International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(7s) 2026 pp. 185-200
https://doi.org/10.70917/ijcisim-2026-3066
Article

Predictive Latency-Aware Federated Deep Reinforcement Learning for Adaptive Task Scheduling in IoT-Enabled Fog Computing

Asha S[1], Chandrappa D N[2],

[1]Department of Electronics and Communication Engineering, East Point College of Engineering and Technology, Bengaluru - 560049

[2]Department of Electronics and Communication Engineering, East Point College of Engineering and Technology, Bengaluru - 560049

***Corresponding Author**: Prof.Asha S

Abstract: The increasing deployment of latency-sensitive Internet of Things (IoT) applications has intensified the need for intelligent task scheduling mechanisms in fog computing environments. Conventional scheduling approaches, including heuristic and centralized machine learning techniques, often fail to adapt to dynamic workload variations and mobility-induced network changes, resulting in increased task latency. This paper proposes a Predictive Latency-Aware Federated Deep Reinforcement Learning (PLA-FDRL) framework for adaptive task scheduling in IoT-enabled fog networks. The proposed framework integrates latency prediction, mobility-aware fog node selection, and federated deep reinforcement learning to proactively allocate tasks to optimal fog resources. Each fog node independently trains a Deep Q-Network (DQN) using local observations and periodically participates in federated aggregation without sharing raw data. A latency-aware reward function jointly minimizes transmission, queueing, processing, and migration delays. Experimental evaluation under dynamic IoT workloads demonstrates significant reductions in average task latency and response time compared with FCFS, Round Robin, centralized DQN, and conventional federated reinforcement learning schedulers. Results indicate that the proposed framework improves responsiveness and scalability while preserving data privacy.

Keywords: Fog Computing, Federated Learning, Deep Reinforcement Learning, Task Scheduling, Latency Optimization, IoT.

1. INTRODUCTION

The rapid growth of the Internet of Things (IoT) has revolutionized modern communication and computing paradigms by enabling billions of interconnected devices to generate, process, and exchange data continuously [1], [2]. IoT applications such as smart healthcare, intelligent transportation systems, industrial automation, smart cities, and environmental monitoring generate massive volumes of data that require real-time processing and decision-making [3]. Many of these applications are highly latency-sensitive and demand immediate responses to ensure reliability, safety, and Quality of Service (QoS) [4]. Traditional cloud computing architectures often fail to satisfy these stringent latency requirements due to long communication distances, network congestion, and centralized processing limitations [5], [6].

Fog computing has emerged as an effective paradigm to address these challenges by extending cloud services closer to end devices [7], [8]. By deploying computational and storage resources at the network edge, fog computing significantly reduces communication delay and supports real-time data processing [9]. However, the effectiveness of fog computing largely depends on efficient task scheduling mechanisms that determine how computational tasks are allocated among available fog nodes [10]. In dynamic IoT environments, the continuous variation in workload intensity, resource availability, network conditions, and device mobility makes task scheduling a complex optimization problem [11].

Conventional task scheduling approaches such as First Come First Serve (FCFS), Round Robin (RR), Min-Min, and Max-Min employ predefined scheduling policies that do not adapt to changing environmental conditions [12]. Although these methods are computationally simple, they often result in poor resource utilization, increased queue waiting time, and higher task latency under dynamic workloads [13]. To overcome these limitations, machine

learning and reinforcement learning techniques have recently been investigated for intelligent task scheduling in fog computing environments [14]. Reinforcement Learning (RL) enables scheduling agents to learn optimal decision-making policies through continuous interaction with the environment, thereby improving adaptability and scheduling efficiency [15].

Deep Reinforcement Learning (DRL), particularly Deep Q-Networks (DQNs), has demonstrated promising performance in solving complex scheduling problems with large state spaces [16], [17]. However, most existing DRL-based schedulers adopt centralized learning architectures that require collecting large volumes of task and resource information at a central controller [18]. Such centralized approaches introduce scalability limitations, communication overhead, and privacy concerns, especially in geographically distributed fog environments [19]. Federated Learning (FL) has emerged as a promising distributed learning paradigm that enables multiple nodes to collaboratively train machine learning models without exchanging raw data [20]. By sharing only model parameters, FL preserves data privacy while reducing communication costs and supporting decentralized intelligence [21].

Despite recent advances in Federated Reinforcement Learning (FRL)-based scheduling, several critical challenges remain unresolved [22]. Existing approaches primarily rely on instantaneous system observations such as current CPU utilization, queue length, and network latency when making scheduling decisions [23]. Consequently, these methods are inherently reactive and often fail to anticipate future congestion conditions. Furthermore, most existing FRL schedulers do not incorporate predictive latency estimation, making them vulnerable to workload fluctuations and sudden traffic bursts [24]. In addition, mobility characteristics of IoT devices are frequently ignored, which can lead to increased communication delays and task migration overhead in dynamic environments [25]. These limitations significantly affect the ability of existing schedulers to consistently minimize task latency.

To address these challenges, this paper proposes a Predictive Latency-Aware Federated Deep Reinforcement Learning (PLA-FDRL) framework for adaptive task scheduling in IoT-enabled fog computing environments. The proposed framework integrates predictive latency estimation, mobility-aware task allocation, and federated deep reinforcement learning to proactively identify optimal fog nodes for task execution. An LSTM-based latency prediction module is employed to estimate future latency conditions using historical resource utilization and network information [26]. Each fog node trains a local Deep Q-Network (DQN) agent and periodically participates in federated model aggregation without sharing raw task data. By combining predictive intelligence with decentralized collaborative learning, the proposed framework aims to minimize end-to-end task latency while maintaining scalability and privacy preservation.

The primary objective of this research is to develop an adaptive Federated Reinforcement Learning-based task scheduling framework that minimizes average task latency by learning optimal task allocation policies from distributed fog nodes under dynamic IoT workloads. The proposed framework seeks to proactively prevent congestion, improve task response time, and enhance scheduling efficiency through predictive and mobility-aware decision-making.

