EAI Endorsed Transactions
on Scalable Information Systems
EAI.EU Research Article

# Wireless Federated Learning Based Building Temperature Estimation With Latency Constraint

Kemin Zhang[1,*]

[1]Guangzhou Donghua Vocation College, Guangzhou, China.

## Abstract

This paper proposes a novel approach for temperature estimation in buildings using wireless federated learning (FL) while considering latency constraint. The proposed model utilizes a hierarchical federated learning architecture within a wireless network, incorporating one base stations (BS), multiple access points (APs), and user equipment (UEs). In this sytem, each UE performs local learning and shares model updates with APs, which aggregate them and forward them to the BS for final aggregation. We design the system aiming to minimize both the latency and energy consumption while ensuring accurate temperature prediction. Simulation results show the effectiveness of the proposed scheme in comparison to the conventional deep reinforcement learning (DRL) and genetic algorithm (GA) approaches. Specifically, at the latency threshold of 10s, the proposed scheme achieves a prediction accuracy of approximately 0.60, while DRL reaches 0.50 and GA stays around 0.48. These results highlight the superior performance of the proposed federated learning-based method, especially in high-latency scenarios, and demonstrate its potential for real-time applications in smart building environments under wireless communication constraints.

Received on 11 April 2024; accepted on 27 September 2025; published on 16 October 2025

**Keywords:** Wireless federated learning, latency constraint, industrial IoT networks.



doi:10.4108/eetsis.9068

## 1. Introduction

Information technology (IT) has experienced exponential growth and transformation over the past few decades, profoundly impacting nearly every aspect of modern life, from communication to business operations, healthcare, education, and beyond [1–3]. Among the most significant advancements in IT, wireless technologies have played a pivotal role in shaping the future of connectivity. In this field, cutting-edge techniques have been proposed to enhance data transmission, network efficiency, and coverage. Specifically, 5G and beyond, along with advancements in Wi-Fi 6, can help boost the speed, capacity, and reliability of wireless networks, enabling faster download speeds, low-latency connections, and support for a large number of devices simultaneously [4–6]. The integration of technologies like millimeter-wave (mmWave), massive multiple-input multiple-output (MIMO), and beamforming has made it possible to support ultra-high data rates and provide seamless connectivity in dense urban environments. Moreover, wireless sensor networks (WSNs), Internet of Things (IoT), and low-power wide-area networks (LPWAN) are becoming integral to smart cities, industrial automation, and remote healthcare, facilitating real-time data collection and decision-making [7–9]. In addition, the advent of satellite-based communication systems, including low Earth orbit (LEO) satellite constellations, promises global internet access even in the most remote regions. In further, Terahertz communication has been investigated to significantly increase the transmission data rate in the future. As the demand for always-on connectivity continues to grow, the convergence of wireless technologies with machine learning, artificial intelligence, and edge computing leads to revolutionize how to interact with digital systems, making them faster, more intelligent, and adaptive to the evolving needs of users.

Mobile edge computing (MEC) has emerged as a transformative paradigm in the realm of mobile networks, particularly in addressing the critical issue of latency and energy consumption in modern wireless

*Corresponding author. Email: keminzhang.eecs@hotmail.com

systems [10, 11]. Specifically, MEC involves deploying computational resources at the edge of the network, closer to end-users, which reduces the dependency on centralized cloud infrastructures and enables faster data processing, real-time decision-making, and low-latency services [12–14]. This is especially crucial for applications such as augmented reality, autonomous vehicles, and industrial automation, where milliseconds of delay can significantly impact performance. It has been shown that reducing latency is one of the primary advantages of MEC, as it minimizes the need for long-distance data transmission to distant cloud servers [15]. Various strategies have been proposed for optimizing MEC networks to reduce the latency, such as dynamic resource allocation, task offloading mechanisms, and efficient scheduling techniques [16]. On the other hand, energy consumption is another critical concern in MEC, especially in mobile devices and edge nodes with limited battery life. Various methods have been proposed to optimize the energy efficiency, such as energy-efficient task offloading, load balancing, and the use of green communication protocols [17]. Energy efficiency is particularly important in MEC systems as a large number of devices should be supported in heterogeneous environments while minimizing the environmental impact. Moreover, trade-offs between latency and energy consumption are often explored, where reducing one may increase the other. Techniques such as joint optimization of latency and energy consumption through machine learning-based approaches, predictive analytics, and adaptive task offloading have been proposed to strike a balance between performance and energy savings.

