21AIE401_Deep Reinforcement Learning

End Semester project presentation – Evaluation Sheet

Team No: **Team Name:**

PROJECT TITLE:

Team Member Reg. no.				
Team No Name				
Implementation	/15	/15	/15	/15
Hyper parameter tuning and results	/5	/5	/5	/5
Report and project video	/5	/5	/5	/5
VIVA	/5	/5	/5	/5
MARK	/30	/30	/30	/30

CHECKLIST Insert a tick in each cell in the table before the presentation

Documents submitted	**Introduction**	**Literature Survey**	**System Architecture**	**Implementation**	**Results and Analysis**	**CODE**
PPT(Soft/Hard)	**Conclusion and Future Scope**	Timeline Diagrams	Overall architecture	Each Module architecture - Figures	Training Parameters- Tables and plots	Custom Environment/Existing Environment
VIDEO(Soft)	**References**	Comparison Tables	DRL Formulation	Class Diagram	Hyperparameter training – Tables and plots	MATLAB/PYTHON AGENT DESIGN
Report (Soft/Hard)		Summary of all the papers	Algorithm	Description of all function modules	Comparison with respect to literature survey papers – summary table	TRAINING HYPERPARAMETER TUNING AGENT PERFORMANCE
CODE(Soft)		Your specific contribution to the literature	Design points	Code Explanation	Comparing with test cases the agent performance	

Additional Comments:

Signature of the faculty

Reinforcement Learning-Based Telehealth System for Automated Patient Health Monitoring

Bommala Deepak Kumar
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Bengaluru,India
deepakbommala@gmail.com

Kaivaram Sivarama Krishna
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Bengaluru,India
sivaramak214@gmail.com

Morumpalli Deekshith Reddy
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Bengaluru,India
deekshithreddy940@gmail.com

Bheema Sai Varun
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Bengaluru,India
saivarun2024@gmail.com,

Amudha J
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Bengaluru,India
j_amudha@blr.amrita.edu

***Abstract*—This work gives a linearized telehealth monitoring system that is run by Deep Reinforcement Learning (DRL) for managing patient care automatically. The framework consists of an ensemble of Deep Q-Network based (DQN) agents to complete a series of tasks in order; data preprocessing, health data abnormality detection, treatment suggestions, and subsequent examination monitoring. The system uses a neural network structure to process nonlinear and high-variance patient measurements and make clinical decisions in real-time. Replay buffer restores events during training, making learning to be efficient in this aspect. This architecture is aimed at enhancing the health care delivery system by automating some of these intercessions besides monitoring. The findings reveal that the system has high efficacy in the prediction of anomalies, determination of the best treatments' ways, and delivery of individualized care that is beneficial to patients. .**

***Index Terms*—Telehealth, Health care, RL, Patient data, DQN , Q Learning , Agent, Environment, Action Space, Gradio, Epsilon Greedy.**

I. Introduction

Health care systems demand to be efficient and prompt therefore are experiencing increasing demand given the current practice and adoption of remote care, telehealth solutions. These systems are mostly running manual interventions so it is hard to perform continuous monitoring and drastic changes of patient states. These systems face several challenges, such as: Massive data sets of general health which can be captured by several sensors at once. Situational responses, unpredictable and can change frequently in a matter of days.Due to its state-dependency, adaptive learning requires a way to suggest treatments and track patients' changes in status. These challenges can be best solved with the help of DRL because of its ability to incorporate neural networks with sequential decision-making models. The proposed multi-agent DQN-based telehealth monitoring system is introduced in this project. Every agent of the system has its own role: data pre-processing of health records, discovering abnormality, or facilitating recommendations for treatments. The DRL allows the agents to learn from feedback at each time step so as to incrementally achieve the best actions. The buffers improve stability in replay by repeating prior experiences in training in order to make learning effective.