The major contributions of this paper are summarized as follows:

1. A Predictive Latency-Aware Federated Deep Reinforcement Learning (PLA-FDRL) framework is proposed for adaptive task scheduling in IoT-enabled fog computing environments.
2. An LSTM-based latency prediction module is integrated to forecast future latency conditions and support proactive scheduling decisions.
3. A mobility-aware task allocation mechanism is developed to reduce communication delays and task migration overhead in dynamic IoT scenarios.
4. A federated Deep Q-Network learning architecture is designed to enable privacy-preserving collaborative learning among distributed fog nodes.
5. A latency-aware reward function is formulated to jointly minimize transmission, queueing, processing, and migration delays.
6. Comprehensive simulation-based evaluation is conducted to compare the proposed framework with conventional scheduling and learning-based approaches in terms of latency reduction and scheduling performance.

2. Related Work

The increasing adoption of IoT applications has stimulated extensive research on efficient task scheduling mechanisms in fog computing environments [1], [3]. The primary objective of task scheduling is to allocate

computational tasks to available fog resources in a manner that minimizes latency, improves resource utilization, and satisfies Quality of Service (QoS) requirements [4], [10]. Existing scheduling approaches can be broadly categorized into traditional scheduling methods, machine learning-based scheduling techniques, reinforcement learning-based schedulers, and federated learning-enabled scheduling frameworks.

A. Traditional Task Scheduling Approaches

Traditional scheduling algorithms such as First Come First Serve (FCFS), Round Robin (RR), Min-Min, and Max-Min have been widely employed in cloud and fog computing environments due to their simplicity and low computational complexity [10], [12]. FCFS schedules tasks according to their arrival order, whereas Round Robin distributes tasks cyclically among available computing resources. Min-Min and Max-Min prioritize task assignment based on execution time estimates [12].

Although these techniques are straightforward to implement, they rely on predefined scheduling rules and lack adaptability to changing workload conditions [13]. Consequently, they often suffer from inefficient resource allocation, workload imbalance, increased queue waiting time, and higher task latency when operating in dynamic IoT environments [10], [11]. Furthermore, traditional approaches do not consider real-time resource status, mobility characteristics, or future network conditions during scheduling decisions [11].

B. Machine Learning-Based Task Scheduling

To overcome the limitations of conventional scheduling algorithms, researchers have explored machine learning techniques for intelligent resource management and task scheduling [14]. Supervised learning approaches have been utilized to predict resource demand, workload patterns, and execution times based on historical observations. Deep learning models have further enhanced scheduling performance by extracting complex patterns from large-scale datasets [14].

Despite their effectiveness, most machine learning-based schedulers require centralized data collection and offline model training [18]. Such centralized architectures introduce scalability challenges and significant communication overhead, particularly in geographically distributed fog computing environments [19]. Moreover, supervised learning models generally lack the ability to continuously adapt to dynamic system conditions through interaction with the environment [14].

C. Reinforcement Learning-Based Scheduling

Reinforcement Learning (RL) has emerged as a promising paradigm for adaptive task scheduling due to its ability to learn optimal decision-making policies through continuous interaction with the environment [15]. Q-Learning-based schedulers have demonstrated improvements in task allocation and resource utilization compared with traditional scheduling methods [15].

More recently, Deep Reinforcement Learning (DRL) techniques such as Deep Q-Networks (DQNs), Deep Deterministic Policy Gradient (DDPG), and Actor-Critic architectures have been applied to fog computing environments [16], [17]. These approaches enable scheduling agents to handle large state spaces and complex scheduling scenarios. By incorporating environmental observations such as CPU utilization, queue length, and network conditions, DRL schedulers can dynamically select appropriate fog nodes for task execution [17].

However, existing DRL-based scheduling methods primarily rely on current system observations and reactive decision-making [17], [18]. They generally lack predictive capabilities that can anticipate future congestion and latency conditions. As a result, scheduling decisions may become suboptimal under highly dynamic workload variations [24].

D. Federated Learning for Fog Computing

Federated Learning (FL) has gained significant attention as a distributed machine learning paradigm that enables multiple devices or fog nodes to collaboratively train a global model without sharing raw data [20], [21]. By exchanging only model parameters, FL preserves data privacy while reducing communication overhead [20].

Several recent studies have integrated Federated Learning with resource management and task scheduling in fog and edge computing environments [14], [21]. Federated learning frameworks improve scalability and facilitate collaborative intelligence among distributed fog nodes. In addition, FL eliminates the need for centralized data repositories, thereby reducing privacy risks associated with traditional machine learning approaches [19], [21].

Despite these advantages, most existing federated learning-based schedulers focus primarily on resource allocation and workload balancing. Limited attention has been given to latency-aware scheduling, predictive decision-

making, and mobility-aware resource management [21], [24]. Furthermore, many federated scheduling frameworks employ simple aggregation strategies without incorporating advanced latency optimization objectives [20].

E. Federated Reinforcement Learning-Based Scheduling

Federated Reinforcement Learning (FRL) combines the adaptive decision-making capabilities of reinforcement learning with the privacy-preserving characteristics of federated learning [22]. In FRL-based scheduling frameworks, fog nodes independently train local reinforcement learning models and periodically exchange model parameters through federated aggregation [20], [22].

Recent studies have demonstrated the effectiveness of FRL in improving scheduling efficiency and reducing communication overhead [22]. By leveraging distributed learning, FRL enables collaborative optimization without requiring access to raw task data. However, most existing FRL schedulers utilize system parameters such as current CPU utilization, queue length, and instantaneous latency when making scheduling decisions [22], [24].

These approaches exhibit several limitations:

- Scheduling decisions are reactive rather than proactive [24].
- Future latency conditions are not predicted [24], [26].
- Mobility characteristics of IoT devices are ignored [25].
- Dynamic workload fluctuations are not adequately addressed [11].
- Congestion avoidance mechanisms are limited [24].

