Journal Pre-proof

A reinforcement federated learning based strategy for urinary disease dataset processing

Saleem Ahmed, Tor-Morten Groenli, Abdullah Lakhan, Yi Chen,
Guoxi Liang

PII: S0010-4825(23)00675-3
DOI: https://doi.org/10.1016/j.compbiomed.2023.107210
Reference: CBM 107210

To appear in: *Computers in Biology and Medicine*

Received date : 19 December 2022
Revised date : 8 June 2023
Accepted date : 25 June 2023

Please cite this article as: S. Ahmed, T.-M. Groenli, A. Lakhan et al., A reinforcement federated learning based strategy for urinary disease dataset processing, *Computers in Biology and Medicine* (2023), doi: https://doi.org/10.1016/j.compbiomed.2023.107210.

A Reinforcement Federated Learning based Strategy for Urinary Disease Dataset Processing

Saleem Ahmed[a], Tor-Morten Groenli[b], Abdullah Lakhan[b,c], Yi Chen[d,*], Guoxi Liang[e,*]

[a]*Department of Computer System Engineering, Dawood University of Engineering and Technology, Sindh, Karachi, Pakistan (Email: saleem.ahmed@duet.edu.pk)*
[b]*Mobile Technology Laboratory, School of Economics, Innovation and Technology, Kristiania University College, 0153 Oslo, (Email: abdullah.lakhan@kristiania.no, Tor-Morten.Gronli@kristiania.no)*
[c]*Department of Computer Science, Dawood University of Engineering and Technology, abdullah (Email: abdullah.lakhan@duet.edu.pk)*
[d]*Key Laboratory of Intelligent Informatics for Safety & Emergency of Zhejiang Province, Wenzhou University, Wenzhou 325035, China (Email: kenyoncy2016@gmail.com)*
[e]*Department of Artificial Intelligence, Wenzhou Polytechnic, Wenzhou 325035, China (Email: guoxiliang2017@gmail.com)*

Abstract

Urinary disease is a complex healthcare issue that continues to grow in prevalence. Urine tests have proven valuable in identifying conditions such as kidney disease, urinary tract infections, and lower abdominal pain. While machine learning has made significant strides in automating urinary tract infection detection, the accuracy of existing methods is hindered by concerns surrounding data privacy and the time-intensive nature of training and testing with large datasets. Our proposed method aims to address these limitations and achieve highly accurate urinary tract infection detection across various healthcare laboratories, while simultaneously minimizing data security risks and processing delays. To tackle this challenge, we approach the problem as a combinatorial optimization task. We optimize the accuracy objective as a concave function and minimize computation delay as a convex function. Our work introduces a framework enabled by federated learning and reinforcement learning strategy (FLRLS), leveraging lab urine data. FLRLS employs deterministic agents to optimize the exploration and exploitation of urinary data, while the actual determination of urinary tract infections is performed at a centralized, aggregated node. Experimental results demonstrate that our proposed method improves accuracy by 5% and reduces total delay. By combining federated learning, reinforcement learning, and a combinatorial optimization approach, we achieve both high accuracy and minimal delay in urinary tract infection detection.

Keywords: Reinforcement Learning, Urinary Disease, Federated Learning, Processing Delay.

*Corresponding author

1. Introduction

Urinary tract infections, a type of urinary disease, are serious medical conditions that can indicate underlying healthcare issues. These infections can be detected through various laboratory tests [1]. Common symptoms and characteristics of urinary tract infections resemble those observed in hospital laboratory tests, such as frequent urination, painful or burning sensations during urination, lower back pain, blood in the urine, and cloudy or strong-smelling urine [2]. The high prevalence of urinary tract infections results in substantial patient data that requires analysis. Consequently, digital healthcare systems incorporating artificial intelligence (AI) for predicting urinary diseases have gained considerable popularity in the healthcare industry [3]. AI algorithms, including machine learning, artificial neural networks, and deep learning, have been widely employed in digital healthcare for various purposes, including predicting cancer, lung disease, and urinary tract infections [4–7]. As a result, AI-based digital healthcare has demonstrated significant potential in predicting urinary diseases and has become a valuable tool in the healthcare industry [8]. Currently, two primary areas of investigation in healthcare applications are scheduling healthcare data across different computing nodes and predicting healthcare data using machine learning techniques [9].

Previous research has explored the prediction of urinary tract infections using reinforcement learning techniques such as SARSA (state-action-reward-sate-action), Q-learning, and deterministic policy [10, 11]. However, these studies have focused solely on a single data type when addressing scheduling issues. To address the challenge of handling multiple data types, machine learning classification algorithms such as artificial neural networks (ANNs) [12] and convolutional neural networks (CNNs) [13–16] have been proposed for simultaneous processing of diverse datasets [17, 18]. However, this approach can result in increased processing delays as the datasets must be split, trained, and tested on individual nodes during preprocessing. In recent years, federated machine learning has emerged as a potential solution. This preprocessing technique involves training and testing models on different nodes and sharing aggregated models with a central node for decision-making [19–23]. Federated learning shows promise in scenarios where applications require processing diverse models, such as predicting urinary diseases, urinary infections, and lower back urine issues, each with unique attributes and datasets that cannot be efficiently processed by a single machine learning-enabled node in healthcare applications. In such cases, healthcare models are trained and tested on different nodes using federated learning, and the aggregated model is utilized for further processing. However, existing healthcare networks that incorporate reinforcement learning and federated learning face challenges, including increased delays and the need for improved accuracy when handling urinary workloads.

The primary objective of this paper is to propose a comprehensive strategy that combines reinforcement learning and federated learning to achieve more accurate training of urinary diseases while minimizing processing delays during execution. The study considers various laboratories as local agents, each comprising states and actions without decision-making processing. All agents train their environments using deep Q-learning. The infrastructure is based on the federated reinforcement learning paradigm, where local states share trained data and collaborate to reach a final decision with cumulative rewards. This work brings innovative contributions to the existing state-of-the-art studies, and the following are the key highlights of this research.

- The study introduces the FLRLS method for urinary disease tasks. Existing federated learning methods only train weights and do not consider delay factors during the training process in the system [9, 10, 17, 19, 20, 22, 24, 25]. FLRLS is more efficient regarding the delay and considers training and processing delays for urinary applications.
- This paper describes delay-optimal preprocessing training and testing schemes for urinary datasets, such as urinary bladder inflammation and renal pelvis nephritis, on different computing nodes. Existing training and testing schemes of federated learning on fog cloud incurred long delays during processing [21, 22].
- The study also presents a urinary disease simulator demonstrating the practical implementation of the considered problem. However, existing federated learning schemes have not been adequately deployed in different urinary task simulators [21, 22].

The remaining sections of the paper are organized as follows. Section 2 provides an elaboration on the related work conducted in this field. Section 3 explains the proposed description and formalizes the problem under study, while Section 4 presents a heuristic approach proposed for the considered problem. Section 5 presents the results of the experimental study, and Section 6 concludes the paper by offering insights into potential areas for future research.

2. Related Work

In this section, we discuss digital healthcare approaches for urinary diseases. These digital systems are designed based on fog cloud infrastructure. Fog and cloud servers are geographically distributed and offer processing services from different locations.

Digital healthcare is a paradigm that utilizes digital and information communication technologies to enhance the quality and efficiency of healthcare. It is crucial to manage the urinary detection workload and predict underlying diseases and their causes based on available data. Various AI and machine learning-based schemes and systems have been proposed for digital healthcare [1–3]. These studies primarily focus on training urinary datasets and workloads using machine learning models. They extract different features from urinary data to predict diseases. However, the approaches discussed in these studies face several challenges, including homogeneous resource and data utilization, increased processing delays, missed deadlines, and high resource consumption. These issues remain key challenges in previous research.

Supervised learning methods, including Random Forest and XGBoost, have been employed to classify disease types within the ureters using urinary data reports [7, 9, 10]. These studies have focused on data prediction techniques and have shown improved efficiency in training and testing time compared to previous research. However, the centralized data server still experiences resource and time consumption challenges when scheduling urinary data during execution.

To reduce training, testing, and centralized processing time, a urinary system based on federated learning is implemented [11, 17–19]. This approach aims to gather data from various local models across different locations and analyze it on a single node for disease prediction and classification. Additionally, deep reinforcement learning techniques enable computational methods and deep learning models for processing urinary data using given features. These studies propose aggregated policies based on deep and federated learning to handle urinary data on fog and cloud nodes. While these studies enhance computation, training, and testing time, they still need to address important issues such as resource consumption, meeting deadlines, and data sharing within their models.

Federated learning-enabled schemes based on fog and cloud networks have been presented for resource balancing and consumption for urinary processing [20, 21]. Each federated learning-enabled node trains and validates the workloads of urinary datasets on the edge layer with the minimum resource consumption. After the edge processing, all datasets are aggregated to the centralized nodes. All the local weights of the federated nodes offload the locally trained models to the aggregated node. However, local iteration and global training iteration in federated learning models consume much more resources and must meet many urinary task deadlines.

To efficiently manage local and global iterations of dataset training, testing, and sharing, researchers have developed resource-efficient methods [22, 23, 26–31]. These studies propose a decision support system that combines reinforcement learning and federated learning for urinary workloads. The strategy aims to train, split, and test data on different nodes and then aggregate the results based on different states and a finite set of local and global iterations. Simultaneously, the urinary systems accept real-time data and train and test it on different nodes in various hospitals. The processed data is offloaded to centralized hospitals for further network analysis. During the processing of workloads in the system, parameters such as reward, discount factors, policy, and metrics are optimized using federated learning. However, these studies primarily focus on data training, prediction, and classification on the system side. The data is trained and tested with new data using finite iterations of local and global models. While they succeed in minimizing resource consumption compared to existing studies, computational time, overhead training, and testing time concerning deadlines need to be adequately balanced in fog cloud networks.

Federated learning resource scheduling techniques have been utilized for the task allocations of healthcare applications [24, 25, 32–39]. However, the deadlines for urinary processing tasks still need to be included. Another issue is that data privacy among federated learning models exists in these studies. These studies [40–49] optimized urinary disease prediction regarding bladder cancer issues through Endoscopic images. The different meta-heuristics suggested improving the local and global searching of computing nodes to find the optimal results of given urinary datasets. However, they have considered the fixed parameters for urinary datasets such as training, testing, and accuracy and incurred longer delays. These studies [50–54] suggested the local and global searching optimal schemes to minimize some delays for urinary datasets in the distributed computing nodes. However, these studies are still missing deadlines or urinary tasks due to resource issues in their computing nodes in distributed networks.

