperparameters found were $X_w = 100\,s$, $X_w step = 10\,s$, kernel type = polynomial $(degree = 3)$, $\nu = 0.2$, $t_r^{'} = 110\,s$.

E. Model Hyperparameters Analysis

We conducted a sensitivity analysis on the hyperparameters to evaluate their impact on the model's behavior and performance, inspired by [32]. Then, we employed Morris sensitivity analysis [33] to discern the influence of individual hyperparameters and their interactions on model performance. This analysis aids in fine-tuning the model for implementation in various applications. We initiated the analysis using default optimized values for Dataset PD1. The sensitivity analysis was conducted based on the model's weighted F1-score.

1) Sliding Window Hyperparameters: Fig. 4(a) displays the effects of changing the sliding window size (X_w) on performance. As it grows, the model performs better. After reaching an optimal point, the performance drops with an overly long window. X_w selection must ensure adequate data duration to recognize PD patient movement changes due to medication. Fig. 4(b) presents the analysis of X_w step sizes. Predictably, larger step sizes diminish model performance. Appropriate step size is critical to providing more regular and detailed data analysis for identifying medication state transitions.

2) Anomaly Detectors Hyperparameters: Fig. 4(c) assesses different kernel functions for the anomaly detectors. A 3rd-degree polynomial function emerged as the most effective for medication state identification, suggesting its capability to understand the data's inherent patterns. The parameter ν was scrutinized, as it limits the fraction of training mistakes and plays a pivotal role in the RL agent's learning stability. Larger ν values, especially when $\nu \geq 0.5$, permit more training errors during anomaly detectors' updates. Such errors, however, might disrupt the RL agent's learning since they're reflected in state s and influence the agent's choices.

3) RL Framework Hyperparameters: Fig. 4(e) showcases the effect of adjusting the value of u. Utilizing smaller u values, specifically when $u < 4$, offered restricted environmental insights for constructing a comprehensive state s. This restriction negatively affected the agent's learning ability, resulting in a decline in the model's efficiency. On the other hand, a larger u (i.e., $u > 4$) delayed the agent's recognition of shifts in the data stream's behavior, leading to suboptimal outcomes. Additionally, the length of the rewarding window ($t_r^{'}$) underwent evaluation. While smaller $t_r^{'}$ dimensions were favored for rigorous training, excessively tight rewarding windows resulted in inconsistent training patterns. For instance, as depicted in Fig. 4(f), diminished $t_r^{'}$ dimensions caused the agent's learning to waver, causing a notable drop in efficiency. This occurrence is linked to the extended period required for medication states to become evident in the dataset. Hence, opting for a more restrictive rewarding window places unreasonable demands on efficiency.

4) Hyperparameter Ranking: The Morris sensitivity analysis technique was utilized to assess the hyperparameters, and they were subsequently ranked based on their impact on performance. The analysis's results produce values for $\mu^* \in [0, 1]$ and $\sigma \in [0, 1]$. The former metric, μ^*, offers insight into the overarching effect a specific parameter exerts on the model's

TABLE II
'WITHIN-DOMAIN' PERFORMANCE (%) ON DATASET PD1

Method	Accuracy	Sensitivity	Specificity	F1-score
Proposed (*utilizing deep LSTM Q-network*)	**82.71** ± 14.19	**85.98** ± 20.74	**83.89** ± 17.47	**82.94** ± 14.14
Proposed (*utilizing deep GRU Q-network*)	72.29 ± 21.14	72.40 ± 27.81	77.71 ± 21.21	72.47 ± 21.20
Hssayeni *et al.* [14]	73.24 ± 17.27	63.44 ± 29.57	76.68 ± 22.09	72.72 ± 17.46
Um *et al.* [15]	61.62 ± 9.41	44.61 ± 17.14	71.43 ± 11.41	61.46 ± 10.51
Pfister *et al.* [16]	60.49 ± 9.49	38.88 ± 10.61	72.88 ± 9.86	60.24 ± 10.18

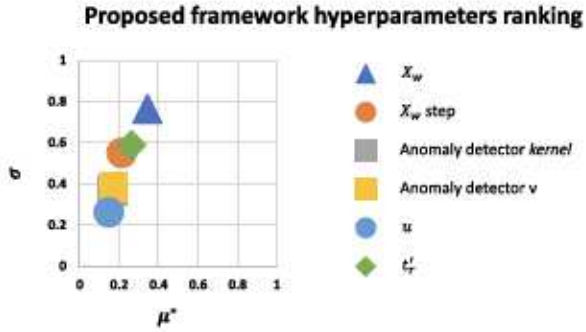

Fig. 5. The measures, μ^* and σ, from Morris method for model hyperparameters: value of u, rewarding window size t_r', sliding window X_w size, sliding window X_w step size, anomaly detectors kernel function, and anomaly detectors ν. The analysis utilized Dataset PD1.