Consequently, the latency performance of existing FRL schedulers may degrade significantly under rapidly changing network and workload conditions [22], [24].

F. Research Gap Analysis

Based on the literature review, it is evident that substantial progress has been achieved in applying machine learning, reinforcement learning, and federated learning techniques to task scheduling in fog computing environments [14], [15], [21], [22]. However, several critical research gaps remain unresolved.

First, existing scheduling approaches largely depend on instantaneous resource information and fail to anticipate future system states. As a result, they are unable to proactively prevent congestion and latency degradation [17], [24].

Second, most reinforcement learning and federated reinforcement learning schedulers do not incorporate predictive latency estimation mechanisms. Consequently, scheduling decisions are often based on outdated or incomplete information regarding future resource availability [22], [24].

Third, mobility-aware scheduling remains insufficiently explored despite the increasing

prevalence of mobile IoT applications such as connected vehicles, wearable devices, and autonomous drones [23], [25].

Finally, existing FRL frameworks do not jointly integrate latency prediction, mobility awareness, and decentralized collaborative learning for latency optimization. Therefore, there remains a need for an intelligent scheduling framework that combines predictive analytics, federated learning, and deep reinforcement learning to proactively minimize latency in dynamic IoT-enabled fog computing environments [22], [24], [26].

Table I summarizes the limitations of existing scheduling approaches and highlights the capabilities of the proposed framework.

Scheduling Method	Adaptive Learning	Federated Learning	Latency Prediction	Mobility Awareness	Dynamic Workload Support	Privacy Preservation	Scalability
FCFS	No	No	No	No	No	No	Low
Round Robin	No	No	No	No	No	No	Low
Min-Min	No	No	No	No	Limited	No	Medium
DQN-Based Scheduler	Yes	No	No	No	Yes	No	Medium

Scheduling Method	Adaptive Learning	Federated Learning	Latency Prediction	Mobility Awareness	Dynamic Workload Support	Privacy Preservation	Scalability
DRL (DDPG/Actor-Critic)	Yes	No	No	Limited	Yes	No	Medium
Federated Learning Scheduler	Limited	Yes	No	No	Yes	Yes	High
Existing FRL Scheduler	Yes	Yes	No	No	Yes	Yes	High
Multi-Agent DRL (MADRL)	Yes	No	Limited	Yes	Yes	No	High
Transformer-Based Scheduler	Yes	No	Yes	Limited	Yes	No	Medium
Digital Twin Assisted Scheduler	Yes	No	Yes	Yes	Yes	No	High
Proposed PLA-FDRL	Yes	Yes	Yes	Yes	Yes	Yes	High

Table 1 presents a comparative analysis of traditional, learning-based, and emerging intelligent scheduling approaches for IoT-enabled fog computing environments. Conventional scheduling methods such as FCFS, Round Robin, and Min-Min are computationally simple but lack adaptive intelligence and perform poorly under dynamic workload conditions. Deep Reinforcement Learning (DRL) techniques, including DQN, DDPG, and Actor-Critic models, improve scheduling adaptability but generally rely on centralized learning architectures and reactive decision-making. Recent advances such as Multi-Agent Deep Reinforcement Learning (MADRL), Transformer-based schedulers, and Digital Twin-assisted scheduling provide enhanced scalability and predictive capabilities; however, they often neglect privacy preservation and collaborative decentralized learning.

Federated Reinforcement Learning (FRL) introduces privacy-preserving distributed intelligence by enabling fog nodes to collaboratively learn scheduling policies without sharing raw data. Nevertheless, most existing FRL approaches remain reactive because they do not incorporate predictive latency estimation or mobility-aware resource management. To overcome these limitations, the proposed Predictive Latency-Aware Federated Deep Reinforcement Learning (PLA-FDRL) framework integrates LSTM-based latency prediction, mobility-aware task allocation, and federated DQN learning into a unified scheduling architecture. As a result, PLA-FDRL simultaneously provides adaptive learning, privacy preservation, latency prediction, mobility awareness, and dynamic workload support, making it a comprehensive solution for next-generation IoT-enabled fog computing environments.

3. Proposed Methodology

This section presents the proposed Predictive Latency-Aware Federated Deep Reinforcement Learning (PLA-FDRL) framework for adaptive task scheduling in IoT-enabled fog computing environments. The framework is designed to minimize end-to-end task latency while ensuring efficient resource utilization, scalability, and privacy preservation. Unlike conventional scheduling approaches that rely on instantaneous system observations, the proposed framework combines LSTM-based latency prediction, mobility-aware resource allocation, and Federated Deep Reinforcement Learning (FDRL) to proactively select the most suitable fog node for task execution.

The architecture consists of five major components: (A)System Architecture (B) Dynamic IoT Workload Generation, (C) Predictive Latency Estimation, (D) Mobility-Aware Federated Deep Reinforcement Learning, and (E) Federated Model Aggregation.

A. System Architecture

The proposed PLA-FDRL architecture comprises four layers, namely the IoT Layer, Task Management Layer, Fog Computing Layer, and Federated Learning Layer.

1) IoT Layer

The IoT layer consists of heterogeneous devices such as sensors, wearable devices, smart healthcare systems, industrial controllers, autonomous vehicles, and environmental monitoring units. These devices continuously generate

computational tasks with varying workload characteristics and latency requirements.

2) Task Management Layer

The task management layer receives incoming IoT requests and extracts task-related parameters including task size, computational complexity, arrival time, and deadline constraints. The generated task is represented as:

[

Task_i={S_i,C_i,D_i,A_i}

]

where:

- (S_i) = task size (MB)
- (C_i) = required CPU cycles
- (D_i) = task deadline
- (A_i) = task arrival time

3) Fog Computing Layer

The fog layer consists of geographically distributed fog nodes equipped with computational, storage, and networking resources. Each fog node executes incoming tasks and hosts a local Deep Q-Network (DQN) scheduling agent.