Although these studies [9, 10, 17, 19, 20, 22, 24, 25] are related or closely related to our work, to the best of our knowledge, the execution of urinary tasks in federated reinforcement learning based on deadline and computation balancing has not yet been proposed. This study proposes aggregated FLRLS strategies considering deadline, processing delay constraints, and resource consumption. With the proposed strategies, fog cloud networks can optimize deadlines and processing time (e.g., delay) for different urinary tasks.

3. Federated Learning and Reinforcement Learning Architecture

The proposed federated learning and reinforcement learning strategy Architecture is shown in Figure 1, combining different components such as datasets and tasks, local laboratories, and aggregated hospital nodes. The dataset D has the following tasks. For instance, i1=Temperature, i2=nausea, i3=Lumbar i4=Urine Pushing, PEE, i5=Micturition Pains; i6=Burning of urethra, and i7= urinary bladder. We divided the process of execution based on federated learning, where differ-

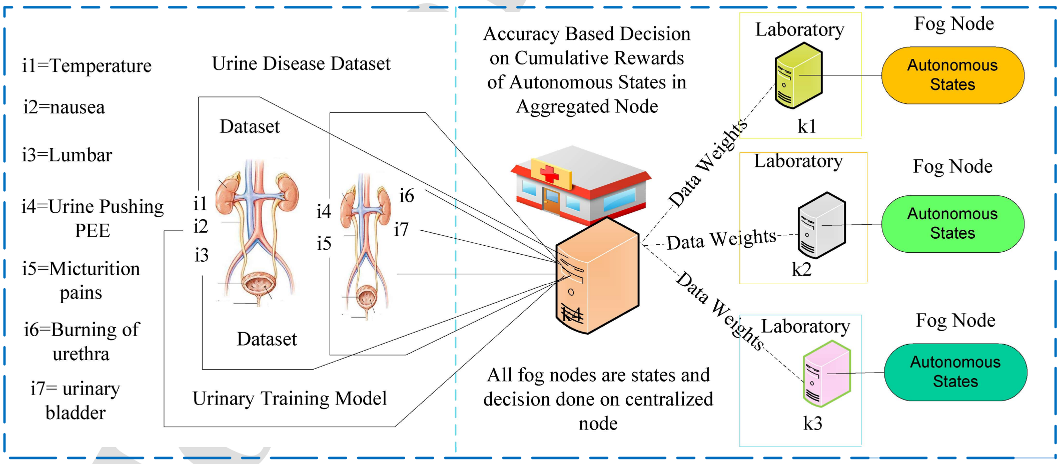


Figure 1: Urinary Disease Processing Expert System Based on Reinforcement Federated Learning.

ent local laboratories train and test their models locally and share them with the aggregated centralized hospital for further analysis. The study considers the I number of urinary dataset tasks, e.g., $\{i = 1, \ldots, I\}$. Each task has different features: $i = \{w_i, i_d, status_i\}$. Whereas w_i is the workload determined in megabytes (MB), i_d

Table 1: Math Notations of Problem

Notation	Value
S	Number of problem episodic states
s	Particular episodic state
A	Number of performed actions
a	Particular defined action
T	Transition probability in states
t	Probability transition
R	Set of rewards
t	Particular reward in nodes
α	Learning rate of agent
γ	Discount factor
K	Number of local laboratories and aggregated nodes
ζ_k	Particular node computing capability
ϵ_k	Computing resource of l node
D	Urinary dataset
I	Set of tasks of urinary dataset D
$i = \{w_i, i_d, status_i\}$	Workload, deadline and status of task
$\lambda^* = \sum_{t\geq 0} r^t \gamma^t$	Optimal policy
$Q^\lambda(S, A, \phi)$	Optimal DQN values

is the workload deadline, $status_i$ is the execution status of the workload. The urinary dataset component is taken as the input from different laboratories as the workloads. We divided the urinary dataset among three different healthcare laboratories. All laboratories are autonomous and cannot share data for business or security reasons. Each laboratory must keep the full information of patients private from the centralized hospital for further analysis. The main reason is that each laboratory trains and tests the dataset model locally and offloads it to the aggregate node for cumulative decision. We assumed all the nodes (e.g., laboratories) and centralized hospitals are healthcare nodes, e.g., $\{k = 1, \ldots, K\}$. We divided the urinary dataset among different laboratories. Therefore, we train and test the randomly divided dataset based on deep q-learning. So, part of the problem is described as Markov decision model-based processing, where the model is trained and tested in different states [55]. Each laboratory executed the random dataset locally and offloaded their gradients of tasks to the aggregated node. We defined all notations of the problem in Table 1. The local processing time on the local laboratories determined equation (1).

$$\phi = \sum_{i=1}^{I} \sum_{k=1}^{K} \frac{w_i}{\zeta_k}, \quad \forall i = 1, \ldots, I, \ w_i \in D. \tag{1}$$

Equation (1) determines the processing time at local laboratories during their training and test models before being shared to the aggregated node for processing. The communication between aggregate nodes and different nodes during uploading and downloading models is determined in equation (2).

$$Com = \sum_{i=1}^{I} \sum_{k=1}^{K} \frac{\phi_k}{bandwidth_{up} + bandwidth_{down}}, i = 1, \ldots, I. \tag{2}$$

The aggregate node taking time for the final decision is determined in equation (3).

$$Remote = \sum_{i=1}^{I} \sum_{k=1}^{K} \frac{\phi_k}{\zeta_k}, \quad \forall k = 1, \ldots, K. \tag{3}$$

The total time for all tasks is determined in equation (4).

$$Total = \sum_{i=1}^{I} \sum_{k=1}^{K} \phi + Com + Remote. \tag{4}$$

Each laboratory consisted of different states and sets of actions in different time transitions, as determined in the following equation.

$$\phi^* = \sum_{k=1}^{K} \sum_{t=1}^{T} \sum_{s=1}^{S} \sum_{a=1}^{A} \phi(s, a, t, r, k, w_i) r^t \gamma^t. \tag{5}$$

In equation (5), "." represents the multiplication among numbers, whereas $\phi(s1, a1, t1, r1, k1, w_i)$ are the tuples of ϕ function in terms of states within nodes. The optimal policy in different laboratories is determined in equation (6).

$$Q^{\lambda}(S, A, \phi). \tag{6}$$

We determined the Bellman equation in the following way.

$$Q^{\lambda}(S_t, A_t, \phi) = (1 - \alpha) + Q^{\lambda}(S_t, A_t, \phi) + \alpha(R_t + \lambda^* \max_a(S_t, A_t, \phi)). \tag{7}$$

Equation (7) determines the update-q-table values of the local models in the federated learning model. We determined the accuracy of the model equation (8), where all trained dataset meets the given quality of service requirements. The $logi(\cdot)$ function is a logical operation that equals 1 if the statement is true and 0 if it is false.

$$Accuracy = \sum_{k=1}^{K} \sum_{i=1}^{I} logi(total_i \leq d_i), \quad \forall i = 1, \ldots, I. \tag{8}$$

We determined the loss of the model equation (9) where trained dataset models meet the given quality of service requirements. The $logi(\cdot)$ function is the same as in equation (8).

$$Loss = \sum_{k=1}^{K} \sum_{i=1}^{I} logi(total \leq \epsilon_k), \quad \forall i = 1, \ldots, I. \tag{9}$$

4. Federated Deep Reinforcement Learning Scheme

We are focused on developing federated reinforcement learning (FRL) as a groundbreaking technology with tremendous potential. We utilize the fundamental federated learning deep Q-learning technique to enhance reinforcement learning performance while maintaining data privacy and processing time on different nodes. FRL algorithms can be classified into two groups based on the distribution features of the framework's agents: horizontal federated reinforcement learning and vertical federated reinforcement learning. The former only supports homogeneous nodes, while the latter supports heterogeneous nodes in the same network. We focus on different laboratories and have presented the vertical federated reinforcement learning strategy to solve the urinary dataset combinatorial and concave functions-enabled problem. Our proposed Federated Learning and Reinforcement Learning Strategies (FLRLS) consist of deep Q-learning networks (DQN) and federated learning schemes. Algorithm 2 is the main framework of our study, which processes the input to output via different states in the system. It trains the urinary dataset using a federated learning scheme on different nodes. Initially, mobile agents process urinary tasks locally and then offload to the federated nodes, such as server agendas, for execution.

Algorithm 1: Proposed FLRLS Algorithmic Framework

Input : $\{I, K, I \leftarrow D\}$

Output: $\min Total, \max R$;

begin

  for $i = 1 \sim I \in D$ **do**

    Determined the local training based on Algorithm 1;

    $i \in D \leftarrow k$;

    Determined the local processing time based on equation (1);

    $\phi = \sum_{i=1}^{I} \sum_{k=1}^{K} \frac{i \leftarrow w_i \in D}{\zeta_k}, \quad \forall i = 1, \ldots, I$;

    Determined the local laboratories' communication time based on equation (2);

    $Com = \sum_{i=1}^{I} \sum_{k=1}^{K} \frac{\phi_k}{bandwidth_{up} + bandwidth_{down}}, i = 1, \ldots, I$;

    Determined the aggregated node decision time determined based on equation (3);

    $Remote = \sum_{i=1}^{I} \sum_{k=1}^{K} \frac{\phi_k}{\zeta_k}, \quad \forall k = 1, \ldots, K$;

    Optimize $Total, R$;

  End of Algorithm Scheme;

End of Main;

Algorithm 2 is the main framework of the study in which the system will show how to process the input to output in the system via different states. Algorithm 2 trains the urinary dataset based on a federated learning scheme on different nodes. The urinary tasks are initially processed locally by mobile agents and offloaded to the federated nodes for processing, such as server agendas for execution.