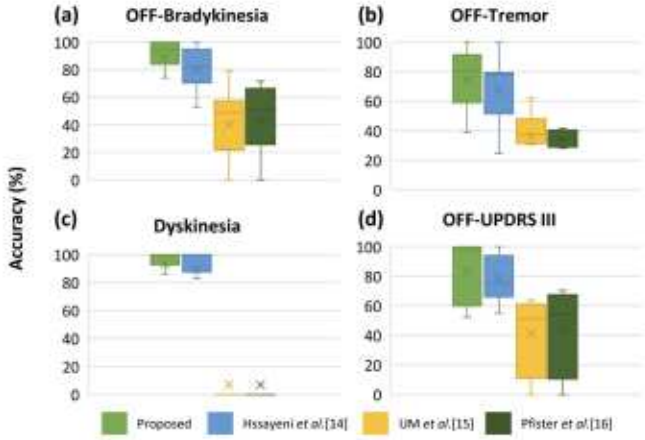

Fig. 6. Medication state accuracy of patients while displaying severe symptoms characterized by the scores of (a) OFF-Bradykinesia, (b) OFF-Tremor, (c) ON-Dyskinesia (mAIMS), and (d) OFF-UPDRS-III.

output. A value of 1 signifies maximum influence. In contrast, the latter metric, σ, illustrates how interdependent the parameter is with other parameters. A value of 1 here denotes the peak level of dependency. The graphical representation in Fig. 5 exhibits both μ^* and σ. Drawing conclusions from this analysis, it becomes evident that the sliding window X_w stands out as the hyperparameter wielding the most substantial influence on the model's efficacy and exhibits the highest interdependence with other parameters. Given the nature of its application, the pronounced influence and dependency of X_w were anticipated. Other hyperparameters, including the step size of X_w, and rewarding window t_r', display comparably significant impacts on the model's performance and their mutual influences. In contrast, the kernel function, anomaly detector parameters, ν, and u showcase the least influence and interdependence. Considering that ν and u govern the error rates of the anomaly detectors and the volume of time windows, it is logical for them to operate more autonomously, irrespective of the dataset type and other hyperparameters.

V. RESULTS

A. Within-Domain

In the Within-Domain phase, we evaluate the model's performance when trained and tested on the subjects in Dataset PD1. Table II shows the models' average testing performance for all 12 subjects. The initial observation from the evaluation is that the proposed adaptive classifier outperformed the other comparative models across all four evaluation metrics. One notable finding was that the comparative models exhibited low sensitivity scores, indicating a bias towards the majority class in the data, the medication ON state, during training. In contrast, the adaptive classifier did not display such bias. The adaptive classifier utilizing an LSTM input layer for the Q-network demonstrated a notably higher average testing sensitivity, scoring 13.58% higher than the proposed utilizing GRU input layer and 22.54% higher than the best state-of-the-art comparative model. This outcome indicates that the adaptive classifier correctly identified the medication state OFF in patients. Additionally, the adaptive classifier demonstrated the highest specificity score, signifying its ability to detect the medication ON state accurately. The average testing weighted F-score of the proposed adaptive classifier at 82.94% was higher than the other comparative methods, highlighting its ability to provide a balanced F-score across different classes, considering class imbalances in the data. The Within-Domain evaluation results for Dataset PD2 are presented in supplementary Table S2, demonstrating the superior performance of our proposed framework compared to the comparative models.