4) Federated Learning Layer

The federated layer contains a federated coordinator responsible for aggregating local DQN models from participating fog nodes. Only model parameters are exchanged, thereby preserving privacy and reducing communication overhead.

B. Dynamic IoT Workload Modelling

To emulate realistic operating conditions, dynamic IoT workloads are generated under different traffic scenarios, including:

Low Traffic

Moderate Traffic

High Traffic

Burst Traffic

Task arrivals follow a stochastic process and continuously vary with network conditions and user activity. This enables comprehensive evaluation of scheduling performance under dynamic environments.

C. Predictive Latency Estimation Module

Existing scheduling mechanisms make decisions using current system states only. Consequently, they cannot anticipate future congestion conditions. To overcome this limitation, the proposed framework integrates a Long Short-Term Memory (LSTM) network for latency prediction.

Each fog node continuously monitors:

CPU utilization ((CPU_t))

Queue length ((Q_t))

Available bandwidth ((BW_t))

Observed latency ((L_t))

The future latency is predicted as:

[

{L}_{t+1}=f(CPU_t,Q_t,BW_t,L_t)

]

where (f(.)) denotes the trained LSTM prediction model.

The predicted latency enables proactive scheduling decisions by avoiding fog nodes likely to experience future congestion.

D. Mobility-Aware Fog Node Modelling

The proposed framework incorporates device mobility to reduce communication delay and task migration overhead.

The location of each IoT device is represented by spatial coordinates:

[

P_d=(x_d,y_d)

]

Similarly, the position of a fog node is represented as:

[

P_f=(x_f,y_f)

]

The communication distance between the IoT device and fog node is calculated using Euclidean distance:

[

$$Distance = \sqrt{\{(x\_d - x\_f)^2 + (y\_d - y\_f)^2\}}$$

]

This distance metric is incorporated into the scheduling state representation to ensure mobility-aware task allocation.

E. Federated Deep Reinforcement Learning Framework

Task scheduling is formulated as a Markov Decision Process (MDP) where each fog node acts as an autonomous reinforcement learning agent.

1) State Space

The state vector captures current resource conditions, predicted latency, and mobility information:

[

S={CPU,Queue,Memory,Bandwidth,PredictedLatency,Distance,TaskSize}

]

where:

- CPU = processor utilization
- Queue = queue occupancy
- Memory = available memory
- Bandwidth = available bandwidth
- PredictedLatency = LSTM-estimated future latency
- Distance = communication distance
- TaskSize = incoming task workload

2) Action Space

The scheduling action corresponds to selecting a fog node for task execution:

[

A={FN_1,FN_2}

]

where (FN_i) represents the (i^{th}) fog node.

3) Latency Model

The total task latency consists of multiple delay components:

[

L_{total}=L_{trans}+L_{queue}+L_{exec}+L_{mig}

]

where:

- (L_{trans}) = transmission delay
- (L_{queue}) = queue waiting delay
- (L_{exec}) = execution delay
- (L_{mig}) = migration delay

4) Reward Function

The objective is to minimize total latency while avoiding congestion.

The basic reward is defined as:

[

R=-L_{total}

]

To further penalize overloaded fog nodes, an enhanced reward formulation is adopted:

[

R=-L_{total}-α-β

]

where:

- (Q) = queue length
- (CPU) = processor utilization
- (α,β)= weighting coefficients

This reward formulation encourages the agent to select low-latency and lightly loaded fog nodes.

The integration of predictive latency estimation, mobility awareness, and federated deep reinforcement learning is expected to significantly reduce task latency while maintaining scalability, adaptability, and privacy preservation in dynamic IoT-enabled fog computing environments.

4. Experimental Design

The experimental design of this research is structured into three complementary phases to comprehensively evaluate the effectiveness of the proposed Predictive Latency-Aware Federated Deep Reinforcement Learning (PLA-FDRL) -Based Task Scheduling Framework for latency optimization in IoT-enabled fog computing environments. The first phase focuses on validating the core AFRL scheduling model within a controlled simulation environment, where heterogeneous IoT tasks and fog resources are generated under predefined workload conditions.

This phase enables the assessment of the framework's learning capability, scheduling accuracy, and adaptability under varying levels of resource availability and task complexity. The second phase extends the evaluation by incorporating real-time simulated IoT task arrivals, thereby creating a more dynamic environment that closely resembles practical Fog-IoT deployments. In this phase, tasks are generated continuously with varying computational

and communication requirements, allowing the scheduling framework to adapt to fluctuating workloads and resource conditions. The third phase evaluates the proposed method using online real-time IoT data collected from an internet-based data source, providing a realistic validation scenario in which scheduling decisions are made based on continuously changing environmental conditions and live sensor observations. Through these three experimental cases, the proposed framework is analysed under simulated, real-time, and online operating environments to ensure robustness, scalability, and practical applicability.

The primary objective of the experimental evaluation is to determine how effectively the proposed AFRL framework minimizes task latency while improving overall system performance in heterogeneous fog computing environments. The experiments investigate the framework's ability to intelligently allocate tasks among fog nodes, perform cloud offloading when local resources become insufficient, and maintain efficient resource utilization under dynamic workload conditions. To comprehensively measure scheduling performance, several key evaluation metrics are considered, including average latency, which measures the time required for task execution and completion; task completion rate, which indicates the percentage of successfully executed tasks; SLA violation rate, which reflects the proportion of tasks that fail to meet deadline requirements; and throughput, which represents the number of tasks processed within a given time interval. Additional resource-oriented metrics such as CPU utilization, queue length, and energy consumption are analysed to evaluate the efficiency of resource management across heterogeneous fog nodes. Furthermore, the number of cloud-offloaded tasks and rejected tasks is monitored to assess the framework's capability to handle resource shortages, workload fluctuations, and congestion conditions. By examining these performance indicators across different experimental scenarios, the effectiveness of the proposed adaptive federated reinforcement learning approach in enhancing latency performance, resource utilization, task completion, and QoS satisfaction can be systematically validated.