Federated learning (FL) is an emerging machine learning paradigm that enables decentralized model training across multiple devices or edge nodes while keeping data localized, addressing concerns related to privacy and data security [18–20]. In FL, model parameters are updated by aggregating local updates from various devices, rather than centralizing raw data. The impact of training latency on the convergence rate of the global model of FL has been widely investigated, where the training latency refers to the time it takes for devices to complete local computations, transmit updates to the central server, and aggregate the results [21, 22]. It has been shown that higher training latency, often caused by network delays, device heterogeneity, and limited computational resources, can significantly hinder the convergence speed of the global model. Moreover, the convergence rate in federated learning is highly dependent on the synchronization of local updates and the frequency of communication between devices and the central server. As training latency increases, the model's ability to converge quickly diminishes, leading to prolonged training times and reduced overall system efficiency. Additionally, a high latency can result in stale or outdated model updates, further exacerbating the convergence problem and potentially leading to suboptimal performance [23, 24]. Various techniques have been proposed to mitigate the impact of training latency, such as adaptive aggregation methods, which prioritize recent updates, and decentralized optimization algorithms, which allow for local model updates to be more independent and less dependent on global synchronization. Other strategies include reducing the number of communication rounds, optimizing the local update frequency, and employing compression techniques to reduce the communication load [25, 26]. Moreover, hybrid federated learning models that combine centralized and decentralized training strategies have been proposed to balance the trade-off between the latency and convergence.

This paper introduces a novel approach for building temperature estimation through wireless federated learning, with a focus on addressing the latency constraint. In this framework, a hierarchical federated learning architecture is leveraged within a wireless network, involving one base stations (BS), multiple access points (APs), and user equipment (UEs). In this system, each UE conducts local model training and subsequently shares its updates with the associated AP, which aggregates them and then transmits the consolidated model to the BS for final aggregation. We design the system through minimizing both the latency and energy consumption, while ensuring the accuracy of temperature predictions. Simulation results demonstrate the effectiveness of the proposed method in comparison to the competing approaches, such as deep reinforcement learning (DRL) and genetic algorithm (GA). Notably, at the latency threshold of 10s, the proposed method achieves a prediction accuracy of approximately 0.60, outperforming DRL (0.50) and GA (0.48). These findings underscore the superior performance of the federated learning-based model, particularly in high-latency scenarios, highlighting its potential for real-time applications in smart building systems, where wireless communication constraints are a significant consideration.

## 2. System Model

In this paper, we consider a hierarchical federated learning (HFL) architecture for the building temperature estimation, implemented within a three-tier wireless network. The architecture consists of the key components of one BS and multiple APs, where the base stations are responsible for aggregating the local models received from the access points, and AP indicates the devices that act as intermediaries, responsible for collecting local models from multiple UEs and forwarding them to the BS. The UEs are the end-user devices

that generate local datasets, perform local learning, and upload their models to the APs for aggregation. Each UE is linked to a designated AP, establishing a dedicated connection through orthogonal frequency division multiple access (OFDMA). Each AP is assigned a set of UEs that it serves, denoted by $\mathcal{K}_i$, where $i \in \{1,\ldots,I\}$ represents the index of the AP, and the set $\mathcal{K} = \{1,\ldots,K\}$ represents all UEs in the network. The set of APs is represented by $\mathcal{I} = \{1,\ldots,I\}$, in which the BS aggregates the local models forwarded by APs. Each UE $k \in \mathcal{K}_i$ maintains a local dataset $D_{i,k} = \{(\mathbf{x}_{i,k,j}, y_{i,k,j}) \mid 1 \le j \le |D_{i,k}|\}$, where $\mathbf{x}_{i,k,j}$ denotes the feature vector and $y_{i,k,j}$ corresponds to the label of the $j$-th sample. The aggregated dataset size across all network devices is expressed as,

$$D = \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} D_{i,k}. \tag{1}$$

During each learning round, the BS disseminates the global model, which is generated by aggregating the local models uploaded by the UEs. After completing their local training, a selected group of UEs uploads their updated models to the APs, which then combine these updates and send the aggregated model to the BS. The global model for round $t$ is calculated by incorporating the models from the previous round,

$$w^{t+1} = \frac{\sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} a_{i,k}^t D_{i,k} w_i^t}{\sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} a_{i,k}^t D_{i,k}}, \tag{2}$$

where $w_i^t$ represents the local model of UE $k$ at AP $i$, and $a_{i,k}^t$ is a binary indicator variable that denotes whether UE $k$ at AP $i$ is selected to upload its model at round $t$. Specifically, $a_{i,k}^t = 1$ signifies that UE $k$ has been chosen to upload its model, while $a_{i,k}^t = 0$ indicates that it has not been selected.