4.1. DQN (Deep Q-Learning Network Enabled Local Scheduling Scheme

Q-learning reinforcement learning strategy that doesn't need a model and learns local laboratory values in local states. Q-learning doesn't have a model in the local laboratory environment to handle the dataset update with randomness and transition to maximize rewards. The q-table is the data structure used to keep track of the

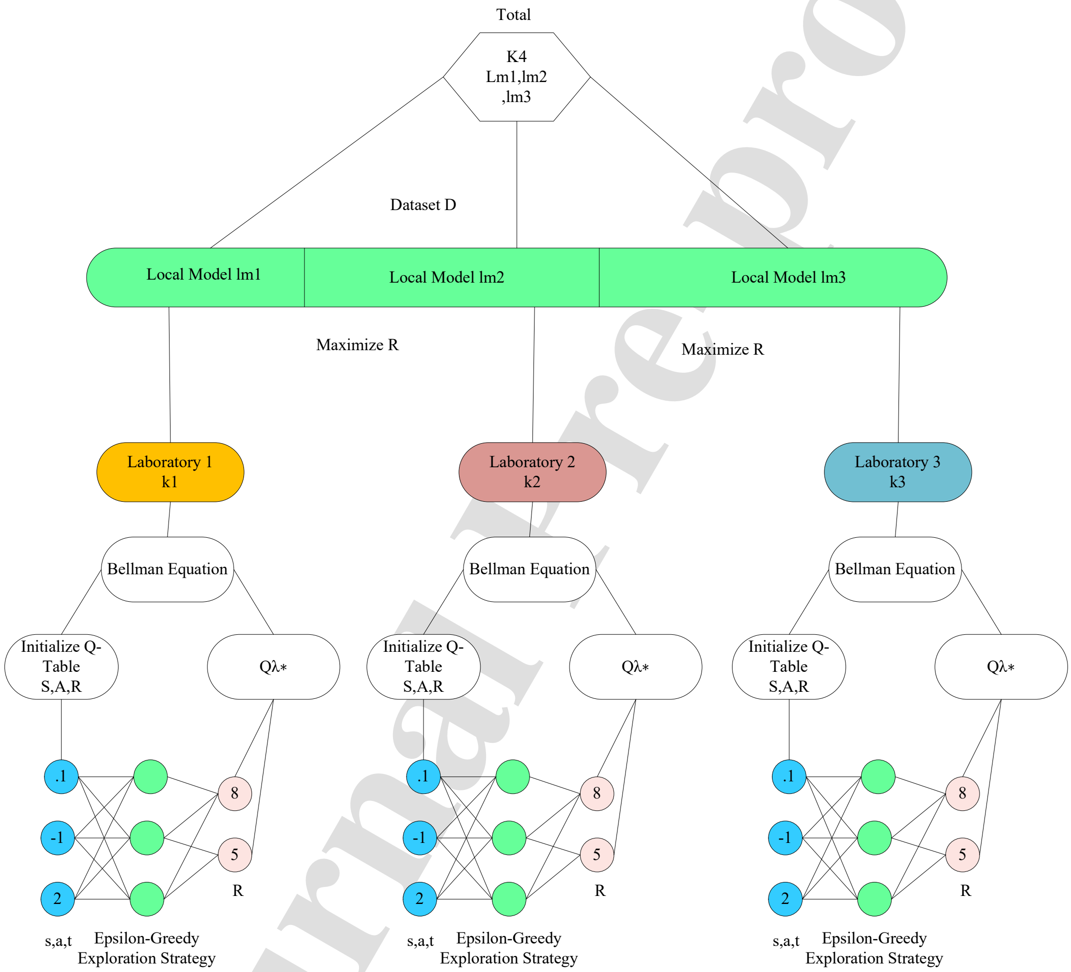


Figure 2: Deep Q-Learning Enabled Federated Learning Laboratory Networks

local states, actions, and expected maximum rewards on a given dataset. The local laboratory agent tries different actions in different states through trial and error, as shown in Figure 2. When it mistakenly tries out various actions at various states, the agent learns the anticipated reward for each state-action pair and updates the Q-table with the current Q-value. Exploration is achieving rewards based on trial and error in the local learning environment. One of the goals of our deep q-learning scheme is to adopt the q-values of different trained dataset gradients in the federated learning environment. A notation γ shows the local laboratory agent's maximum expected rewards from one state to another transition state. The agent is learning based on the shared pairing of states, and choosing the best state is called "exploitation." We exploited a deep neural network to find the optimal states, which is exploitation based on known trial-and-error-based exploration in the environment. Therefore, to influence the trade-off between exploration and exploitation, the study used the Epsilon-Greedy exploration scheme to help agents choose the best states to maximize rewards in local laboratories. We exploited the Bellman equation (7) to update the local q-table with the number of iterations in different state episodes. All the local laboratories updated the local q-table and optimized the patient data with the maximum rewards. We assumed each local laboratory updated the q-table based on 1000 local iterations in different states. The deep neural network maps the input as the states and updates the states pair in the local environment. As shown in Figure 2, the neural network initially takes states as input, such as 1, -1, and 2. We obtained the maximum rewards (R), such as 5 and 8, in the neural networks based on Epsilon-Greedy Exploration Strategy and the Bellman equation. The agent does not need to train after each step, which is important. Instead of training on tiny batches once every state, we employ experienced Replay. This technique significantly sped up our deep Q-Learning implementation, as we discovered. The study used the replay memory off-policy algorithm to store and update the local gradients based on given parameters among previous states. The study exploits the experience replay strategy to learn in small batches to avoid the dataset skewing in neural network-embedded states and actions. In our scenario, we assumed that the healthcare laboratories are part of the federated learning environment. Each state in a local laboratory trained and validated its models locally in different states. All the actions are taken based on optimal policy and offload trained models to the aggregated node for the final decision. Therefore, in our case, all the local agents only trained the models and did not take any decision. Each local laboratory has performed actions in different states and time transitions to improve the overall rewards of the objective function in the minimum processing time. The main advantage is that each local laboratory has trained models of all connected partners. Therefore, resources and processing could be minimized in our proposed system for the urinary dataset.

4.2. Federated Learning Enabled Strategy

The study integrated the federated learning-enabled environment, where different laboratories shared their trained model with the aggregated node for further execution. In our scenario, each node sends and receives an updated trained dataset with gradients from the aggregated node. Therefore, the aggregated node shared the trained models of all laboratories with all connected laboratories. The main reason is that individual laboratories cannot communicate with each other. Therefore, the aggregated node is responsible for sharing trained data among interconnected laboratories. We show the federated learning algorithm, which uses the final results of the commutative dataset gradients to make the final decision. In our scenario, we assumed that the healthcare laboratories trained and validated their models locally in different states. All the actions are taken based on optimal policy and offload trained models to the aggregated node for the final decision. Therefore, in our case,

Algorithm 2: Local Laboratories Deep-Q-Learning Strategy

Input : $Q - Table[s, a, r, k, i_w]$, Replay-Memory[],
Output: Offload Train[]

begin
  Initialize value function Q with random weights;
  Episodes P;
  Mini-Batch Dataset Transition D;
  Layers N number of layers;
  Z Number of neurons;
  $X = 100$ Number of training iteration;
  foreach $(p = 1\ to\ P)$ **do**
    Initialize states sequence at local laboratories L;
    s, a, r, t, k, i_w;
    for $(t = 1\ to\ T)$ **do**
      Select the probability ϵ from random action a_t;
      Otherwise;
      select $\lambda^* = \sum_{t \geq 0} Q(s, \max_a r^t \gamma^t)$;
      Execute: a_t choose optimal policy to maximize rewards;
      $\phi \leftarrow Q - Table[s, a, r, k, i_w, \alpha]$;
      $\max R$;
      $Q^\lambda(S, A, \phi)$;
      Replay-Memory$[s, a, r, k, i_w, D]$;
    foreach $k = 1\ to\ K$ **do**
      if $(Total \leftarrow i \leq d_i \& status_i == 1)$ **then**
        Determine the local training time based on equation (1);
        Train$[s, a, r, k, i_w] \times N, Z$;
        Update $Q - Table[s, a, r, k, i_w]$ based on equation (7);
        $X = X + 1$;
      End of Local Training;
    Offload Train$[s, a, r, k, i_w]$;
  End of Training of Mini-Batch laboratories;
End of the Main;

all the local agents only trained the models and did not take any decision. Only the aggregated node in the federated learning network can decide based on trained data models, as shown in Algorithm 3. There is much diversity in the decisions, where concave and convex objectives are optimized based on updated models in different iterations. The healthcare laboratories trained and validated their models locally in different states with finite training iterations. All the actions are taken based on optimal policy and offload trained models to the aggregated node for the final decision based on initial and newly available data. Therefore, in our case, all the local

Algorithm 3: Federated Aggregated Node Scheme

Input : Update $Q - Table[s, a, r, k, i_w]$;
Output: $\min Total, \max R$;
begin
  Iteration=5000;
  Training Loss 100;
  Accuracy 100;
  foreach *($k = 1$ to K)* **do**
    $Q - Table[s, a, r, k, i_w] \leftarrow k1$ $Q - Table[s, a, r, k, i_w] \leftarrow k1$
    $Q - Table[s, a, r, k, i_w] \leftarrow k1$
    $K = Q - Table[s, a, r, k, i_w] \leftarrow k1, k2, k4$;
    for *(Iteration=1 to 5000)* **do**
      Shared models among all laboratories;
      $K = Q - Table[s, a, r, k, i_w] \leftarrow k1, k2, k4$;
      Determined the $Total$ based on equation (4);
      Determined the $Accuracy$ based on equation (8);
      Determined the $Loss$ based on equation (9);
End of the Main;

agents only trained the models and did not make any decisions for results. Only the aggregated node in the federated learning network can decide based on trained data models, as shown in Algorithm 3. Therefore, only aggregated centralized make final decisions on urinary-trained data for their tasks based on their given deadlines in the fog and cloud networks.

5. Experimental Study

The study designs the simulator based on existing Python libraries such as TensorFlow and PyTorch. We simulate the model based on four computing nodes with the same execution environment, such as the X86 operating system. All the simulation parameters are defined in Table 2. For the user interfaces for data uploading to the laboratories, we designed the patient interface based on JAVA and XML that supports the X86 operating for further data analysis. To ensure data security and privacy, each laboratory shares the trained model based on a given gradient, where the patients' private data will not be shared with the aggregated node for further processing[56, 57].