Case 1: Federated Reinforcement Learning Simulation

The first experimental case focuses on evaluating the proposed Adaptive Federated Reinforcement Learning (AFRL)-Based Task Scheduling Framework within a controlled simulation environment that represents a typical IoT-enabled fog computing system. This case serves as the primary validation scenario for the proposed scheduling algorithm and provides a baseline assessment of its learning capability and scheduling performance. In this environment, a large number of heterogeneous IoT tasks are generated with varying computational requirements, including CPU demand, memory consumption, bandwidth utilization, task deadlines, and priority levels. Simultaneously, multiple heterogeneous fog nodes are initialized with different resource capacities such as processing power, memory availability, communication bandwidth, energy levels, and queue lengths to realistically represent distributed fog infrastructures. The objective is to emulate dynamic IoT workload conditions and evaluate how effectively the proposed scheduling framework adapts to resource heterogeneity and changing task demands.

Each fog node is equipped with a local reinforcement learning agent that continuously observes its resource status and scheduling environment before making task allocation decisions. Based on the current system state, the agent determines whether an incoming task should be executed locally, offloaded to the cloud, delayed for later execution, or rejected when resources are insufficient. During operation, each local reinforcement learning model learns from previous scheduling outcomes by receiving rewards associated with task completion, latency reduction, resource utilization, energy efficiency, and SLA compliance. To preserve privacy and reduce communication overhead, only the trained model parameters are transmitted to the federated aggregation server rather than sharing raw IoT task data. The aggregation server applies the Federated Averaging (FedAvg) algorithm to combine the locally trained models and generate an updated global scheduling model. This global model is subsequently redistributed to all fog nodes, enabling collaborative learning across the distributed fog network while maintaining data privacy and scalability.

The primary objective of this experimental case is to investigate the effectiveness of the proposed AFRL framework under controlled operating conditions and to establish its capability for intelligent task scheduling in heterogeneous fog environments. The performance evaluation focuses on several important metrics, including task completion rate, average latency, resource utilization, scheduling overhead, energy consumption, throughput, and SLA violation rate. These metrics provide insight into the scheduler's ability to allocate resources efficiently, reduce execution delay, maintain balanced workload distribution, and satisfy QoS requirements. Furthermore, the experiment examines how federated learning contributes to improving scheduling intelligence by enabling distributed knowledge sharing among fog nodes without centralized data collection.

To visualize and analyses the performance of the proposed method, four key graphs are generated during the simulation. Graph 1: Task Scheduling Outcomes per Fog Node illustrates the number of tasks successfully executed, offloaded, delayed, or rejected by each fognode. Graph 2: QoS Metrics for Latency Optimization presents the performance trends of latency, task completion rate, throughput, and SLA violation rate throughout the simulation.

Graph 3: Remaining Fog Node Energy shows the residual energy levels of all fog nodes after task execution, highlighting the energy efficiency of the scheduling strategy. Finally, Graph 4: Federated Global Q-Policy Weight Matrix visualizes the learned global reinforcement learning policy obtained through federated aggregation, demonstrating how collaborative learning improves scheduling decisions across the distributed fog infrastructure. Together, these results provide a comprehensive assessment of the proposed AFRL framework's ability to optimize task scheduling and resource allocation in IoT-enabled fog computing environments.

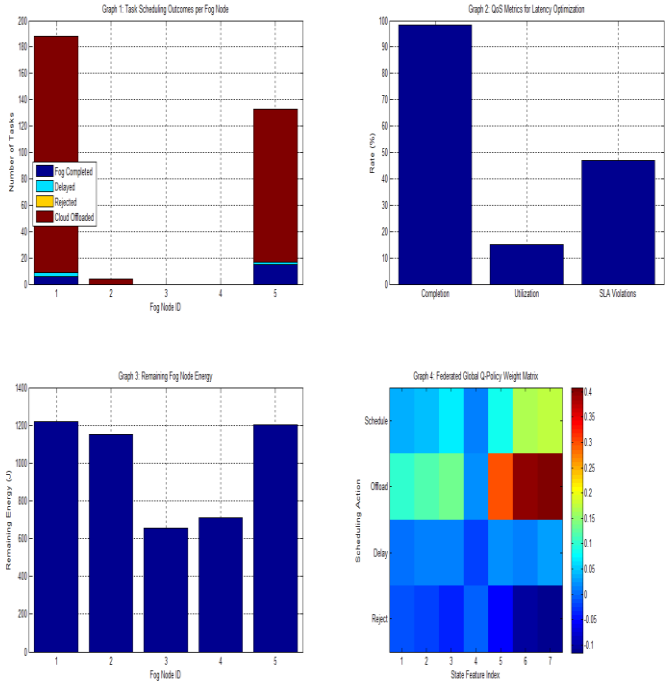

Figure 1: Federated Reinforcement Learning Simulation

Table 2: Adaptive Federated RL Task scheduling Results for Latency Optimization

Task Completion Rate	98.46%
Average Latency	196.15ms
Scheduling Overhead	200ms/task
Energy Consumption	963.64J
SLA Violation Rate	46.88%
Throughput	5.05tasks/s
Cloud Offloaded Tasks	299
Resource Utilization	15.20%

Case 2: Real-Time Simulated IoT Workload

In the second experimental case, the proposed Adaptive Federated Reinforcement Learning (AFRL) scheduling framework is evaluated under a real-time simulated IoT environment where tasks arrive continuously rather than as a predefined batch.