In the communication and computation model, each AP in this hierarchical system allocates a set of $N$ available resource blocks (RBs), indexed by $\mathcal{N} = \{1,\ldots,N\}$, to support its associated UEs. These RBs are shared using the OFDMA mode, with each UE $k \in \mathcal{K}_i$ occupying a dedicated RB when selected. In cases of limited resources, the RBs are multiplexed across multiple APs. The RB allocation is indicated by $\delta_{i,k,n} \in \{0,1\}$, where $\delta_{i,k,n} = 1$ means RB $n$ is assigned to UE $k$, and $\delta_{i,k,n} = 0$ means it is not. The signal received by AP $i$ on RB $n$ from UE $k$ is,

$$\begin{aligned} y_{i,k,n} = {} & \delta_{i,k,n} h_i^t h_i, k, n \sqrt{p_{i,k,n}} \mathbf{x}_{i,k,n} + n_{i,k,n} \\ & + \sum_{i'\in\mathcal{I}\setminus i} \sum_{k'\in\mathcal{K}_i} \delta_{i,k,n} h_i^t h_i, k, n \sqrt{p_{i,k'}} \mathbf{x}_{i,k',n}. \end{aligned} \tag{3}$$

Here, $p_{i,k,n}$ denotes the transmit power of UE $k$ on resource block (RB) $n$, and $h_{i,k,n}^t$ represents the channel coefficient between AP $i$ and UE $k$ over RB $n$. Additionally, $n_{i,k,n}$ represents the additive white Gaussian noise term at the AP, and $p_{i,k',n}$ refers to the interference from other UEs.

We now proceed on the data rate and latency analysis. The uplink data rate $r_{i,k,n}^U$ for UE $k$ on RB $n$ is calculated using the following,

$$r_{i,k,n}^U = B^U \sum_{n=1}^{N} \log_2 \left(1 + \frac{\delta_{i,k,n} p_{i,k,n} |h_{i,k,n}|^2}{I_{i,k,n} + B^U N_0}\right), \tag{4}$$

where $B^U$ is the bandwidth of an RB, and a larger $B^U$ leads to an increased data rate $r_{i,k,n}^U$. Additionally, $I_{i,k,n}$ denotes the interference from other UEs and $N_0$ is the noise power.

Additionally, the computation latency $\tau_{C,i,k}^t$ for UE $k$ at AP $i$ is given by,

$$\tau_{C,i,k}^t = \frac{\kappa_{i,k} D_{i,k}}{\vartheta_{i,k}^t}, \tag{5}$$

where $\kappa_{i,k}$ represents the number of CPU cycles needed to process a single data sample, and $\vartheta_{i,k}^t$ denotes the CPU frequency of UE $k$ during the $t$-th round.

We now analyze the model upload latency and total latency in the HFL system. Specifically, the model upload latency for UE $k \in \mathcal{K}_i$ during the $t$-th round is expressed as,

$$\tau_{i,k}^U = \frac{Z(w_i^t)}{r_{i,k}^U}, \tag{6}$$

where $Z(w_i^t)$ represents the data size of the model to be uploaded, and $r_{i,k}^U$ denotes the uplink data rate of UE $k$ at AP $i$.

The overall latency in the $t$-th round of HFL is determined by the maximum of the latency among all APs and UEs, considering both model upload latency and computation latency. Therefore, the total latency $\tau$ for HFL in round $t$ is:

$$\tau^t = \max_{i\in\mathcal{I}, k\in\mathcal{K}_i} \left\{a_{i,k}^t \left(\tau_{i,k}^U + \tau_{i,k}^C\right)\right\}, \tag{7}$$

where $a_{i,k}^t$ denotes the selection indicator for UE $k$. To simplify the notation, the UE selection indicator $a_{i,k}^t$ is formally as,

$$a_{i,k}^t = \sum_{n=1}^{N} \delta_{i,k,n}, \quad \forall i \in \mathcal{I}, k \in \mathcal{K}_i. \tag{8}$$

This indicator ensures that if a UE is not selected, $a_{i,k}^t = 0$, or not otherwise.

The energy consumption required for computation by UE $k$ at AP $i$ is,

$$E_{C,i,k}^t = \xi_{i,k} \kappa_{i,k} \vartheta_{i,k}^2 D_{i,k}, \tag{9}$$

where $\xi_{i,k}$ represents the capacitance coefficient, and $\kappa_{i,k}$ is the CPU cycles required to process one sample. Additionally, the energy consumption for transmission by UE $k$ at AP $i$ is given by,

$$E_{i,k}^{U,t} = \left(\sum_{n=1}^{N} \delta_{i,k,n} p_{i,k,n}\right) r_{i,k}^{U} \tau_{i,k}^{U}. \tag{10}$$

The total energy consumption for UE $k$ at AP $i$ during the $t$-th round of HFL is,

$$E_{i,k}^{t} = E_{i,k}^{U,t} + E_{C,i,k}^{t}. \tag{11}$$

Thus, the total energy consumption $E_{i,k}^{t}$ incorporates both the transmission and computation energies.