In the simulation, we integrate the virtual fog servers as the local laboratories based on virtualization. VMware virtualization is often used to share the data among fog and cloud servers for uniform healthcare systems. We integrate the REST API to make the common communication between fog and cloud laboratories and hospital servers. The major goal of this urinary dataset is to design the urinary detection and processing expert system for the training and execution of acute bladder and

Table 2: Experiment Configuration

Configuration/Parameters	Values
Programming	Java, Python, XML
Sensors	Urinary testing tool
Sensor	Osteoarthritis Thigh Arduino
Agents	Mobile Socket Client, Server Socket Agent
S	100
A	100
R	$1 \sim 5$
ER	$0.1 \sim 0.8$
TR	$1 \sim 5$
Replay	20,000 (MB)
λ	$50 \sim 300$
Iterations	$50 \sim 5000$
i1	Temperature
i2	nausea 5000
i3	Lumbar 6000
i4	Urine Pushing and PEE 4500
i5	Micturition Pains 7000
i6	Burning of urethra 6000
i7	urinary bladder 6543
$k = 1$	Laptop X86 5000 GB, 16 RAM
$k = 2$	Laptop X86 15000 GB, 32 RAM
$k = 3$	Laptop X86 25000 GB, 64 RAM
$k = 4$	Laptop X86 50000 GB, 96 RAM
Layers N	5000
Z	6000
X	100 local training iterations

urinary system inflammation on different computing nodes. The urinary dataset[1] consisted of different urinary tasks. The goal is to divide and train the urinary dataset in different laboratories, hide the patient's personal data from aggregated nodes, and avoid sharing among laboratories. The results of the urinary tasks based on proposed schemes and existing baseline approaches without any replica of tasks are presented in Table 3. All workloads have a particular total processing time, deadline, and workload size. For instance, we schedule urinary task $i = 1$ on node $k1$, then determine the total time of 150, 100, and 79 ms, rewards of 5, 5, and 10, and accuracy rates of 92%, 93%, and 98%. However, we have assigned the deadlines, e.g., 80, 80, and 80 ms, during scheduling in the designated node. It is observed that FLP and RFLS still missed the deadline during the execution of the urinary task on

[1] https://archive.ics.uci.edu/ml/datasets/Acute+Inflammations

Table 3: Processing of Tasks on Node $k1$

Methods	Tasks	Total (ms)	deadline (ms)	Rewards	w_i (mb)	Accuracy (%)
FLP	$i = 1$	150	80	5	100	92
RFLS	$i = 1$	100	80	5	100	93
FLRLS	$i = 1$	79	80	10	100	98
FLP	$i = 2, 3$	(200, 190)	(200, 180)	4	(80, 110)	91, 92
RFLS	$i = 2, 3$	(201, 199)	(200, 180)	5	(80, 110)	93, 93.4
FLRLS	$i = 2, 3$	(189, 166)	(200, 180)	11	(80, 110)	98.9, 98.6
FLP	$i = 4, 5, 6$	(170, 99, 212, 297)	(150, 100, 200, 300)	5	(60, 80, 50, 110)	92.3, 92.5, 93
RFLS	$i = 4, 5, 6$	(159, 89, 199, 300)	(150, 100, 200, 300)	7	(60, 80, 50, 110)	92, 93.9, 93.3
FLRLS	$i = 4, 5, 6$	(130, 80, 170, 260)	(150, 100, 200, 300)	18	(60, 80, 50, 110)	98.9, 99, 98.5

node $k1$. The main reason is that the existing baseline approaches only focused on execution in limited space; it requires more time and resources and missed deadlines. At the same time, we determined the space required at the early stage to execute all urinary tasks within deadlines, as shown in Table 3.

The results of the urinary tasks based on proposed schemes and existing baseline approaches without any replica of tasks are shown in Table 3. All workloads have a particular total processing time, deadline, and workload size. For instance, we schedule urinary task $i = 2, 3$ on node $k2$, then determine the total time of 200, 190, 201, 199, 189, and 166 ms. Each urinary task has assigned deadlines, e.g., 200 and 180 ms, for all schemes during scheduling in the designated node. The obtained rewards and accuracy rates are 4, 5, 11, and 91%, 92%, 93%, 93.4%, and 98.9%, 98.6%. It is observed that FLP and RFLS still miss the deadline during the execution of the urinary task on node $k1$. The main reason is that the existing baseline approaches only focused on execution in limited space; it requires more time and resources and missed deadlines. At the same time, we determined the space required at the early stage to execute all urinary tasks within deadlines, as shown in Table 3.

We analyzed the results of the urinary tasks based on proposed schemes and existing baseline approaches without any replica of tasks as shown in Table 3. We added the five columns such as method, tasks, total time (ms), deadline (ms), computing node, and assigned size of urinary workloads. In Table 3, we analyzed that all workloads have a particular total processing time, deadline, and workload size. For instance, we scheduled urinary task $i = 4, 5, 6, 7$ on node $k3$, then determined the total time of 170, 99, 212, 297, 9, 89, 199, 300, and 30, 80, 170, 260 ms. However, we have assigned the deadlines, e.g., 50, 100, 200, and 300 ms, during scheduling in the designated node. The rewards and accuracy rates are 5, 7, 18, 92.3%, 92.5%, 93%, 92.9%, 93.3%, 98.9%, 99%, and 98.5%. It is observed that FLP and RFLS still miss the deadline during the execution of the urinary task on node $k1$. The main reason is that the existing baseline approaches only focused on execution in limited space; it requires more time and resources and missed deadlines. At the same time, we determined the space required at the early stage to execute all urinary tasks within deadlines, as shown in Table 3. We have seen that all the baseline schemes, FLP and RFLS, have a higher ratio of deadline missing compared to the proposed scheme. Another side, it increases the resource consumption ratio of the nodes during scheduling in distributed computing nodes. Existing schemes face critical resource scalability issues when executing urinary tasks. Therefore, the proposed scheme balances task deadlines and resource consumption to enable distributed processing on fog cloud networks.

5.1. Results Analysis

This section presents the results of our study and evaluates the performance of various algorithms for the problem at hand. Specifically, we employ deep DQN mobile and cloud agents with different connected states. We train and test these strategies based on available resources and processing constraints. We use Q-Table to maintain each state's performance and rewards for different mobile and server agents. Training involves connecting each state with another state, and the probability of transitioning to a new state depends on the knowledge of the previous state. Each state takes an action based on the previous state, which determines the scheduling strategy. The main advantage of this approach is that if the current state does not meet the requirements, it is marked as "0" in the table. Conversely, when a new state acts differently from the previous state, it receives a reward of "0". If the action is successful, the state is rewarded with "1". All the states generate "0" and "1" rewards, which are then stored in the Q-Table.

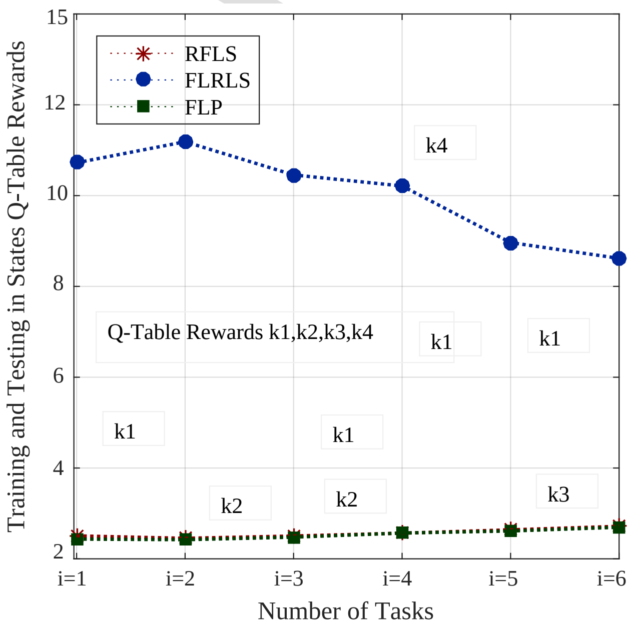


Figure 3: Training and Testing in Q-Table With Different Regarding Rewards

The testing involves comparing the current and previous states to obtain cumulative rewards in different states. We compared the training and testing performance in terms of cumulative rewards. The different tasks, represented as $i = 1, i = 2, i = 3, i = 4, i = 5, i = 6, ...I$, are plotted on the x-axis as shown in Figure 3. The y-axis represents the obtained rewards in the Q-Table on different computing nodes for the different number of tasks, as shown in Figure 3. For different numbers of tasks, the rewards vary based on the computing capability and available resources to meet the quality of service. In Figure 3, it is observed that task $i = 1$ scheduled on node $k1$ obtained rewards close to 3 in existing studies.

However, using FLRLS, the maximum rewards were achieved by executing it on node $k4$. The remaining tasks ($i = 2, 3, 4, 5, 6$) were scheduled on nodes $k2$ and $k3$, resulting in cumulative rewards close to 3. In contrast, FLRLS was scheduled on the federated learning-enabled node $k4$ to achieve maximum cumulative rewards compared to existing studies. Despite reducing rewards from the initial task $i = 1$ to $i = 6$ in our proposed scheme, attributed to task variations, resource availability, and optimal execution, the proposed method still obtained higher cumulative rewards. These rewards were stored in the Q-Table during training and testing in the network. The main reason behind our approach's success is the collection of resource information from different computing nodes and aggregation on the main node. A particular node is suboptimal; our scheme recommends another node for execution through the federated learning mechanism in different states. By employing this approach, we achieved optimal rewards for all tasks.

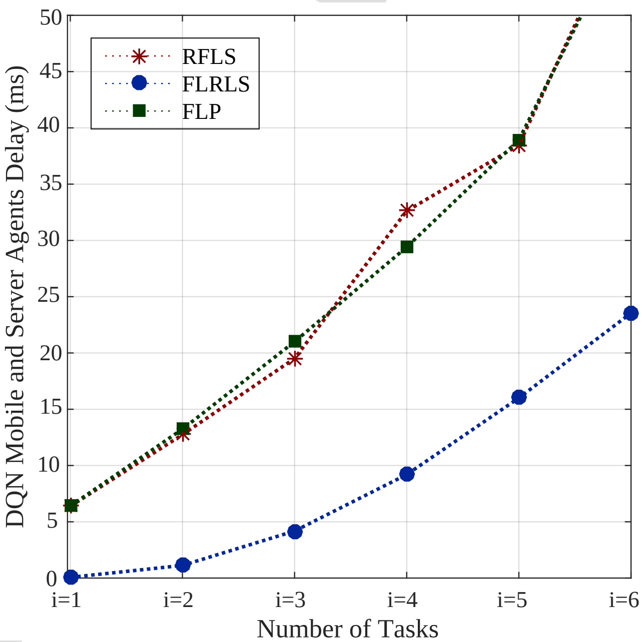


Figure 4: DQN Mobile Server Agents Delay Performances for Urinary Tasks.

The Q-table is a fundamental data structure that monitors states, actions, and corresponding expected rewards. It maps state-action pairs and Q-values, representing the agent's estimation of optimal future values. At the beginning of the

Q-Learning algorithm, the Q-table is initialized with zeros, indicating the agent's lack of knowledge about the environment. Through trial and error, the agent explores different actions in various states, gradually learning the expected rewards of each state-action pair. As the agent acquires new knowledge, it updates the Q-table by replacing existing Q-values with newly learned values. This iterative learning process, based on trial and error, is commonly known as exploration. The primary objective of the DQN mobile and server algorithms is to achieve the maximum expected reward when the agent takes specific action from a given state S. Once the agent has learned the DQN-Table for each state-action pair, it strives to maximize its expected reward by selecting the action with the highest expected value when in state S. This intentional selection of the best-known action in the state is known as exploitation. Our urinary laboratory clinical solution encounters various computing options, such as mobile and server. To address this, we have integrated the application on both the user and server sides.