Next, we investigated the model's ability to monitor motor fluctuations in patients, specifically during pronounced disease symptoms such as OFF-Bradykinesia, ON-Dyskinesia (mAIMS), OFF-Tremor, and OFF-UPDRS-III scores. We opted to utilize the model employing an LSTM for the Q-network input, as it has demonstrated outstanding performance compared to when using a GRU layer. The model's detection accuracy was measured when symptom scores crossed certain thresholds: Bradykinesia $\geq$ 20, Dyskinesia $\geq$ 5, Tremor $\geq$ 2, and UPDRS-III $\geq$ 33. These thresholds were set based on the average scores for each symptom in its respective medication state. The findings are showcased in Fig. 6, juxtaposed with other models. The adaptive classification framework consistently surpassed the other models. When patients showed Bradykinesia in the OFF

TABLE III
'CROSS-DOMAIN' PERFORMANCE (%) ON DATASET PD2

Method	Accuracy	Sensitivity	Specificity	F1-score
Proposed (*utilizing deep LSTM Q-network*)	**78.39** ± 14.69	**84.64** ± 16.31	67.69 ± 39.43	**76.67** ± 16.77
Proposed (*utilizing deep GRU Q-network*)	68.08 ± 19.76	65.46 ± 14.97	68.60 ± 43.64	70.74 ± 19.36
Hssayeni *et al.* [14]	47.31 ± 23.08	30.28 ± 29.98	**100** ± 0	43.15 ± 29.94
Um *et al.* [15]	56.69 ± 10.45	45.60 ± 8.98	69.13 ± 17.15	56.24 ± 9.48
Pfister *et al.* [16]	56.44 ± 11.95	42.88 ± 8.45	71.65 ± 15.29	57.24 ± 10.80

The highest performance under each metric is in bold.

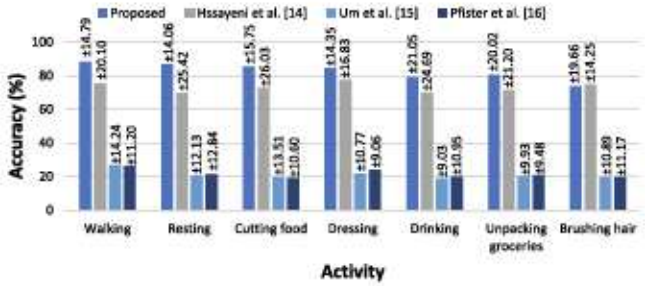

Fig. 7. The accuracy (± standard deviation) in monitoring the PD medication states while performing seven activities of daily living.

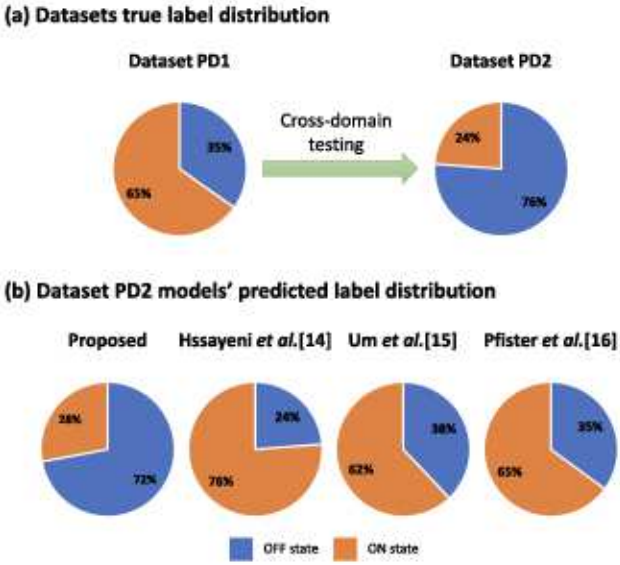

Fig. 8. (a) The true label distribution for the medication states across Datasets PD1 and PD2, illustrating the inherent variability in class occurrence due to the nature of data collection. (b) The distribution of the models' predicted labels in cross-domain testing when the proposed model and the three comparative models are trained on Dataset PD1 and applied on Dataset PD2, emphasizing the models' ability to generalize and manage label disparities under real-world conditions.

state, the model recorded an accuracy of 88.70%, highlighting its reliability during slow movements. The model also performed well during involuntary, dyskinetic movements, with accuracy rates of 84.01% and 83.61% during high UPDRS-III scores. Despite Tremor symptoms influencing the model, it still achieved an accuracy of 74.50%.

Furthermore, we assessed the model's performance during various ADLs. This evaluation highlighted the model's adaptability across diverse daily routines and its capability in medication state detection in real-world settings. Fig. 7 illustrates the accuracy as patients undertook seven ADLs.