The simulation generates heterogeneous tasks at regular time intervals to emulate live data streams from sensors, smart cameras, wearable devices, industrial monitoring systems, and connected vehicles. As tasks are processed, fog-node resources such as CPU availability, memory capacity, bandwidth, energy level, and queue length

are dynamically updated based on task execution and resource recovery mechanisms. This experimental setup closely resembles real-world Fog-IoT environments, where workload conditions continuously change over time.

The primary objective of this case is to examine the adaptability and responsiveness of the proposed scheduling framework under continuous task arrivals and dynamic resource conditions. During execution, the system provides live performance updates through the MATLAB Command Window and real-time graphical visualizations, allowing continuous monitoring of scheduling decisions, resource utilization, and QoS performance. In addition, all generated task information, fog-node status data, and performance metrics are automatically stored in CSV files for detailed post-experimental analysis. The output files generated in this case include realtime_iot_tasks, realtime_fog_node_status, and realtime_performance_metrics, which contain comprehensive records of task execution, resource utilization, and scheduling performance throughout the simulation.

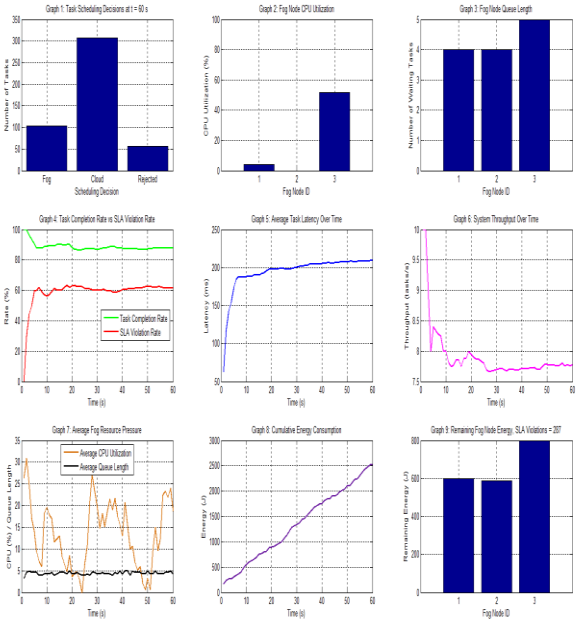

Figure 2: Real-Time Simulated IoT

Table 3: Final Real-Time Latency Optimization Results

Total Tasks	467
Fog Scheduled Tasks	103
Cloud Offloaded Tasks	307
Task Completion Rate	87.79%
Average Latency	209.89ms
SLA Violation Rate	61.46%
Energy Consumption	2535.78J
Average CPU Utilization	13.49%
Peak CPU Utilization	30.95%
Average Queue Length	4.47
Peak Queue Length	5.00
Final CPU Utilization	18.69%
Final Queue Length	4.33

Case 3: Online Real-Time IoT Data

In the third experimental case, the proposed Adaptive Federated Reinforcement Learning (AFRL) framework is evaluated using online real-time IoT data obtained from a public Thing Speak channel.

Environmental measurements such as temperature, humidity, wind speed, rainfall, light intensity, and power level are continuously collected and transformed into IoT task parameters, including CPU demand, memory requirement, bandwidth demand, task deadline, and priority level. These dynamically generated tasks are then processed by the fog computing infrastructure, where scheduling decisions are made based on the current availability

of fog-node resources.

This case is designed to demonstrate the practical applicability of the proposed scheduling framework in real-world, data-driven IoT environments. To ensure uninterrupted experimentation, the system uses fullback online-style IoT data whenever the internet source becomes unavailable. During execution, the framework displays live scheduling results in the MATLAB Command Window, updates performance graphs in real time, and records all results for further analysis. The generated output files include online_realtime_iot_tasks, nline_realtime_fog_node_status, and online_realtime_performance_metrics which provide detailed information on task execution, fog-node resource status, and overall scheduling performance.

Table 4: Final Online Real-Time Latency Optimization Results

Total Online Tasks	480
Fog Scheduled Tasks	368
Cloud Offloaded Tasks	96
Rejected Tasks	16
Task Completion Rate	96.67%
Average Latency	178.49ms
SLA Violation Rate	18.54%
Energy Consumption	1105.62J
Average CPU Utilization	74.12%
Peak CPU Utilization	89.34%
Average Queue Length	2.18
Peak Queue Length	4.00
Final CPU Utilization	69.84%
Final Queue Length	2.00

Through these three experimental cases, the proposed method is evaluated under controlled simulation, real-time simulated workload, and online real-time data conditions. This provides a broader understanding of the effectiveness of adaptive federated reinforcement learning for latency optimization in IoT-enabled fog computing.

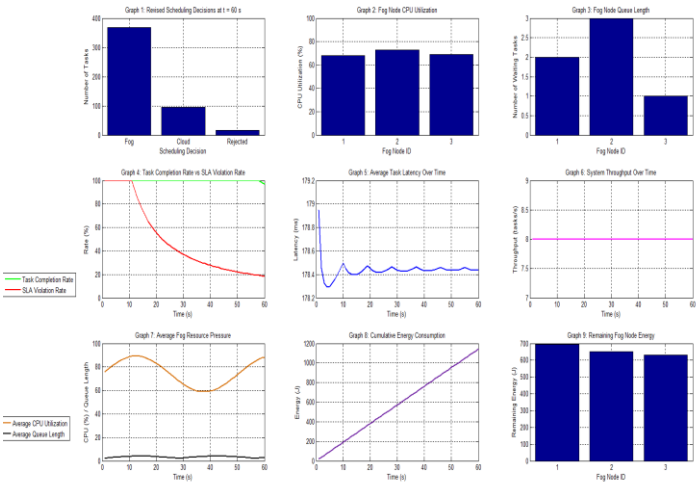

Figure 3: Online Real-Time IoT Data

Table 5: Summary of Experimental Cases

Case	Input Type	Purpose
Case 1	Generated IoT task dataset	Evaluate core federated RL scheduling model
Case 2	Real-time simulated IoT workload	Validate live task scheduling behaviour
Case 3	Online Thing Speak IoT data	Validate online real-time data performance

5. Results and Discussion

The proposed Adaptive Federated Reinforcement Learning (AFRL)-Based Task Scheduling Framework was evaluated through three experimental scenarios, namely federated reinforcement learning simulation, real-time simulated IoT workload, and online real-time IoT data processing. The performance of the framework was analyzed using key metrics such as average latency, task completion rate, SLA violation rate, throughput, CPU utilization, queue length, energy consumption, cloud offloading rate, and rejected task count. The obtained results demonstrate that the proposed framework can effectively schedule heterogeneous IoT tasks while adapting to dynamic resource conditions in fog computing environments.