Need for the Problem

One of the issues is the creation of the RL models that would be usable in the setting of synchronous tele rehabilitation. This task requires fast and reliable mechanisms for analysis of continuous streams and support for real-time recommending. Additionally, DRL can be employed in the sphere of healthcare especially since the state of a person's health largely depends on personal features and the DRL model has to work on the basis of specifics of a particular person and his or her history while using the data for the analysis and making forecasts and recommendations which is even more complicates the system.

This is because Risk-less Learning strategies in healthcare should be according to ethical standards and rules observed in healthcare. It is imperative to discuss the need for ethical standards and the issue of confidentiality with regard to patients. The key contributions of this work are: 1. Automating the process of treatment 2. Developing the AI in healthcare In Section 2, a brief of Literature survey is mentioned. In Section 3, A detailed explanation of the DRL formulation is discussed. In section 4 the system architecture is explained. In section 5 the implementation of the model is given. The results are presented in Section

II. Literature Survey

[1]Using wearable sensors, cloud computing, and AI, this paper focus on RPM for managing chronic diseases with impaired physiological signs. Another key area is the role

of RPM, and how the programme can increase the efficiency of care delivery and decrease the cost across the system; At the same time, the report discusses issues surrounding data protection and integration, as well as the finite nature of the existing system.

[2]This review aims at applying reinforcement learning (RL) in critical care decision-making. It groups RL applications, underlines treatment personalisation, and lists some issues, like model validation and practical application.

[3]Formulating for early detection of autoimmune disease through machine learning algorithms New algorithms are being employed to improve the diagnosis of Sjögren's Syndrome where clinical information and imaging studies are used in the process. Using deep learning, doctors make more accurate and faster diagnoses than using conventional approaches. These enhancements are changing the diagnostic paradigms, delivering faster accuracy impacts.

[4]The images of salivary gland tissues are also being processed using CNNs to classify the area as healthy or affected. This pattern recognition also assist in early identification, which is important for treatment. Each automated image analysis has time saving benefits and minimized dependence on manual skills.

[5]The forecast of autoimmune diseases is done through appears of machine learning approaches inclusive of random forests and SVMs. The clinical data coming from patients aid in treating patients and enhancing their experience in the hospitals. These predictive systems make early intervention achievable.

[6]Machine learning in diagnosis often has flawed results due to lacking data and significant data disparity in rare diseases such as Sjögren's Syndrome. Moreover, there is a critical issue associated with the explanation of models that makes it challenging to integrate them clinically. Thus, it is crucial to overcome them as a prerequisite to sound application of AI in medical diagnosis.

[7]In this work, a MAFRL framework for allocating resources in IoMT network assisted by UAV is proposed and developed. It is intended to increase the computational offloading effectiveness and reduce latency and energy utilization simultaneously through the application of Markov decision process festooned with the evaluation of interference and dependency.

[8]In this paper, recent progress in managing multiple cooperative cooperative multi-agent systems, especially by utilizing DRL to serve the healthcare-related problem of tasks is covered. The study Classifies different types of DRL approaches, and its application discusses its applicability in real world, cooperation of agents system wide for enhancing the patient care outcomes.

[9]This survey evaluates the current level of integrating IoT and AI toward building remote healthcare monitoring systems with focus on the importance of these technologies towards improving health care in smart cities. It describes several paradigms and strategies that involve IoT sensors and big data and analytical intelligence for patient status management and health care organization productivity enhancement.

[10]Focusing on smart healthcare systems the paper is devoted to the analysis of AI and IoT integration and their further development and possible difficulties in implementation. It shifts an emphasis on how the IoT improves in connection with AI to help deliver improved patients' care, data control, and prognosis at healthcare sectors while discussing the present development and possible trends.

[11]The literature on the use of AI-based systems for SPM indicates a growing emphasis towards the use of machine learning models for the development of the predictive analytics and improving processes for project delivery.

[12]Compiler design research also focuses on the timely implementation of parsing approaches, such as Earley Parsing, for transforming natural language pseudocode into a well-organized form for execution.

[13]Work done on the topic of image segmentation in medical diagnosis particularly in the case of GI tract analysis indicates the application of deep learning models like U-Net and Transformer based models for better and efficient diagnosis with improved and increased scalability..