B. Cross-Domain

We assessed the model's generalizability by applying it to the seven subjects in Dataset PD2 after training it on the data from all 12 subjects in Dataset PD1. Table III summarizes the average performance metrics obtained from Dataset PD2, reflecting the model's ability to generalize across different patient cohorts. The framework employing an input LSTM layer for the Q-network achieved a 76.67% accuracy, surpassing other methods in monitoring PD medication states. It also had the top testing sensitivity of 84.64%, emphasizing its capability to detect the PD medication state OFF. Other models lagged in detecting the OFF state. The SVM-based [14] classifier's 100% specificity was influenced by Dataset PD2's OFF state majority (Fig. 8) and a training bias from Dataset PD1. The adaptive classifier balanced classifying both states, with its weighted F-score being 19.43% higher than other models, showcasing its balanced performance amidst class imbalances. The Cross-Domain evaluation results, where the models were trained on Dataset PD2 and tested on all subjects in Dataset PD1, are presented in supplementary Table S3. Our proposed framework exhibited outstanding performance compared to the comparative models.

Our study emphasizes the model's capability to tackle data class imbalance. Fig. 8 shows significant class imbalance in Datasets PD1 and PD2. Such imbalance can affect machine learning model performances, especially in cross-domain testing. While comparative models showed a bias to Dataset PD1's majority class when tested on Dataset PD2, the proposed model employing an LSTM for the Q-network input was resilient. On applying to Dataset PD2, it classified 72% as medication state OFF and 28% as ON, showcasing its adaptability to different class distributions.

VI. DISCUSSION

In the rapidly evolving landscape of PD research, the challenge of accurately monitoring medication states remains paramount [17]. We addressed this by introducing an RL-based framework specifically designed for detecting medication ON and OFF states using wearable sensor data during daily activities. This approach not only captures dynamic PD symptoms but also uses the rich information from real-world data. With RL, our model continuously refines its predictions, offering a

TABLE IV
MEDICATION STATE MONITORING METHODS COMPARISON

Reference	Sensors	Activities	Monitoring approach	Individualized approach	Results
Pérez-López *et al.* [10]	Waist	Walking, non-walking	Identify dyskinesia and bradykinesia	No	Sen: 99.90%
Ossig *et al.* [11]	Wrist	At-home	Threshold on PKG data	Yes	Corr: 0.658
Rodríguez-Molinero *et al.* [12]	Waist	Walking	Identify dyskinesia and bradykinesia	Yes	Acc: 92.20%
Fisher *et al.* [13]	Waist	Free-living, at-home	Identify medication state OFF	No	Sen: 60%
Hssayeni *et al.* [14]	Wrist and ankle	ADL	Identify medication state ON/OFF	Yes	Acc: 90.50%
Pfister *et al.* [16]	Wrist	Free-living	Identify medication state ON/OFF/Dyskinetic	No	Balanced Acc: 65.40%
Um *et al.* [15]	Wrist	Free-living; excluding walking, lying, and eating	Identify medication state ON/OFF	No	Acc: 86.88%
Proposed	Wrist and ankle	ADL	Identify medication state transitions	No	F1: 82.94%

real-time representation of medication states. This represents a significant advancement in PD monitoring.

A key finding is our model's ability to detect intricate patterns indicating transitions between medication states. This was evident in its performance on Dataset PD1 and Dataset PD2, surpassing other benchmarks as shown in Table II and supplementary Table S2, respectively. Its capability was further observed during severe symptom manifestation and when patients were engaged in specific ADLs, as shown in Figs. 6 and 7 respectively. Our model also excelled in cross-domain testing, especially when trained on Dataset PD1 and tested on Dataset PD2, as illustrated in Table III. It also showed a remarkable performance when trained on the larger Dataset PD2 and tested on the smaller Dataset PD1 as demonstrated in Table S3. While other models showed bias towards Dataset PD1, ours remained resilient, showcasing its robustness. Crucially, for real-world applications, it is imperative that models demonstrate generalizability to new datasets, ensuring consistent and trustworthy performance across diverse conditions [34].

The model's ability to handle imbalanced training data sets it apart. Despite the inherent imbalances in our datasets (see Fig. 8), our model achieved commendable weighted F1 scores in both within and cross-domain tests. Addressing imbalanced training data remains a formidable challenge in machine learning applications, and our approach signals a promising advancement in this domain, potentially revolutionizing how we handle and interpret skewed datasets [35].