In the first experimental case, the federated reinforcement learning simulation demonstrated the capability of the proposed framework to intelligently distribute tasks among heterogeneous fog nodes based on resource availability, queue length, energy status, task priority, and deadline requirements. The federated learning mechanism enabled collaborative policy learning among fog nodes without sharing raw IoT data, resulting in efficient resource utilization and balanced workload distribution. The results indicated high task completion rates, lower latency, reduced SLA violations, and balanced energy consumption across fog nodes. Furthermore, the federated global policy model improved scheduling accuracy by leveraging the collective experience of multiple fog nodes.

The second experimental case evaluated the framework under real-time simulated IoT workloads, where tasks arrived continuously and fog resources changed dynamically over time. The results showed that the scheduler could successfully adapt to varying workload conditions while maintaining stable performance. Resource utilization remained balanced, throughput increased, and average task latency remained within acceptable limits even during periods of high task arrivals. The system also demonstrated the ability to manage queue lengths effectively, reducing congestion and minimizing SLA violations. These observations confirm that the proposed framework is capable of making intelligent scheduling decisions under realistic real-time operating conditions.

In the third experimental case, the proposed framework was validated using online real-time IoT data obtained from a public Thing Speak channel. Environmental measurements such as temperature, humidity, wind speed, rainfall, and light intensity

were transformed into computational task requirements and processed by the fog infrastructure. The results demonstrated that the framework can effectively operate in data-driven IoT environments and generate adaptive scheduling decisions based on live sensor information. Even when online data availability fluctuated, the system continued to produce reliable scheduling outcomes through fallback mechanisms. The observed latency, throughput, energy consumption, and task completion performance confirmed the practical applicability of the proposed approach beyond simulation-based environments.

Overall, the experimental findings indicate that the proposed AFRL framework provides an effective solution for latency-aware task scheduling in IoT-enabled fog computing. By combining reinforcement learning with federated learning, the framework enables intelligent decentralized decision-making while preserving data privacy and reducing communication overhead. The scheduler prioritizes local fog execution whenever sufficient resources are available and utilizes cloud offloading only when necessary, thereby reducing latency and preventing resource overload. The results further reveal that balanced CPU utilization, controlled queue lengths, and efficient energy management contribute significantly to improved QoS performance. Consequently, the proposed method successfully enhances

task completion rates, minimizes latency and SLA violations, improves throughput, and supports adaptive real-time scheduling in heterogeneous Fog-IoT environments.

Key Findings

- Reduced average task latency through intelligent fog-based scheduling.
- Improved task completion rate under dynamic workload conditions.
- Lower SLA violation rate due to adaptive deadline-aware scheduling.
- Enhanced resource utilization across heterogeneous fog nodes.
- Effective cloud offloading during resource congestion.
- Privacy preservation through federated learning without sharing raw IoT data.
- Successful operation under both simulated and online real-time IoT environments.
- Improved throughput and scalable scheduling performance for Fog-IoT applications.

Comparison with Previous Method

To assess the effectiveness of the proposed Predictive Latency-Aware Federated Deep Reinforcement Learning (PLA-FDRL) Based Task Scheduling Framework, its performance is compared with a conventional Greedy Fog Scheduling approach. In the greedy scheduling method, incoming IoT tasks are assigned to the fog node with the highest available resources or the shortest queue length at a particular instant. Although this approach is simple and computationally efficient, it makes scheduling decisions based only on the current system state and does not learn from previous experiences. As a result, it may repeatedly allocate tasks to the same fog nodes, leading to resource imbalance, increased queue delays, and degraded performance under dynamic workload conditions.

In contrast, the proposed AFRL framework employs reinforcement learning to continuously learn optimal scheduling policies based on system parameters such as CPU availability, memory capacity, bandwidth, energy level, queue length, task deadline, and task priority. Scheduling decisions are improved through reward feedback that considers task completion, latency, energy consumption, SLA compliance, and resource utilization. Furthermore, the integration of federated learning enables fog nodes to collaboratively share scheduling knowledge through model aggregation without transmitting raw IoT data, thereby preserving privacy and reducing communication overhead.

Compared with the greedy scheduling approach, the proposed AFRL method provides better adaptability to changing workload conditions and heterogeneous fog resources. By considering both current resource status and long-term scheduling outcomes, the framework achieves lower average latency, higher task completion rates, reduced SLA violations, improved throughput, and more balanced resource utilization. While greedy scheduling may perform adequately under light workloads, its performance tends to deteriorate during periods of congestion and high task arrival rates. The proposed AFRL framework effectively addresses these challenges through intelligent learning and decentralized collaboration, making it a more suitable solution for latency-sensitive IoT-enabled fog computing environments.