[14]Artificial intelligence solutions applied to SPM use machine learning techniques to allocate resources more efficiently and to foresee project performances. Early Parsing and other being developed methodologies are commonly used techniques in designing a compiler for dealing with advanced syntactical structures and for producing machine readable code.

Overall the effectiveness of RPM for CD in context of wearable sensors, cloud computing, and AI while also discussing the challenges of data privacy and data management for integration. It also revisits the RL approaches applied in decision support in the intensive care units and cares to discuss existing limitations in external and everyday usage of models. Deep learning and CNNs are improving diagnosis and prognosis of autoimmune diseases such as Sjögren's Syndrome. Several healthcare benefits have been achieved through combining AI with IoT; contemporary work on Multi-agent systems and Reinforcement learning is also enhancing he care systems. However, some great issues as dissimilar data and model interpretability still are the main hurdles for clinical AI implementation.

III. DRL Formulation

1.1 Markov Decision Process (MDP) Formulation :
In Multi-Agent Patient Health Environment, every agent follows a specific role and interacts with the environment, the roles include normalizing metrics, detecting anomalies and

recommending treatment there is a certain order to the interaction (e.g., follow-up monitoring). The problem may be modeled through MDP where:

State Space (S): The state of the system is defined through many health metrics, as for example HeartRate, BloodPressure, GlucoseLevel etc. This space is more or less a vector of 12 dimensional real numbers that correspond to certain observations.

$$S = [\text{HeartRate}, \text{BloodPressure}, \text{GlucoseLevel}, \ldots]$$

Action Space (A): The action that an agent performs at each step of the process is controlled, therefore discrete in nature. There are a set of actions each agent is assigned (metric normalisation, anomaly detection etc).

Transition Function (T): The transition function outlines the way an action taken by an agent modifies the state in question. This movement is fluid and very much dependent on the agent, for instance, his/her action of changing certain health metrics such as HeartRate or BloodPressure.

Reward Function (R): The reward function made it possible to highlight what factors need to be acted upon in particular so as to achieve the action's goal. In this particular case, the action can be considered successful or failing when the goal set is aimed at improving the health of the patient.

$$R(s, a, s') = \text{Reward based on action's improvement}$$

Discount Factor (γ): It is basically a value that enables to define how much, as an agent, preference is given towards immediate rewards versus those further in time. Values used are usually between 0 to 1 (e.g. 0.99).

Policy(π) : The policy is the strategy that the agent uses to determine its next action based on the current state. It can be deterministic or stochastic.

$$\pi : S \to A$$

1.2 Objective

The objective for each agent is to maximize the cumulative discounted rewards over time:

$$E\left[\sum_{t=0}^{T} \gamma^t R_t\right]$$

Where R_t is the reward at time step t, and γ is the discount factor.

IV. System Architecture

1. Neural Network Architecture:

Every agent has a model in the form of a fully connected neural network which takes health metrics to derive Q-values and actions. The neural network architecture for the DQN agent consists of:

A. Input Layer:

Receives the state vector S_t which encompasses one or more aspects of health such as pulse rate or blood pressure.

B. Three Hidden Layers:

There are four nodes in each layer to involve non-linear transformation engaging the patient data. The leveled rectified linear unit activation function is used to add non-linearity into the model.

C.Output Layer:

Provides the policy which is the probability distribution over actionable states, and Q-values for them. In the telehealth system, actions include changing the dosage of a drug, triggering an alarm, or making an entry of patient records.

The network is trained using backpropagation, where the Q-value updates are performed based on the following equation:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_t + \gamma \max_{a'} Q(S_{t+1}, a') - Q(S_t, A_t)\right]$$

2. Environment, State, and Goal:

A. Environment:

A patient environment contains patient health data obtained either from wearable devices or through direct input via a smartphone. It is also continually updated by the agent's actions.

B. State Space:

The state space S_t also encompasses numerous patient characteristics which show the current state of health, for example, heart rate and oxygen saturation.