Further investigations of our model's performance in the sensitivity analysis, particularly the Morris analysis, brought our next interesting insights. A standout observation was the paramount importance of the hyperparameter X_w window length, as indicated by its highest μ^* value. This emphasizes that when venturing into new applications or fine-tuning the model, the foremost priority should be to adjust the X_w window length. Following this, refining the X_w step size, t_r', and deciding on the optimal AAU kernel function and ν would be the logical next steps. On the flip side, hyperparameters like u, which recorded the lowest μ^* values, can be relegated to the latter stages of the fine-tuning process.

Previous research investigated medication state monitoring in PD patients using wearable sensors and are listed in Table IV. Some studies have employed statistical methods in their research. For instance, Pérez-López et al. [10] achieved a sensitivity of 99.90% in detecting medication states by identifying dyskinesia and bradykinesia *during walking* using a waist sensor. Ossig et al. [11] analyzed the spectral power of low-frequency wrist accelerometer data from the PKG to monitor three medication state categories. Their correlation with patients' diaries ranged from 0.404 to 0.658. Rodríguez-Molinero et al. [12] used waist sensor data to detect *walking activity* and then determine medication states as ON when dyskinesia is present and OFF when bradykinesia is detected, achieving an accuracy of 92.20%. Few studies have targeted *medication state classification during ADLs*, leading to reduced classification performance as anticipated. Fisher et al. [13] utilized an ANN in combination with a waist sensor to achieve a sensitivity of 60% when detecting medication state OFF. Hssayeni et al. [14] used an individualized SVM with wrist and ankle sensors, achieving an average accuracy of 90.50% for 24 subjects. Pfister et al. [16] achieved a balanced accuracy of 65.40% using a CNN model and wrist wearable sensor data of 30 patients. Um et al. [15] reported an accuracy of 86.88% employing a data-augmented CNN model. The data was collected from a wrist sensor worn by 25 patients, excluding walking, lying, and eating data.

While our method's 82.94% F1-score is not the absolute highest, our method signifies a series of advancements over previous research. Our method excels in real-world applicability and is evaluated across a diverse range of ADLs, unlike previous studies focusing on narrow, controlled activities. The core innovation of our model is its dynamic capability to capture and learn from the transitions between medication states. This dynamic approach represents a leap over conventional models, typically limited to detecting isolated events such as dyskinesia or bradykinesia. Our study balances personalization and scalability, fine-tuning our model to each patient's disease profile and severity while maintaining the capability to generalize across the PD population. Therefore, the distinction of our study is not confined to its performance metrics alone but is also rooted in enhancing the applicability, generalizability, and relevance of PD state monitoring in real-life conditions. This broader relevance provides a comprehensive view of a patient's medication fluctuation profile, vital for holistic treatment evaluations [17].

Direct comparison with some of the aforementioned methods was not feasible due to their reliance on proprietary or private

datasets. However, to ensure the robustness and validity of our approach, we implemented three comparable methods from the literature that were intended for detecting medication states during different ADLs. When applying these methods to our datasets, the performance results were closely aligned with those reported in their original publications, albeit with a slight decline. For instance, the accuracy of [14] at 73.24% is somewhat lower than their original findings, which can be attributed to the individualized approach they adopted in their study. Similarly, the 60.49% accuracy achieved by [16] aligns with their original results. In the case of [15], their 61.62% accuracy was somewhat diminished compared to their initial study, possibly due to the exclusion of certain activities in their methodology. One notable observation is the higher accuracy and F1 score of our proposed method, not only in within-domain but also in cross-domain scenarios, supporting the superiority of our RL-based framework over these methods in terms of performance and adaptability.

Collectively, these findings support our hypothesis and emphasize the efficacy of our proposed framework in monitoring PD. The versatility and adaptability of our framework are evident, making it an invaluable tool for real-world PD monitoring scenarios. By integrating this adaptive classification framework into health monitoring applications, we can significantly enhance therapy adjustments. This is achieved by providing healthcare providers and physicians with timely and crucial information about the patients' health states. It is essential to highlight that while the model does necessitate knowledge of the initial class label, its primary objective in health monitoring applications remains the detection of changes in system dynamics. This focus on change detection reduces the dependence on initial labels, facilitating effective adaptation. The insights from this study hold the potential to transform PD research and monitoring, ushering in a new era of more personalized and patient-focused treatment methodologies in the future.