By considering the nodes' capabilities in the environment, we schedule related tasks to different nodes. The mobile agent maximizes rewards by deciding whether to offload or execute tasks locally. We optimize rewards by ensuring that assigned tasks meet their requirements. As a result, the mobile agent undergoes a state transition and offloads task data to the server agent for further processing. The main objective of the DQN mobile and server agents is to execute all tasks with a minimal delay while ensuring they are completed within their respective deadlines, as illustrated in Figure 4. We have observed from the simulation result, as shown in Figure 4, that the proposed adaptive FLRLS outperformed for all tasks such as $i = 1, ..., i = 6$ in terms of delay (ms) compared to existing schemes. DQN is adaptive, where only and only make execution with the minimum delay and chooses the optimal node for processing in different states. All the actions were performed adaptive environment where bad and worst nodes in terms of resources are available, and the DQN mobile and servers agent chose only the optimal in terms of delay as shown in Figure 4. Hence, it is important adaptive agents scheme always outperforms in the distributed environment based on DQN-Table values for urinary tasks on different computing nodes, as shown in Figure 4.

In distributed healthcare applications, it is common to encounter a failure ratio of tasks. We analyzed to examine this failure ratio, specifically about missed execution deadlines during task scheduling on different computing nodes. Figure 5 illustrates the deadline failure ratio of all tasks, representing the percentage of tasks that failed to meet their execution deadlines out of the total number of tasks. The x-axis shows the number of tasks, whereas the y-axis shows the failure ratio of tasks in terms of deadline. The ratio starts from 0.1 to 1.1. During the execution of tasks on various computing nodes, including mobile devices and servers, if the task still fails to meet its deadline, we employ a rescheduling approach. This involves rescheduling the task on different nodes to mitigate the risk of missing the deadline. We determined the failure ratio in terms of the task deadline by dividing the failure ratio by the total number of assigned tasks to the different nodes for execution. Figure 5 shows that FLRLS has less deadline missing failure ratio than existing methods. The main

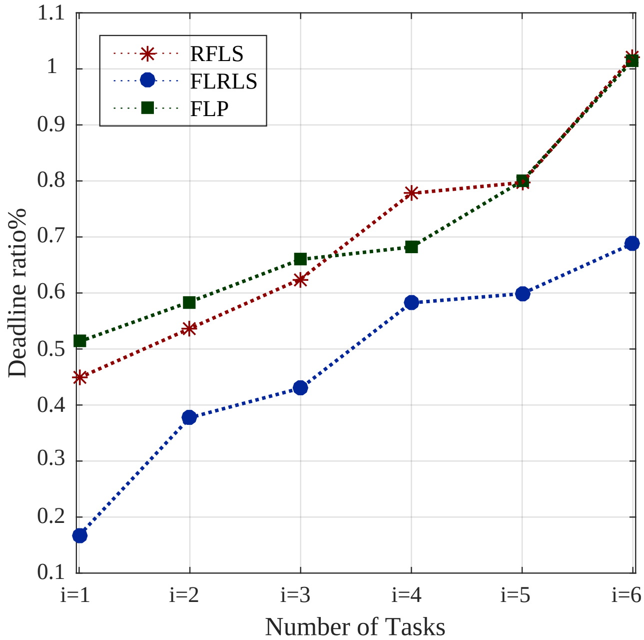


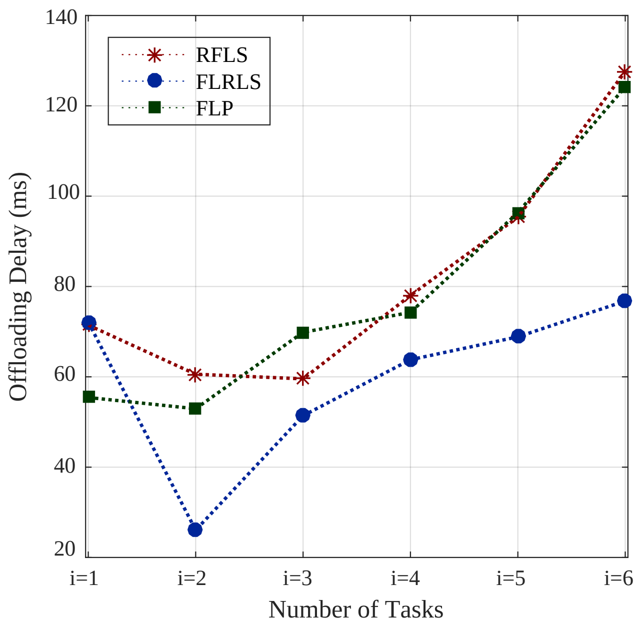
Figure 5: Performance of Deadline Failure ratio of Tasks in Different States.

reason is that we decide to schedule regarding all constraints, such as deadlines and rewards on different nodes. Regarding DQN Table values, we can identify which processing node can run all urinary with the minimum failure of deadlines compared to the existing method.

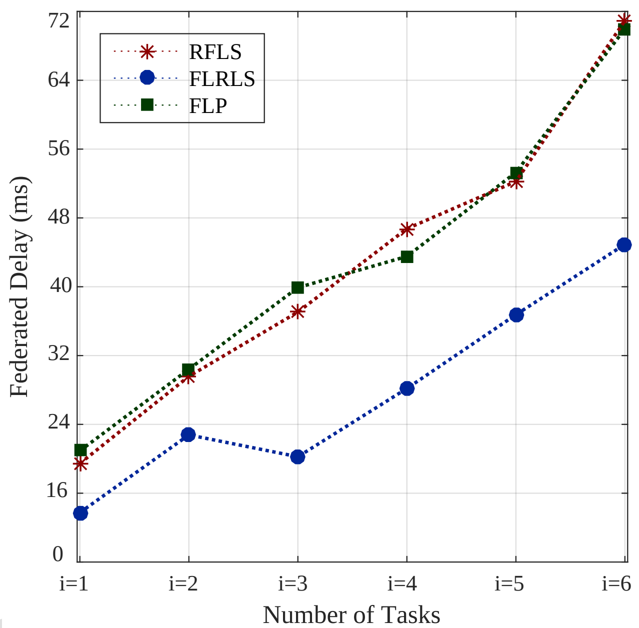
We analyzed the numerical outcomes of the urinary tasks by considering the suggested schemes and existing baseline approaches during their execution. The summarized results are presented in Table 3. Each workload was subject to constraints, including total processing time, deadline, accuracy, and workload size. We further examined the performance of different schemes, such as FLP and RFLS, through graphical representations based on the specified constraints. This analysis allowed us to gain insights into the comparative effectiveness of these schemes.

Each user installed the applications on their mobile phones and encountered initial delays, primarily due to the necessary setup of mobile device interfaces, which requires a certain amount of time. The study introduced the DQN-Mobile and Server Agents Scheme to address this issue. In this scheme, the agent initiates activities on the local devices, albeit with a slightly longer delay. However, offloading tasks to the servers are implemented to mitigate additional delays, as depicted in Figure 6a. This approach helps minimize overall latency and improve the user experience.

In our system, all the states are connected, trained, and tested in terms of quality of service (QoS) and resource availability during the network processing of applications. As a result, as shown in Figure 6a, the study proposed an adaptive agent that schedules offloaded tasks at available resource nodes, such as $k1$, to meet the requirements of tasks with the least amount of delay. We utilized DQN mobile and server



(a) Initial State Mobile Agent Performance



(b) Scheduling Performance of Delays.

Figure 6: Local and Offloading Performance Analysis

agents as adaptive policies to execute tasks at different computing nodes, including mobile devices and servers, according to given constraints and obtained information. One of the primary advantages of employing a DQN-based strategy is its ability to find optimal solutions dynamically during runtime. The deep Q-learning algorithm utilizes a deep neural network to approximate values. It begins by inputting the initial state into the neural network, which calculates all possible actions based on the Q-Table values. This process ensures task execution occurs on the most optimal nodes until it either meets the given constraints (e.g., deadline) or completes the task. However, initial scheduling still has some processing delays, as shown in Figure 6a due to the availability of resources when tasks are offloaded to the system. The proposed agent is adaptive, rescheduling previously scheduled tasks from one node to another, such as $k1 \sim k2 \sim k3, \sim k4 \in K$ as shown in Figure 6b. Nonetheless, resources have fluctuated, and due to scheduling delays, the failure ratio of tasks meeting their deadlines is expected to rise. As a result, scheduling tasks based on available resources rather than the QoS of network resources, as shown in Figure 6b. As a result, for all tasks, the agent must have trade-off performances between delays and failure ratios of the deadline. Task traffic load balancing on single nodes has longer delays. Therefore, the workloads among the available nodes must be shared based on adaptive techniques. The existing fixed and dynamic state Q-learning with policy techniques (e.g., exploitative and exploratory searching trade-off schemes) suffered from load balancing. It incurred higher traffic delays, as shown in Figure 6b. One possible answer is an adaptive technique in which tasks are split and given to different nodes. Another aspect is that the adaptive approach efficiently schedules tasks and meets the network's trade-off between delays and deadlines, as shown in Figure 6b.

5.2. Algorithm Comparison Result Analysis

The study implemented the closely related algorithms [9, 10, 17] related to reinforcement learning and federated learning that can be executed on the urinary dataset. It analyzed their results with both concave and convex functions. We only implemented these algorithms in three different schemes such as FLRLS, FLP (federated learning policy) [9, 10, 17], RFLS (reinforce federated learning scheme) [19, 20, 22, 24, 25] that have different q-learning with both DQN and static and dynamic approaches in their models. Two baseline algorithms, FLP and RFLS, are implemented in the simulation environment. The main reason is that these algorithms considered the same constraints and same functions during their experimental analysis for the urine dataset. Future work will add more methods to optimize the simulation results with multiple constraints inside multi-objectives. However, we have limited constraints and functions, as well as the urine dataset. Therefore, we only implemented two baseline algorithms in the simulation for performance analysis on the urine dataset.