C. Action Space:

Each of them has a finite set of actions that it can take with the environment (request an alert, change magnitude of medication).

D. Goal:

The purpose of the system in this case is to serve as a checks and balance on the patient's health status, coming close to wherever that will require the input of another procedure or medication.

The described system combines four consecutive DQN agents that operate in the environment, which in this case is the patient health data, and are gaining rewards as feedback on their actions for taking the best possible actions. Each agent focuses on a specific task to ensure continuous and adaptive care:

3. System Architecture Overview:

The described system combines four consecutive DQN agents that operate in the environment, which in this case is the patient health data, and are gaining rewards as feedback on their actions for taking the best possible actions. Each agent

focuses on a specific task to ensure continuous and adaptive care.

Agent 1: Health Data Monitoring

Task: Ensure patient health data are bounded by certain fixed scales.
Action Space: Offers no performance of specific agents but offers a normalized dataset for other agents or processes.
Output: Normalized health data.

Agent 2: Anomaly Detection and Alerts

Task: Flag up abnormalities in patient activity and report the same.
Action Space: left=0pt

- Raise Alert
- No Action

Output: System activation and noted abnormalities.

Agent 3: Treatment Recommendation

Task: Suggest that courses of action or treatment be taken when certain abnormalities are found.
Action Space: left=0pt

- Adjust Medication
- Schedule Follow-Up
- Recommend Lifestyle Changes
- Activate Emergency Protocols

Output: Suggested treatment plans.

Agent 4: Follow-Up Monitoring

Backend:
Built with Django as a framework to manage model and data operations.

Task:
For treatment progress, support and assess the status of a patient after availing the treatment service, and create reports.

Action Space:

- Follow up appointment
- Send Data to Cloud
- Log Patient Data
- Generate Report

Output:

Follow-up actions and logs.

4. Reward Function:
The reward function promotes behaviors that produce a better outcome for patients, therefore enhancing patient satisfaction. Rewards are assigned as follows:

- +5: Better treatment recommendations with the health of the patient improving as a result.
- +2: Accurate anomaly detection.
- +1: Specifically, the results indicate the successful logging of follow-up actions.
- -1: If recommendations are wrong or false alarms are given.

AGENTS USED :

A. DQN Agent

The DQN is derived from Q-learning, which utilizes a deep learning network to estimate the Q-function, which zones the relationships between state actions and the expected reward. This is particularly useful in an environment where the system is characterized by large or continuous state space.

Neural Network Architecture: Specifically, a feed-forward neural network model for input 'state' which is equivalent to the health metrics. The output, in this case, is the Q-value assignable to each of the possible actions that can be performed. This network is trained to minimize the Bellman error as seen from the above derivation of the credit assignment path.

$$Q^*(s,a) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t R_t\right]$$

Experience Replay Buffer:
Stores experiences as tuples (s, a, r, s') in a buffer and samples batches of experiences to update the Q-network. Sample mini-batch from the replay buffer and compute the target:

$$y = r + \gamma \max_{a'} Q'(s', a')$$

Target Network: A replication of the Q-network with parameters that is updated at certain time intervals. This helps prevent instability during training.

$$L = \mathbb{E}\left[(Q(s,a) - y)^2\right]$$

Update the Q-network by minimizing the loss between the predicted Q-values and the target by above formula.

A. PPO Agent

PPO is an on-policy algorithm, which makes an optimal balance of how much an agent wants to explore and how much he wants to exploit. It employs a clipped objective function for this reason so that the policy does not switch too much. Collect trajectories using the current policy. Compute advantages for each trajectory using Generalized Advantage Estimation (GAE). Update policy using the clipped objective. Repeat the process for a fixed number of episodes.