Our research offers significant contributions to the monitoring of medication states in PD patients, yet it is important to recognize its limitations. Primarily, the study was conducted with a select group of PD patients, suggesting the need for further research to confirm our model's applicability across a wider range of patients and varying stages of the disease. Despite our efforts to ensure robustness through cross-domain testing, differences between datasets could potentially affect the model's performance, highlighting an area for future exploration, particularly in domain adaptation techniques. Additionally, the application of RL, while innovative, necessitates considerable data for training and substantial computational power. Future work will aim at refining the model and its training processes to mitigate these demands. Addressing these aspects will further enhance the practical utility and impact of our methodology in diverse clinical environments.

VII. Conclusion

We introduced a novel framework grounded in RL principles to adaptively monitor transitions between medication ON and OFF states in PD patients. Rather than merely learning the data patterns distinct to each state, our approach hinges on the RL agent's ability to discern data dynamics during transitions. This shift in perspective, combined with the deep LSTM neural network, allows the agent to capture the intricate nuances of these state changes more effectively. Additionally, by integrating three one-class unsupervised classifiers into our model, we further improved its capability to identify the transitioning between states. We evaluated our framework using two PD datasets, which comprised data from 19 subjects equipped with wrist and ankle wearable sensors, which provided a unique window into the ADLs of PD individuals, especially as they transitioned between ON and OFF states. Our results proved the robustness and adaptability of the proposed approach with an average weighted F1-score of 82.94% was achieved when training and testing on Dataset PD1 and 76.67% when training on Dataset PD1 but testing on Dataset PD2. Furthermore, our method's effectiveness was underscored when benchmarked against three existing techniques. Our approach's primary contribution lies in its potential for accurate medication state monitoring and its generalizability across different domains, which can lead to more tailored therapeutic adjustments in real-world applications with varied sensors. This, in turn, promises an enhanced quality of life for PD patients. We are optimistic that our novel framework will inspire more advancements in this field, aiming to bring real-world benefits to PD patients globally.

References

[1] C. A. Davie, "A review of Parkinson's disease," *Brit. Med. Bull.*, vol. 86, no. 1, pp. 109–127, 2008.
[2] R. B. Dewey, "Management of motor complications in Parkinson's disease," *Neurology*, vol. 62, no. 6_suppl_4, pp. S3–S7, 2004.
[3] J. Jankovic, "Motor fluctuations and dyskinesias in Parkinson's disease: Clinical manifestations," *Movement Disord.*, vol. 20, no. S11, pp. S11–S16, 2005.
[4] W. Maetzler, J. Klucken, and M. Horne, "A clinical view on the development of technology-based tools in managing Parkinson's disease," *Movement Disord.*, vol. 31, no. 9, pp. 1263–1271, 2016.
[5] E. E. Tripoliti et al., "Automatic detection of freezing of gait events in patients with Parkinson's disease," *Comput. Methods Programs Biomed.*, vol. 110, no. 1, pp. 12–26, 2013.
[6] D. Ravì et al., "Deep learning for health informatics," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 1, pp. 4–21, Jan. 2017.
[7] M. D. Hssayeni, J. Jimenez-Shahed, M. A. Burack, and B. Ghoraani, "Wearable sensors for estimation of Parkinsonian tremor severity during free body movements," *Sensors*, vol. 19, no. 19, 2019, Art. no. 4215.
[8] M. D. Hssayeni, J. Jimenez-Shahed, M. A. Burack, and B. Ghoraani, "Dyskinesia estimation during activities of daily living using wearable motion sensors and deep recurrent networks," *Sci. Rep.*, vol. 11, no. 1, 2021, Art. no. 7865.
[9] M. D. Hssayeni, J. Jimenez-Shahed, M. A. Burack, and B. Ghoraani, "Ensemble deep model for continuous estimation of unified Parkinson's disease rating scale III," *Biomed. Eng. Online*, vol. 20, pp. 1–20, 2021.
[10] C. Pérez-López et al., "Monitoring motor fluctuations in Parkinson's disease using a waist-worn inertial sensor," in *Proc. Adv. Comput. Intell.*, 2015, pp. 461–474.
[11] C. Ossig et al., "Correlation of quantitative motor state assessment using a kinetograph and patient diaries in advanced PD: Data from an observational study," *PLoS One*, vol. 11, no. 8, 2016, Art. no. e0161559.
[12] A. Rodríguez-Molinero et al., "A kinematic sensor and algorithm to detect motor fluctuations in Parkinson disease: Validation study under real conditions of use," *JMIR Rehabil. Assistive Technol.*, vol. 5, no. 1, 2018, Art. no. e8335.
[13] J. M. Fisher, N. Y. Hammerla, T. Ploetz, P. Andras, L. Rochester, and R. W. Walker, "Unsupervised home monitoring of Parkinson's disease motor symptoms using body-worn accelerometers," *Parkinsonism Related Disord.*, vol. 33, pp. 44–50, 2016.