Figure 7 depicts the performance of all Inflammation of the Urinary Bladder workloads (e.g., tasks) with lower FLRLS delays than existing federated and reinforcement learning techniques. The study proposed an episodic and adaptive DQN policy that optimizes task continuous execution performance in distributed fog cloud networks. Figures 7a and 7b show that FLRLS outperformed existing FLP and RFLS schemes during the execution of tasks in the system. The primary reason is that FLP only trained and tested the resource contents at the local nodes and shared them

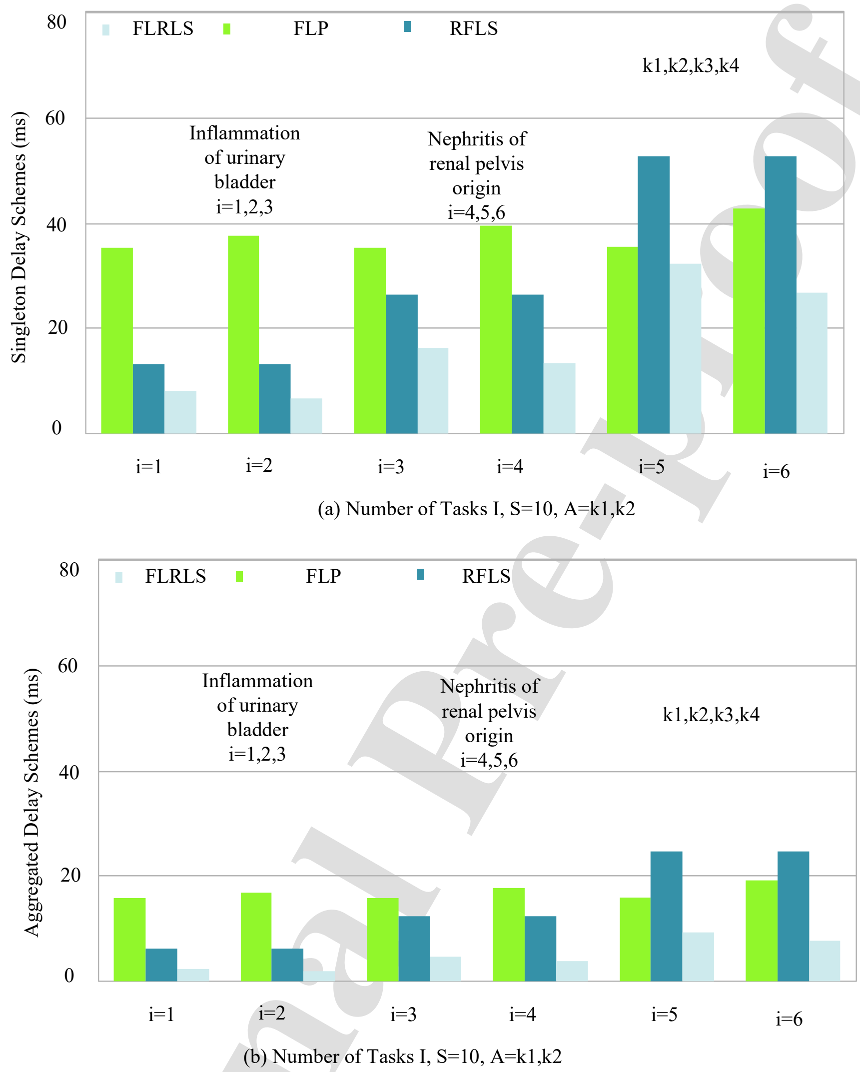
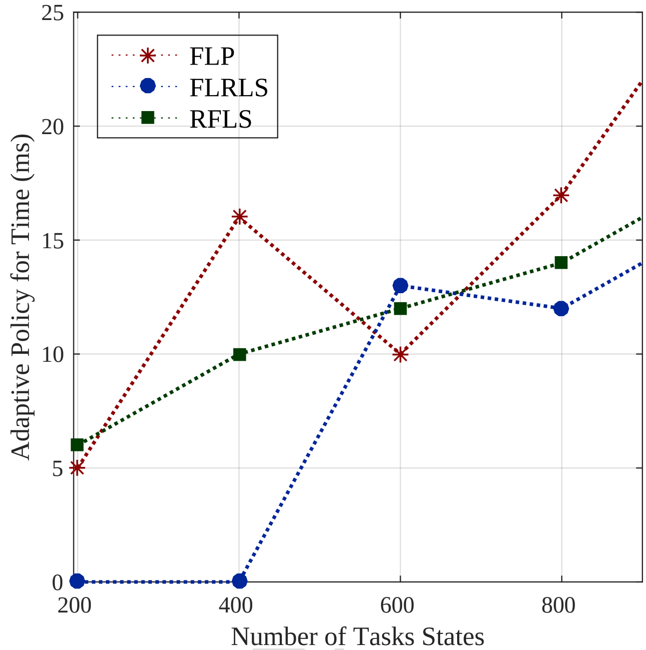


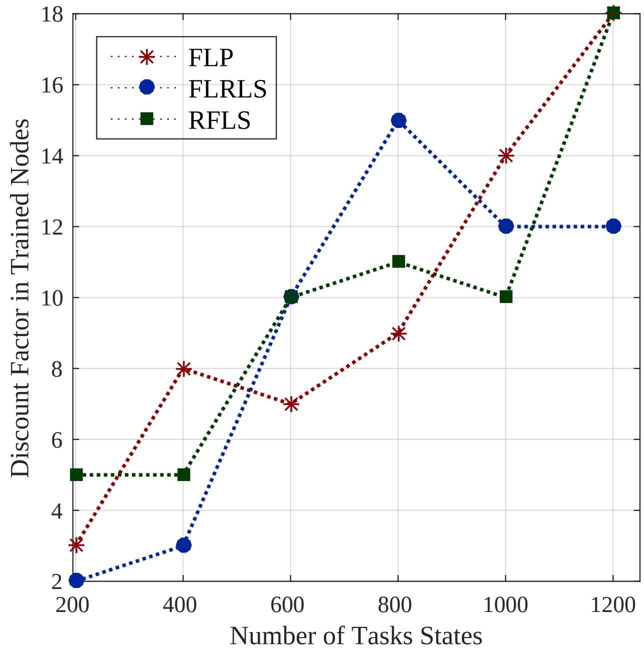
Figure 7: Processing Delays Performance Based on All Schemes.

with the centralized node. Therefore, it incurred longer delays. Meanwhile, the RFLS scheme trained and tested resource models in different states. Based on fixed states, these methods used the backpropagation method to run workloads ad hoc and continuously. The main benefit of FLRLS is that all local states are flexible and can quickly adjust to new resource changes in the episodic approach to meet the quality of service and application delays.

Figure 8a depicts the performance of all Inflammation of the Urinary Bladder workloads (e.g., tasks) with lower FLRLS delays than existing federated and reinforcement learning techniques. The study proposed an episodic and adaptive DQN policy that optimizes task continuous execution performance in distributed fog cloud networks. Figure 8a shows that FLRLS outperformed existing FLP and RFLS schemes during the execution delays of tasks and maximized the rewards in the sys-



(a) Delay Performance of Adaptive Scheduling Tasks.



(b) Delay Performance and Maximized Rewards.

Figure 8: Training and Execution Performance

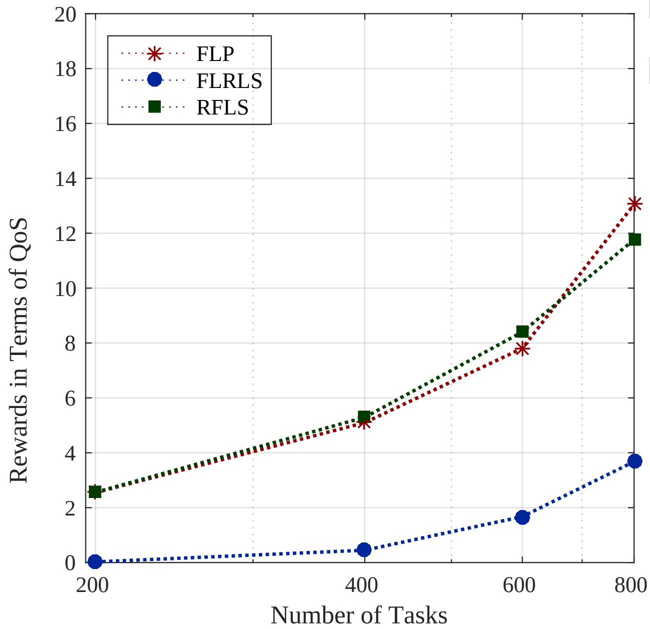
tem. The primary reason is that FLP only trained and tested the resource contents at the local nodes and shared them with the centralized node. Therefore, it incurred longer delays. Meanwhile, the RFLS scheme trained and tested resource models in different states. Based on fixed states, these methods used the backpropagation method to run workloads ad hoc and continuously. The main benefit of FLRLS is that all local states are flexible and can quickly adjust to new resource changes in the episodic approach to meet the quality of service and application delays.

Figure 8a depicts the performance of all Inflammation of the Urinary Bladder workloads (e.g., tasks) with lower FLRLS delays than existing federated and reinforcement learning techniques. The study proposed an episodic and adaptive DQN policy that optimizes task continuous execution performance in distributed fog cloud networks. Figure 8a shows that FLRLS outperformed existing FLP and RFLS schemes during the execution delays of tasks and maximized the rewards in the system. The primary reason is that FLP only trained and tested the resource contents at the local nodes and shared them with the centralized node. Therefore, it incurred longer delays. Meanwhile, the RFLS scheme trained and tested resource models in different states. Based on fixed states, these methods used the backpropagation method to run workloads ad hoc and continuously. The main benefit of FLRLS is that all local states are flexible and can quickly adjust to new resource changes in the episodic approach to meet the quality of service and application delays. Figure 8a depicts the performance of all Inflammation of the Urinary Bladder workloads (e.g., tasks) with lower FLRLS delays than existing federated and reinforcement learning techniques. The study proposed an episodic and adaptive DQN policy that optimizes task continuous execution performance with episodic rewards in distributed fog cloud networks.

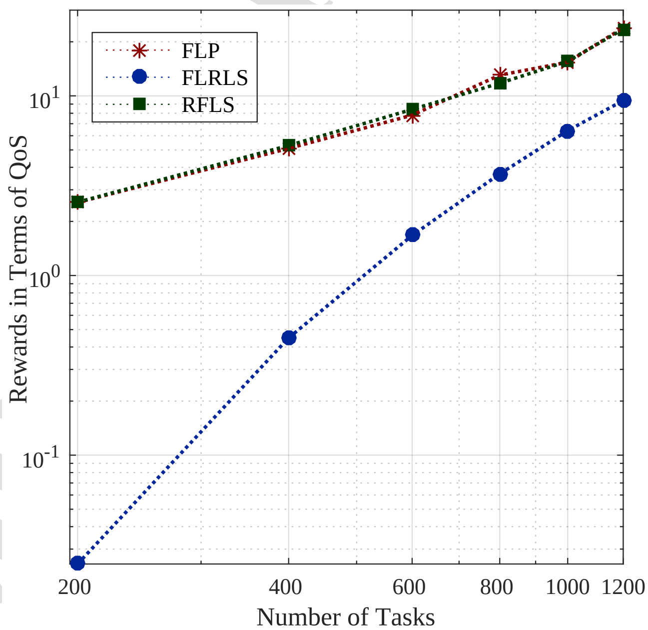
Figure 8a discount factor differentiates between the failure and rewards during execution. Figure 8a shows that FLRLS outperformed existing FLP and RFLS schemes during the execution delays of tasks and maximized the rewards in the system. The primary reason is that FLP only trained and tested the resource contents at the local nodes and shared them with the centralized node. Therefore, it incurred longer delays. Meanwhile, the RFLS scheme trained and tested resource models in different states. Based on fixed states, these methods used the backpropagation method to run workloads ad hoc and continuously. The main benefit of FLRLS is that all local states are flexible and can quickly adjust to new resource changes in the episodic approach to meet the quality of service and application delays.