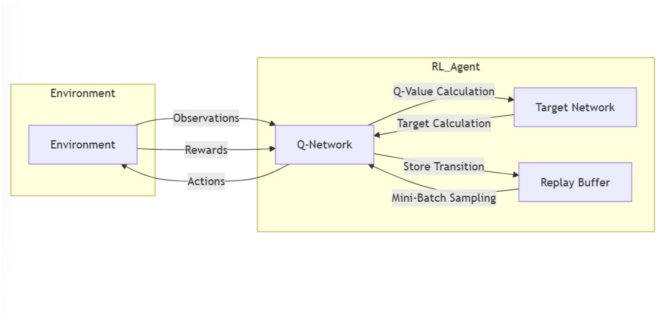

Fig. 1. DQN Architecture

PPO Algorithm : The objective function for PPO is given by:

$$L_{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(\rho_t(\theta)\hat{A}_t,\ \text{clip}\left(\rho_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right]$$

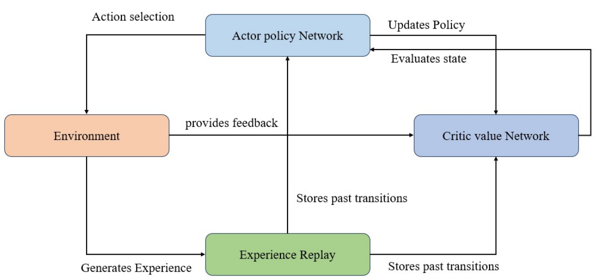

Fig. 2. PPO Architecture

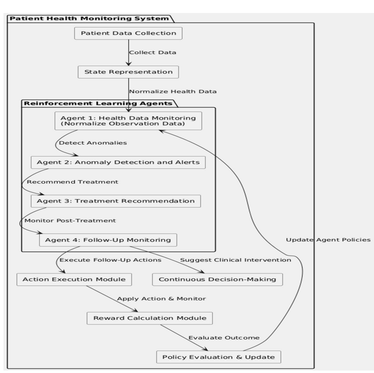

Fig. 3. Global Architecture

V. Implementation

This Project, represented through the 'MultiAgentPatientHealthEnv', including normalizing and predicting metrics, detecting anomalies, recommending treatments, and follow-ups. The state space can be represented by different health observations such as heart rate, blood pressure and glucose levels and the action space is discrete in which each agent has three actions to perform. The function of reward is to bring the agent to improve the condition of the patient and to punish a degenerating state of health hence guiding the agents to healthier conditions.

While the DQN constructs a Q-value function using a neural network, the PPO learns a policy function that picks actions.Improvement performance assessment. These agents were implemented using MATLAB Reinforcement Learning Toolbox that provided a strong supporting structure for implementing and training the agents in a simulated healthcare environment.

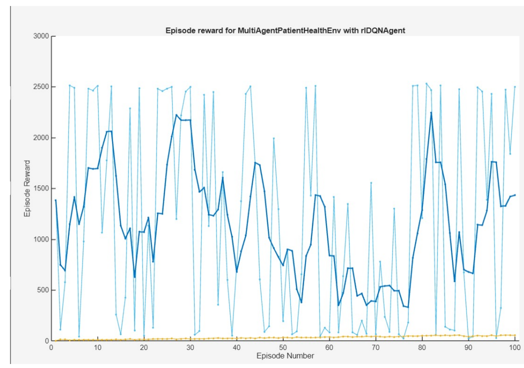

Fig. 4. DQN implementation

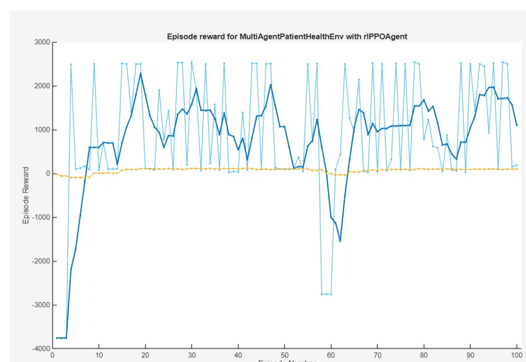

Fig. 5. PPO Implementation

VI. Results and Analysis

VII. Conclusion

The Multi-Agent Patient Health Environment provides a useful framework for reinforcing healthcare scenarios through multi-agent simulations. It encompasses the intricacies of health management in the real world, where various aspects including medication, lifestyle, and monitoring are all present by embedding multiple agents with different assignments. The way in which the environment is structured allows for the agents to develop improvement strategies for health outcomes