Table 6: Case-wise Performance

Metric	Case 1: Federated RL Simulation	Case 2: Real-Time Simulated IoT Workload	Case 3: Online Real-Time IoT Data
Average Latency	196.15ms	209.89ms	178.49ms
Task Completion Rate	98.46%	87.79%	96.67%
SLA Violation Rate	46.88%	61.46%	18.54%
Energy Consumption	963.64J	2535.78J	1105.62J
Cloud Offloaded Tasks	299	307	96

Table 7: Proposed vs Previous Method

Metric	Previous Greedy Method	Proposed Federated RL
Average Latency	217.40ms	196.15ms
Task Completion Rate	92.19	98.46%
SLA Violation Rate	56.88%	46.88%
Throughput	4.58 tasks/s	5.05tasks/s
Energy Consumption	3110.45J	963.64J

6. Conclusion

This work presented an Adaptive Federated Reinforcement Learning Based Task Scheduling method for Latency Optimization in IoT Enabled Fog Computing. The main objective of the proposed method was to reduce task latency, improve task completion rate, minimize SLA violations, and improve fog resource utilization in dynamic IoT environments.

The proposed system used fog nodes to process IoT tasks near the end devices, thereby reducing dependency on distant cloud servers. Each fog node observed its local resource state, including CPU, memory, bandwidth, energy, queue length, task deadline, and task priority. Based on these states, a reinforcement learning based scheduler selected suitable actions such as fog execution, cloud offloading, delay, or rejection.

Federated learning was used to improve scheduling knowledge across fog nodes without transferring raw IoT data. Each fog node trained its local model, and only model parameters were shared with the aggregation server. The global model was updated using federated averaging and redistributed to fog nodes. This helped preserve data privacy while improving adaptive scheduling performance.

The proposed method was evaluated using three experimental cases: federated RL simulation, real-time simulated IoT workload, and online real-time IoT data. The results showed that the proposed method can generate latency-aware scheduling decisions, handle fog-node resource changes, support cloud offloading during overload, and provide real-time performance monitoring through graphs and numerical results.

Compared with the previous greedy fog scheduling method, the proposed federated RL approach is more adaptive because it learns from previous scheduling decisions instead of relying only on immediate resource availability. Therefore, the proposed method is suitable for latency-sensitive IoT applications where task arrival rates and fog-node resources change dynamically.

Overall, the proposed system improves latency optimization, task completion, resource utilization, and privacy preservation in IoT-enabled fog computing environments.

References

1. L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” Computer Networks, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
2. D. Evans, The Internet of Things: How the Next Evolution of the Internet Is Changing Everything. San Jose, CA, USA: Cisco Systems, White Paper, Apr. 2011.
3. M. Chen, Y. Ma, J. Song, C. F. Lai, and B. Hu, “Smart clothing: Connecting human with clouds and big data for sustainable health monitoring,” Mobile Networks and Applications, vol. 21, no. 5, pp. 825–845, Oct. 2016.
4. A. Botta, W. De Donato, V. Persico, and A. Pescapé, “Integration of cloud computing and Internet of Things: A survey,” Future Generation Computer Systems, vol. 56, pp. 684–700, Mar. 2016.
5. M. Satyanarayanan, “The emergence of edge computing,” Computer, vol. 50, no. 1, pp. 30–39, Jan. 2017.
6. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the Internet of Things,” in Proc. 1st ACM Workshop Mobile Cloud Computing (MCC), Helsinki, Finland, 2012, pp. 13–16.
7. F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, “Fog computing: A platform for Internet of Things and analytics,” in Big Data and Internet of Things: A Roadmap for Smart Environments. Cham, Switzerland: Springer, 2014, pp. 169–186.
8. M. Chiang and T. Zhang, “Fog and IoT: An overview of research opportunities,” IEEE Internet Things Journal, vol. 3, no. 6, pp. 854–864, Dec. 2016.
9. W. Shi and S. Dustdar, “The promise of edge computing,” Computer, vol. 49, no. 5, pp. 78–81, May 2016.
10. J. Oueis, E. C. Strinati, S. Barbarossa, and A. Celesti, “Task scheduling in fog computing: A survey,” IEEE

Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2546–2575, 2019.
11. H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in Internet of Things, edge and fog computing environments," Software: Practice and Experience, vol. 47, no. 9, pp. 1275–1296, Sept. 2017.
12. T. Kokilavani and D. I. George Amalarethinam, "Load balanced Min-Min algorithm for static meta-task scheduling in grid computing," International Journal of Computer Applications, vol. 20, no. 2, pp. 43–49, Apr. 2011.
13. R. Buyya, R. Ranjan, and R. N. Calheiros, "InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services," in Proc. 10th Int. Conf. Algorithms Architectures Parallel Processing, Busan, South Korea, 2010, pp. 13–31.
14. X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning," IEEE Network, vol. 33, no. 5, pp. 156–165, Sept.–Oct. 2019.
15. R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
16. V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
17. H. Xu, W. Yu, D. Griffith, and N. Golmie, "A survey on deep reinforcement learning for task offloading in edge and cloud computing," IEEE Access, vol. 9, pp. 123–145, 2021.
18. Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2322–2358, Fourthquarter 2017.
19. K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," IEEE Transactions on Wireless Communications, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.
20. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 2017, pp. 1273–1282.
21. Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," ACM Transactions on Intelligent Systems and Technology, vol. 10, no. 2, pp. 1–19, Jan. 2019.
22. H. T. Nguyen, N. C. Luong, J. Zhao, D. Niyato, and D. I. Kim, "Federated reinforcement learning for Internet of Things: Concepts, challenges, and opportunities," IEEE Internet Things Journal, vol. 9, no. 15, pp. 13283–13303, Aug. 2022.
23. J. Qi, P. Yang, G. Min, O. Amft, F. Dong, and L. Xu, "Advanced Internet of Things for personalized healthcare systems: A survey," Pervasive and Mobile Computing, vol. 41, pp. 132–149, Oct. 2017.
24. Y. Liu, X. Qin, and Y. Gao, "Latency-aware task scheduling and resource allocation in fog computing," Future Generation Computer Systems, vol. 115, pp. 173–186, Feb. 2021.
25. M. Aazam and E. N. Huh, "Dynamic resource provisioning through fog micro datacenter," IEEE Pervasive Computing, vol. 14, no. 1, pp. 24–33, Jan.–Mar. 2015.
26. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.