Figure 8b depicts the performance of all Inflammation of the Urinary Bladder workloads (e.g., tasks) with lower FLRLS delays than existing federated and reinforcement learning techniques. The study proposed an episodic and adaptive DQN policy that optimizes task continuous execution performance with the discount factor in distributed fog cloud networks. Figure 8b discount factor differentiates between the failure and rewards during execution. Figure 8b shows that FLRLS outperformed existing FLP and RFLS schemes during the execution delays of tasks and maximized the rewards in the system. The primary reason is that FLP only trained and tested the resource contents at the local nodes and shared them with the centralized node.

Therefore, it incurred longer delays. Meanwhile, the RFLS scheme trained and tested resource models in different states. Based on fixed states, these methods used the backpropagation method to run workloads ad hoc and continuously. The main benefit of FLRLS is that all local states are flexible and can quickly adjust to new resource changes in the episodic approach to meet quality of service and application delays.



(a) Episodic Rewards Performances of Methods.



(b) Delay Performance and Maximized Rewards.

Figure 9: Training and Execution Performance

Figure 9a depicts the performance of all Inflammation of the Urinary Bladder workloads (e.g., tasks) with lower FLRLS delays than existing federated and reinforcement learning techniques. The study proposed an episodic and adaptive DQN policy that optimizes task continuous execution performance with the discount factor in distributed fog cloud networks. Figure 9a discount factor differentiates between the failure and rewards during execution. Figure 9a shows that FLRLS outperformed existing FLP and RFLS schemes during the execution delays of tasks and maximized the rewards in the system. The primary reason is that FLP only trained and tested the resource contents at the local nodes and shared them with the centralized node. Therefore, it incurred longer delays. Meanwhile, the RFLS scheme trained and tested resource models in different states. Based on fixed states, these methods used the backpropagation method to run workloads ad hoc and continuously. The main benefit of FLRLS is that all local states are flexible and can quickly adjust to new resource changes in the episodic approach to meet the quality of service (QoS) and application delays.

We determined the quality of service requirements (e.g., deadline) of tasks during the scheduling in the computing nodes. The rewards are determined in terms of meeting the deadline. Figure 9a shows the rewards for quality of service performances of 800 tasks with all FLP, FLRLS, and RFLS schemes. The y-axis represents the number of missing deadline tasks, and the x-axis represents the number of tasks during scheduling with the FLP, FLRLS, and RFLS schemes. It can be observed from Figure 9a FLRLS outperformed and less deadline missing tasks as compared to FLP and RFLS schemes. The main is that FLRLS determined the resources at the early stage and the required resources before the execution of tasks at different nodes. However, it is a complex process; maybe a few tasks missed the quality of service (e.g., deadline) with the FLRLS. The main reason is the number of tasks and available resources incurred with fluctuation, and our computing resources are not auto-scale and scaled down by the algorithm. Dynamic scaling up and scaling down required proper resource provisioning. Therefore, runtime scaling is necessary for resource processing urinary tasks on the different nodes. If the number of tasks increases from 800 to 1200 (e.g., replications), the quality of service missing deadline numbers will be increased, as shown in Figure 9b. However, FLRLS still outperformed all existing schemes in terms of quality of service for all urinary tasks during processing on the different computing nodes.

6. Conclusion

We address the problem as a combinatorial optimization problem, incorporating two objectives: optimizing accuracy, considered as a concave objective, and minimizing computation delay, treated as a convex objective. Utilizing lab urine data, we conducted experiments to showcase the efficacy of the Federated Learning and Reinforcement Learning Scheme (FLRLS) enabled framework in simultaneously minimizing both objectives. By employing deterministic agents, FLRLS optimizes the exploration and exploitation of urinary data, with the urinary tract being determined by a centralized, aggregated node.

Simulation results have demonstrated a notable improvement of 5% accuracy, as presented in Table 3, while reducing the overall computation delay. As part of our future work, we aim to propose a secure and decentralized blockchain system for digital healthcare applications. This proposed system will consider various constraints across diverse mobile fog and cloud networks, including security, resource availability, and task delays. Currently, the proposed method has achieved significant results in digital healthcare. In the future, it holds promising potential for extension to address portfolio problems with high requirements for both privacy and efficiency [58, 59], as well as image tracking problems [16, 60].

Acknowledgment

This work is designed at the Mobile and Technology Lab, Kristiania University, Oslo, Norway.

References

[1] C. Zhu, Z. Xu, Y. Gu, S. Zheng, X. Sun, J. Cao, B. Song, J. Jin, Y. Liu, X. Wen, et al., Prediction of post-stroke urinary tract infection risk in immobile patients using machine learning: an observational cohort study, Journal of Hospital Infection 122 (2022) 96–107.

[2] S. Biswas, A. Pal, P. Chakraborty, K. Chaudhury, S. Das, Machine learning based urinary ph sensing using polyaniline deposited paper device and integration of smart web app interface: Theory to application, Biosensors and Bioelectronics 211 (2022) 114332.

[3] S. S. Mahdi, G. Battineni, M. Khawaja, R. Allana, M. K. Siddiqui, D. Agha, How does artificial intelligence impact digital healthcare initiatives? a review of ai applications in dental healthcare, International Journal of Information Management Data Insights 3 (1) (2023) 100144.

[4] Y. Chen, M. Wang, A. A. Heidari, B. Shi, Z. Hu, Q. Zhang, H. Chen, M. Mafarja, H. Turabieh, Multi-threshold image segmentation using a multi-strategy shuffled frog leaping algorithm, Expert Systems with Applications 194 (2022) 116511.

[5] M. Yu, M. Han, X. Li, X. Wei, H. Jiang, H. Chen, R. Yu, Adaptive soft erasure with edge self-attention for weakly supervised semantic segmentation: thyroid ultrasound image case study, Computers in Biology and Medicine 144 (2022) 105347.

[6] C. Zhao, H. Wang, H. Chen, W. Shi, Y. Feng, Jamsnet: A remote pulse extraction network based on joint attention and multi-scale fusion, IEEE Transactions on Circuits and Systems for Video Technology (2022) 1–1doi:10.1109/TCSVT.2022.3227348.

[7] N. Goździkiewicz, D. Zwolińska, D. Polak-Jonkisz, The use of artificial intelligence algorithms in the diagnosis of urinary tract infectionsa literature review, Journal of Clinical Medicine 11 (10) (2022) 2734.

[8] Y. Chen, H. Gan, H. Chen, Y. Zeng, L. Xu, A. A. Heidari, X. Zhu, Y. Liu, Accurate iris segmentation and recognition using an end-to-end unified framework based on madnet and dsanet, Neurocomputing 517 (2023) 264–278.

[9] A. K. Uttam, Urinary system diseases prediction using supervised machine learning-based model: Xgboost and random forest, in: International Conference on Artificial Intelligence and Sustainable Engineering, Springer, 2022, pp. 179–185.

[10] R. Dhanalakshmi, T. Sudalaimuthu, K. Radhakrishnan, Early detection of sepsis using lstm and reinforcement learning, in: Applications of Computational Methods in Manufacturing and Product Design: Select Proceedings of IPDIMS 2020, Springer, 2022, pp. 297–306.

[11] J. Mulani, S. Heda, K. Tumdi, J. Patel, H. Chhinkaniwala, J. Patel, Deep reinforcement learning based personalized health recommendations, Deep learning techniques for biomedical and health informatics (2020) 231–255.

[12] X. Zhang, J. Zheng, L. Zhao, Z. Zhou, Z. Lin, Tensor recovery with weighted tensor average rank, IEEE Transactions on Neural Networks and Learning Systems (2022) 1–15doi:10.1109/TNNLS.2022.3182541.

[13] M. Zhu, X. Guan, Z. Li, L. He, Z. Wang, K. Cai, semg-based lower limb motion prediction using cnn-lstm with improved pca optimization algorithm, Journal of Bionic Engineering 20 (2) (2023) 612–627.

[14] J. Wang, D. Wu, Y. Gao, X. Wang, X. Li, G. Xu, W. Dong, Integral real-time locomotion mode recognition based on ga-cnn for lower limb exoskeleton, Journal of Bionic Engineering 19 (5) (2022) 1359–1373. doi:10.1007/s42235-022-00230-z.
URL https://doi.org/10.1007/s42235-022-00230-z

[15] C.-f. Chen, Z.-j. Du, L. He, Y.-j. Shi, J.-q. Wang, W. Dong, A novel gait pattern recognition method based on lstm-cnn for lower limb exoskeleton, Journal of Bionic Engineering 18 (2021) 1059–1072.

[16] W. Liu, Z. Jing, H. Pan, L. Qiao, H. Leung, W. Chen, Distance-directed target searching for a deep visual servo sma driven soft robot using reinforcement learning, Journal of Bionic Engineering 17 (2020) 1126–1138.

[17] Z. Hu, K. Shaloudegi, G. Zhang, Y. Yu, Federated learning meets multi-objective optimization, IEEE Transactions on Network Science and Engineering (2022).

[18] F. Ma, T. Sun, L. Liu, H. Jing, Detection and diagnosis of chronic kidney disease using deep learning-based heterogeneous modified artificial neural network, Future Generation Computer Systems 111 (2020) 17–26.

[19] S. Sudharson, P. Kokil, An ensemble of deep neural networks for kidney ultrasound image classification, Computer Methods and Programs in Biomedicine 197 (2020) 105709.

[20] J. Nandhini, S. Joshi, K. Anuratha, Federated learning based prediction of chronic kidney diseases, in: 2022 1st International Conference on Computational Science and Technology (ICCST), IEEE, 2022, pp. 1–6.

[21] S. Rajendran, Z. Xu, W. Pan, A. Ghosh, F. Wang, Data heterogeneity in federated learning with electronic health records: Case studies of risk prediction for acute kidney injury and sepsis diseases in critical care, medRxiv (2022) 2022–08.

[22] Y. Kumar, R. Singla, Federated learning systems for healthcare: perspective and recent progress, Federated Learning Systems: Towards Next-Generation AI (2021) 141–156.