Agent	Algorithm	Regret	ComputationalComplexity	EmpiricalPerformance
Normalize Health Metrics	DQN	2.937065017	0.000088	5.535413565
Detect Anamolies	DQN	2.600769229	0.0000615	6.755278387
Treatment Recommendation	DQN	2.819462488	0.0000465	4.136089199
Follow-up	DQN	3.888991515	0.0000488	5.490915947
Normalize Health Metrics	DQN	2.656709744	0.0000762	5.158938025
Detect Anamolies	DQN	3.371612746	0.00001	5.381998678
Treatment Recommendation	DQN	2.898565094	0.0000088	3.543105788
Follow-up	DQN	2.772704033	0.0000098	6.086174693
Normalize Health Metrics	DQN	3.688720069	0.0000083	5.387612751
Detect Anamolies	DQN	3.694942354	0.0000129	5.138968605
Treatment Recommendation	DQN	2.684626239	0.0000106	6.293302798
Follow-up	DQN	2.123244886	0.000008	4.302731987
Normalize Health Metrics	PPO	2.878735767	0.0000071	6.185986625
Detect Anamolies	PPO	3.206266176	0.0000083	5.098364767
Treatment Recommendation	PPO	2.698322577	0.0000081	3.770821962
Follow-up	PPO	2.528670487	0.0000092	5.85502645
Normalize Health Metrics	PPO	3.866752905	0.000008	6.83186749
Detect Anamolies	PPO	4.227428131	0.0000105	5.995075706
Treatment Recommendation	PPO	4.179913232	0.0000091	6.213196313
Follow-up	PPO	3.461467375	0.0000092	6.478761543
Normalize Health Metrics	PPO	2.828714727	0.0000088	5.071349769
Detect Anamolies	PPO	3.945993142	0.0000092	6.724047564
Treatment Recommendation	PPO	3.033070441	0.0000093	4.827138388
Follow-up	PPO	3.033262961	0.000009	4.929411074