[14] M. D. Hssayeni, M. A. Burack, J. Jimenez-Shahed, and B. Ghoraani, "Assessment of response to medication in individuals with Parkinson's disease," *Med. Eng. Phys.*, vol. 67, pp. 33–43, 2019.
[15] T. T. Um et al., "Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks," in *Proc. 19th ACM ICMI*, 2017, pp. 216–220.
[16] F. M. Pfister et al., "High-resolution motor state detection in Parkinson's disease using convolutional neural networks," *Sci. Rep.*, vol. 10, no. 1, 2020, Art. no. 5860.
[17] B. Ghoraani, J. E. Galvin, and J. Jimenez-Shahed, "Point of view: Wearable systems for at-home monitoring of motor complications in Parkinson's disease should deliver clinically actionable information," *Parkinsonism Related Disord.*, vol. 84, pp. 35–39, 2021.
[18] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1, pp. 151–175, 2010.
[19] M. Shuqair, B. Ghoraani, and J. Jimenez-Shahed, "Incremental learning in time-series data using reinforcement learning," in *Proc. IEEE Int. Conf. Data Mining Workshops*, 2022, pp. 868–875.
[20] T. O. Mera, M. A. Burack, and J. P. Giuffrida, "Objective motion sensor assessment highly correlated with scores of global levodopa-induced dyskinesia in Parkinson's disease," *J. Parkinson's Dis.*, vol. 3, no. 3, pp. 399–407, 2013.
[21] C. L. Pulliam, D. A. Heldman, E. B. Brokaw, T. O. Mera, Z. K. Mari, and M. A. Burack, "Continuous assessment of levodopa response in Parkinson's disease using wearable motion sensors," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 1, pp. 159–164, Jan. 2018.
[22] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[23] C. J. C. H. Watkins, *Learning From Delayed Rewards*. Cambridge, U.K.: King's College, 1989.
[24] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[25] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLoS One*, vol. 11, no. 4, 2016, Art. no. e0152173.
[26] E. R. Faria, I. J. Gonçalves, A. C. de Carvalho, and J. Gama, "Novelty detection in data streams," *Artif. Intell. Rev.*, vol. 45, no. 2, pp. 235–269, 2016.
[27] M. M. Moya and D. R. Hush, "Network constraints and multi-objective optimization for one-class classification," *Neural Netw.*, vol. 9, no. 3, pp. 463–474, 1996.
[28] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.
[29] S. Carta, A. Ferreira, A. S. Podda, D. R. Recupero, and A. Sanna, "Multi-DQN: An ensemble of deep q-learning agents for stock market forecasting," *Expert Syst. Appl.*, vol. 164, 2021, Art. no. 113820.
[30] S. Patel et al., "Monitoring motor fluctuations in patients with Parkinson's disease using wearable sensors," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 6, pp. 864–873, 2009.
[31] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, Jan. 2016.
[32] S. Fu, S. Zhong, L. Lin, and M. Zhao, "A novel time-series memory auto-encoder with sequentially updated reconstructions for remaining useful life prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 7114–7125, Dec. 2022.
[33] F. Campolongo, J. Cariboni, and A. Saltelli, "An effective screening design for sensitivity analysis of large models," *Environ. Modelling Softw.*, vol. 22, no. 10, pp. 1509–1518, 2007.
[34] J. Wang et al., "Generalizing to unseen domains: A survey on domain generalization," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 8052–8072, Aug. 2023.
[35] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.