[23] A. O. Almagrabi, R. Ali, D. Alghazzawi, A. AlBarakati, T. Khurshaid, A reinforcement learning-based framework for crowdsourcing in massive health care internet of things, Big data 10 (2) (2022) 161–170.

[24] A. H. Sodhro, A. Majumdar, P. Khuwuthyakorn, O. Thinnukool, A lightweight secure adaptive approach for internet-of-medical-things healthcare applications in edge-cloud-based networks, Sensors 22 (6) (2022) 2379.

[25] X. Li, Content aware task scheduling framework for mobile workflow applications in heterogeneous mobile-edge-cloud paradigms: Catsa framework, in: 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), IEEE, 2019, pp. 242–249.

[26] A. Rahman, M. S. Hossain, G. Muhammad, D. Kundu, T. Debnath, M. Rahman, M. S. I. Khan, P. Tiwari, S. S. Band, Federated learning-based ai approaches in smart healthcare: concepts, taxonomies, challenges and open issues, Cluster Computing (2022) 1–41.

[27] J. Guerrero, G. Miró-Amarante, A. Martín, Decision support system in health care building design based on case-based reasoning and reinforcement learning, Expert Systems with Applications 187 (2022) 116037.

[28] I. A. Zamfirache, R.-E. Precup, R.-C. Roman, E. M. Petriu, Reinforcement learning-based control using q-learning and gravitational search algorithm with experimental validation on a nonlinear servo system, Information Sciences 583 (2022) 99–120.

[29] M. Schenke, O. Wallscheid, A deep q-learning direct torque controller for permanent magnet synchronous motors, IEEE Open Journal of the Industrial Electronics Society 2 (2021) 388–400.

[30] J. Vimala Devi, K. Kavitha, Adaptive deep q learning network with reinforcement learning for crime prediction, Evolutionary Intelligence (2022) 1–12.

[31] S. Y. Luis, D. G. Reina, S. L. T. Marín, A multiagent deep reinforcement learning approach for path planning in autonomous surface vehicles: The ypacaraí lake patrolling case, IEEE Access 9 (2021) 17084–17099.

[32] M. Deuker, L. F. Stolzenbach, C. C. Ruvolo, L. Nocera, M. Mansour, Z. Tian, F. C. Roos, A. Becker, L. A. Kluth, D. Tilki, et al., Micropapillary versus urothelial carcinoma of the urinary bladder: stage at presentation and efficacy of chemotherapy across all stagesa seer-based study, European Urology Focus 7 (6) (2021) 1332–1338.

[33] A. Pottegård, K. B. Kristensen, S. Friis, J. Hallas, J. B. Jensen, M. Nørgaard, Urinary tract infections and risk of squamous cell carcinoma bladder cancer: a danish nationwide case–control study, International Journal of Cancer 146 (7) (2020) 1930–1936.

[34] Q. Wang, S. Zhu, X. Huang, X. Liu, J. Liu, G. Tian, Prognostic value of systemic immune-inflammation index in patients with urinary system cancers: a meta-analysis, Eur Rev Med Pharmacol Sci 25 (3) (2021) 1302–1310.

[35] T. Aboushousha, O. Hammam, A. Aref, A. Kamel, M. Badawy, A. A. Hamid, Tissue profile of cdk4 and stat3 as possible innovative therapeutic targets in urinary bladder cancer, Asian Pacific Journal of Cancer Prevention: APJCP 21 (2) (2020) 547.

[36] T. Morten Groenli, A. Majumdar, P. Khuwuthyakorn, F. Hussain Khoso, O. Thinnukool, Potent blockchain-enabled socket rpc internet of healthcare things (ioht) framework for medical enterprises, Sensors 22 (12) (2022) 4346.

[37] M. Elhoseny, M. A. Mohammed, M. M. Jaber, Sfdwa: Secure and fault-tolerant aware delay optimal workload assignment schemes in edge computing for internet of drone things applications, Wireless Communications and Mobile Computing 2022 (2022).

[38] X. Li, Mobility and fault aware adaptive task offloading in heterogeneous mobile cloud environments, EAI Endorsed Transactions on Mobile Communications and Applications 5 (16) (2019).

[39] M. Hussain, L.-F. Wei, S. Wali, S. Ali, A. Hussain, Energy and performance-efficient task scheduling in heterogeneous virtualized cloud computing, Sustainable Computing: Informatics and Systems 30 (2021) 100517.

[40] W. Qiu, J. Xie, Y. Shen, J. Xu, J. Liang, Endoscopic image recognition method of gastric cancer based on deep learning model, Expert Systems 39 (3) (2022) e12758.

[41] C. Ying, Z. Huang, C. Ying, Accelerating the image processing by the optimization strategy for deep learning algorithm dbn, EURASIP Journal on Wireless Communications and Networking 2018 (2018) 1–8.

[42] X.-T. Wu, Y. Wang, Y. Wan, W. Wu, Annotation is easy: Learning to generate a shadow mask, Computers & Graphics 104 (2022) 152–161.

[43] Z. Wu, G. Li, S. Shen, X. Lian, E. Chen, G. Xu, Constructing dummy query sequences to protect location privacy and query privacy in location-based services, World Wide Web 24 (2021) 25–49.

[44] H. Su, D. Zhao, H. Elmannai, A. A. Heidari, S. Bourouis, Z. Wu, Z. Cai, W. Gui, M. Chen, Multilevel threshold image segmentation for covid-19 chest radiography: a framework using horizontal and vertical multiverse optimization, Computers in Biology and Medicine 146 (2022) 105618.

[45] A. Qi, D. Zhao, F. Yu, A. A. Heidari, Z. Wu, Z. Cai, F. Alenezi, R. F. Mansour, H. Chen, M. Chen, Directional mutation and crossover boosted ant colony optimization with application to covid-19 x-ray image segmentation, Computers in biology and medicine 148 (2022) 105810.

[46] K.-W. Kim, S.-H. Park, G. Im, S. B. Lee, Y. Baba, C. Lee, Y. H. Choi, H.-H. Kim, Cfd study on vesicoureteral reflux in the urinary tract with double j stent, Computers in Biology and Medicine 145 (2022) 105456.

[47] O. Nikkinen, T. Kolehmainen, T. Aaltonen, E. Jämsä, S. Alahuhta, M. Vakkala, Developing a supervised machine learning model for predicting perioperative acute kidney injury in arthroplasty patients, Computers in Biology and Medicine 144 (2022) 105351.

[48] S. Amiri, M. Akbarabadi, F. Abdolali, A. Nikoofar, A. J. Esfahani, S. Cheraghi, Radiomics analysis on ct images for prediction of radiation-induced kidney damage by machine learning models, Computers in Biology and Medicine 133 (2021) 104409.

[49] J. K. Cheong, V. Popov, E. Alchera, I. Locatelli, M. Alfano, L. Menichetti, P. Armanetti, M. Maturi, M. C. Franchini, E. H. Ooi, et al., A numerical study to investigate the effects of tumour position on the treatment of bladder cancer in mice using gold nanorods assisted photothermal ablation, Computers in biology and medicine 138 (2021) 104881.

[50] S. Wu, P. Mao, R. Li, Z. Cai, A. A. Heidari, J. Xia, H. Chen, M. Mafarja, H. Turabieh, X. Chen, Evolving fuzzy k-nearest neighbors using an enhanced

sine cosine algorithm: Case study of lupus nephritis, Computers in Biology and Medicine 135 (2021) 104582.

[51] D. Liu, X. Peng, X. Liu, Y. Li, Y. Bao, J. Xu, X. Bian, W. Xue, D. Qian, A real-time system using deep learning to detect and track ureteral orifices during urinary endoscopy, Computers in Biology and Medicine 128 (2021) 104104.

[52] Z. Ge, L. Tang, Y. Peng, M. Zhang, J. Tang, X. Yang, Y. Li, Z. Wu, G. Yuan, Design of a rapid diagnostic model for bladder compliance based on real-time intravesical pressure monitoring system, Computers in Biology and Medicine 141 (2022) 105173.

[53] F. J. Moreno-Barea, L. Franco, D. Elizondo, M. Grootveld, Application of data augmentation techniques towards metabolomics, Computers in Biology and Medicine 148 (2022) 105916.

[54] J. A. Locke, S. Neu, S. Herschorn, Morbidity and predictors of delayed recognition of iatrogenic ureteric injuries, Canadian Urological Association Journal 16 (1) (2022) E1.

[55] X. Zhang, J. Zheng, D. Wang, G. Tang, Z. Zhou, Z. Lin, Structured sparsity optimization with non-convex surrogates of $\ell_{2,0}$2,0-norm: A unified algorithmic framework, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (5) (2023) 6386–6402. doi:10.1109/TPAMI.2022.3213716.

[56] X. Zhang, D. Wang, Z. Zhou, Y. Ma, Robust low-rank tensor recovery with rectification and alignment, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (1) (2021) 238–255. doi:10.1109/TPAMI.2019.2929043.

[57] X. Zhang, J. Zheng, D. Wang, L. Zhao, Exemplar-based denoising: A unified low-rank recovery framework, IEEE Transactions on Circuits and Systems for Video Technology 30 (8) (2020) 2538–2549. doi:10.1109/TCSVT.2019.2927603.

[58] Y. Chen, A. Zhou, S. Das, Utilizing dependence among variables in evolutionary algorithms for mixed-integer programming: A case study on multi-objective constrained portfolio optimization, Swarm and Evolutionary Computation 66 (2021) 100928.

[59] Y. Chen, A. Zhou, Multiobjective portfolio optimization via pareto front evolution, Complex & Intelligent Systems 8 (5) (2022) 4301–4317. doi:10.1007/s40747-022-00715-8.
URL https://doi.org/10.1007/s40747-022-00715-8

[60] X. Zhang, W. Hu, N. Xie, H. Bao, S. Maybank, A robust tracking system for low frame rate video, International Journal of Computer Vision 115 (3) (2015) 279–304. doi:10.1007/s11263-015-0819-8.
URL https://doi.org/10.1007/s11263-015-0819-8

1. This paper devises the federated learning and reinforcement learning based schemes to find the best way to find CKD diseases with less delay in distributed fog and cloud networks.
2. This work presents the lightweight federated reinforcement learning scheme (FLRLS) for urinary healthcare applications, acute urinary tasks.
3. Based on the simulation results, FLRLS is an adaptive and efficient algorithmic framework for the urinary applications.
4. Study designed the mathematical model for the considered problem with the different constraints.
5. The study suggested the open source code of the designed CKD diseases with the exploited dataset on the online platform.

Conflicts of Interest: The authors declare no conflict of interest.