Fig. 6. Hyperparameter tuning

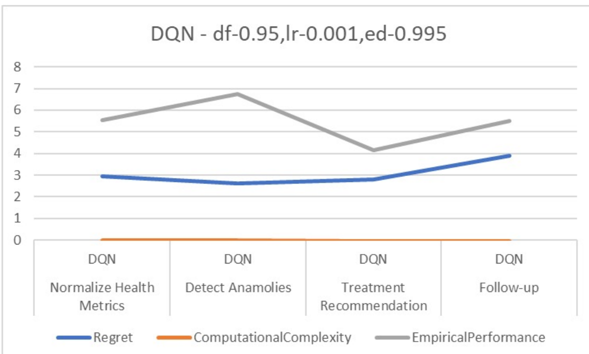

Fig. 7. Hyper parameter tuning DQN 1

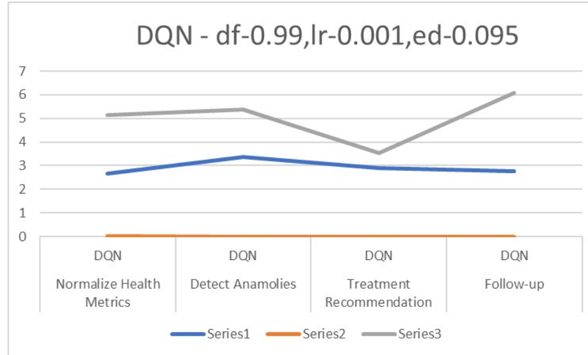

Fig. 8. Hyper parameter tuning DQN 2

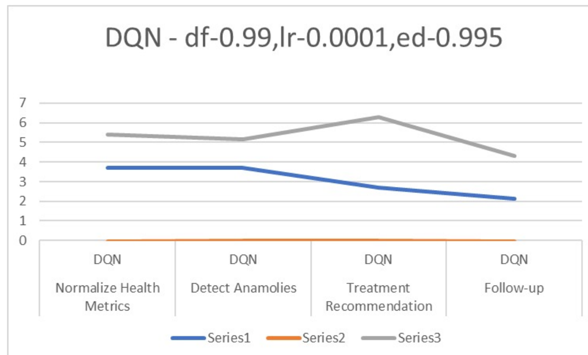

Fig. 9. Hyper parameter tuning DQN 3

via repetitive and implicit exposure to patient indicators. The classification system for health conditions guarantees that the agents are given feedback on their actions, which is, in essence, desired, towards improving patient health which is the target. This method allows us the creation of intelligent real time changes in the recommendations of the healthcare system such that they are at all times, personalized and based on the patient's information. With further development, such environments can be used to train healthcare practitioners and AI models and improve medical decision making and patient care.

References

[1] El-Rashidy N, El-Sappagh S, Islam SMR, M El-Bakry H, Abdelrazek S. Mobile Health in Remote Patient Monitoring for Chronic Diseases: Principles, Trends, and Challenges. Diagnostics (Basel). 2021 Mar 29;11(4):607. doi: 10.3390/diagnostics11040607

[2] Liu S, See KC, Ngiam KY, Celi LA, Sun X, Feng Reinforcement Learning for Clinical Decision Support in Critical Care: Comprehensive Review J Med Internet Res 2020;22(7):e18477

[3] B. Eden, I. Bistritz, N. Bambos, I. Ben-Gal and E. Khmelnitsky, ”Personalized and Energy-Efficient Health Monitoring: A Reinforcement Learning Approach,” in IEEE Control Systems Letters, vol. 7, pp. 955-960, 2023, doi: 10.1109/LCSYS.2022.3229074.

[4] Oroojlooy, A., Hajinezhad, D. A review of cooperative multi-agent deep reinforcement learning. Appl Intell 53, 13677–13722 (2023). https://doi.org/10.1007/s10489-022-04105-y

[5] Mazin Alshamrani,IoT and artificial intelligence implementations for remote healthcare monitoring systems: A survey,Journal of King Saud University - Computer and Information Sciences,Volume 34, Issue 8, Part A,2022,Pages 4687-4701,ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2021.06.005.

[6] Wu, Shangrui, et al. ”DescrA. M. Seid, A. Erbad, H. N. Abishu, A. Albaseer, M. Abdallah and M. Guizani, ”Multiagent Federated Reinforcement Learning for Resource Allocation in UAV-Enabled Internet of Medical Things Networks,” in IEEE Internet of Things Journal, vol. 10, no. 22, pp. 19695-19711, 15 Nov.15, 2023, doi: 10.1109/JIOT.2023.3283353.

[7] S. M. J. Jalali et al., ”New Hybrid Deep Neural Architectural Search-Based Ensemble Reinforcement Learning Strategy for Wind Power Forecasting,” in IEEE Transactions on Industry Applications, vol. 58, no. 1, pp. 15-27, Jan.-Feb. 2022, doi: 10.1109/TIA.2021.3126272.

[8] K. K. Nguyen, T. Q. Duong, N. A. Vien, N. -A. Le-Khac and M. -N. Nguyen, ”Non-Cooperative Energy Efficient Power Allocation Game in D2D Communication: A Multi-Agent Deep Reinforcement Learning Approach,” in IEEE Access, vol. 7, pp. 100480-100490, 2019, doi: 10.1109/ACCESS.2019.2930115

[9] Varghese, Nelson Vithayathil and Qusay H. Mahmoud. “A Hybrid Multi-Task Learning Approach for Optimizing Deep Reinforcement Learning Agents.” IEEE Access 9 (2021): 44681-44703. .

[10] Ian Fox, Joyce Lee, Rodica Pop-Busui, Jenna Wiens,Deep Reinforcement Learning for Closed-Loop Blood Glucose Control https://doi.org/10.48550/arXiv.2009.09051

[11] A. A. Abdellatif, N. Mhaisen, A. Mohamed, A. Erbad and M. Guizani, ”Reinforcement Learning for Intelligent Healthcare Systems: A Review of Challenges, Applications, and Open Research Issues,” in IEEE Internet of Things Journal, vol. 10, no. 24, pp. 21982-22007, 15 Dec.15, 2023, doi: 10.1109/JIOT.2023.3288050

[12] An IoT enabled smart healthcare system using deep reinforcement learning Duraiswamy Jothinath Jagannath, Raveena Judie Dolly, Gunamony Shine Let, James Dinesh PeterAn IoT enabled smart healthcare system using deep reinforcement learning

[13] Naeem, Muddasar & Paragliola, Giovanni & Coronato, Antonio. (2020). A reinforcement learning and deep learning based intelligent system for

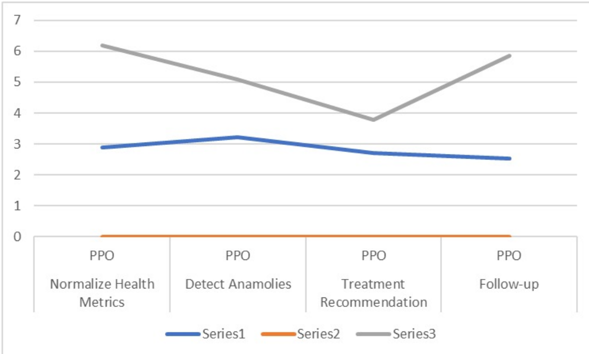

Fig. 10. Hyper parameter tuning PPO 1

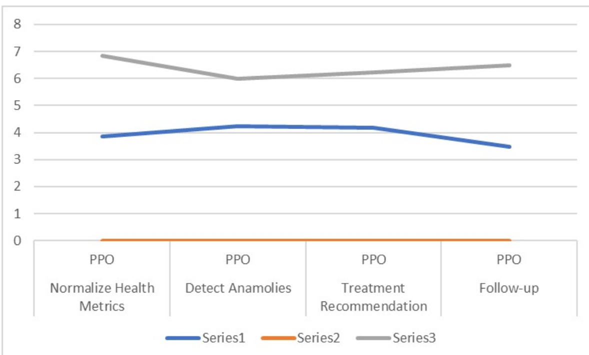

Fig. 11. Hyper parameter tuning PPO 2

the support of impaired patients in home treatment. Expert Systems with Applications. 168. 114285. 10.1016/j.eswa.2020.114285.

[14] Ziqiaing Ye, Yulan Gao, Yue Xiao, Zehui Xiong, Dusit Niyato,Deep Reinforcement Learning Empowered Activity-Aware Dynamic Health Monitoring Systems, https://doi.org/10.48550/arXiv.2401.10794

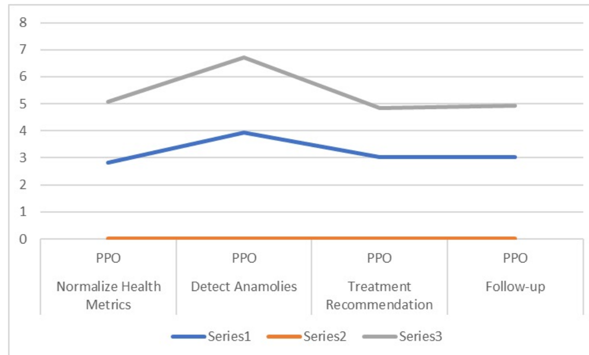

Fig. 12. Hyper parameter tuning PPO 3

Combination ID	Learning Rate	Clip Range	Batch Size	Num Epochs	Discount Factor	GAE Lambda
1	0.001	0.2	64	3	0.95	0.95
2	0.0005	0.3	128	5	0.97	0.95
3	0.0001	0.4	256	10	0.99	0.98

Fig. 13. PPO Hyper parameter tuning values