For UE $k \in \mathcal{K}_i$, the overall energy consumption during the $t$-th round is the sum of transmission energy and computation energy. The total energy consumption of UE $k$ at AP $i$ in the $t$-th round is,

$$E_{i,k}^{t} = E_{i,k}^{U,t} + E_{C,i,k}^{t}. \tag{12}$$

This energy is pivotal for ensuring the efficient operation of the system while balancing resource consumption and model accuracy.

## 3. Convergence Analysis

We now perform the convergence analysis on the considered system, where the following assumptions are considered:

- Assumption 1 (Lipschitz continuous gradient): The global loss function $F(w)$ is Lipschitz continuous, with a positive Lipschitz constant $L$. Accordingly, for any two points $w$ and $v$, the gradient satisfies:

$$\|\nabla F(w) - \nabla F(v)\| \leq L\|w - v\|. \tag{13}$$

- Assumption 2 ($\mu$-strongly convex): The global loss function $F(w)$ is strongly convex, characterized by a positive modulus $\mu$. Consequently, for any two points $w$ and $v$, the following inequality holds,

$$\|\nabla F(w) - \nabla F(v)\| \geq \mu\|w - v\|. \tag{14}$$

- Assumption 3 (Bounded gradient): The gradient with respect to any given data sample is bounded by,

$$\|\nabla F(w; \mathbf{x}_i, \mathbf{y}_i)\|^2 \leq \beta_1 + \beta_2 \|F(w)\|^2, \tag{15}$$

where $\beta_1$ and $\beta_2$ are constants that limit the growth of the gradient.

Using the assumptions mentioned above, we can bound the expected gap between the optimal solution and the actual loss of HFL as,

$$E\left[F(w^{t+1}) - F(w^*)\right] \leq A^t E\left[F(w^t) - F(w^*)\right] + \frac{2\beta_1 B^t}{LD^2}, \tag{16}$$

in which $A^t$ and $B^t$ are,

$$A^t = 1 - \frac{\mu}{L} + \frac{A_2 \beta_2 B^t}{D^2}, \quad B^t = \left(\sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} \left(1 - a_{i,k}^t\right) D_{i,k}\right)^2. \tag{17}$$

This result shows that selecting more UEs to participate in the learning process leads to a faster convergence. However, excessive selection of UEs could lead to interference among UEs, which may degrade performance due to the possibly arising inter-cell interference in the OFDMA among multiple UEs. The above analysis aims to balance model accuracy and energy consumption while ensuring convergence and minimizing interference.

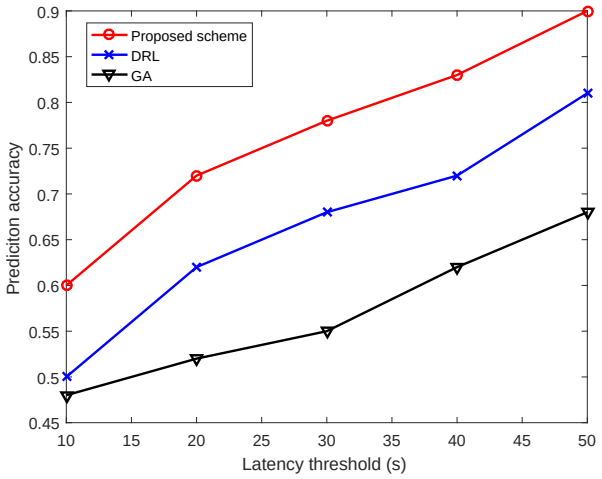


**Figure 1.** Prediction accuracy versus the training latency threshold with $K = 10$ and $N_D = 10$.

## 4. Simulation Results and Discussions

In this part, we evaluate the proposed hierarchical FL system in a three-tier wireless network (one BS, multiple APs, and UEs) under OFDMA uplink access, where each selected UE occupies a dedicated resource block and the uplink rate, computation latency, upload latency, and energy consumption. Specifically, the latency threshold varies from 10s to 50s, and the prediction accuracy is presented versus this threshold. To examine scalability and data richness, we test multiple operating points with the number of UEs $K \in \{10, 20\}$ and a data-size parameter $N_D \in \{10, 20\}$.

Fig. 1 and Table I show the relationship between the training latency threshold and prediction accuracy with

**Table 1.** Prediction accuracy of several schemes versus the latency threshold with $K = 10$ and $N_D = 10$.

| Latency Threshold (s) | Proposed Scheme | DRL | GA |
|---|---|---|---|
| 10 | 0.60 | 0.50 | 0.48 |
| 20 | 0.72 | 0.62 | 0.52 |
| 30 | 0.78 | 0.68 | 0.55 |
| 40 | 0.83 | 0.72 | 0.62 |
| 50 | 0.90 | 0.81 | 0.68 |

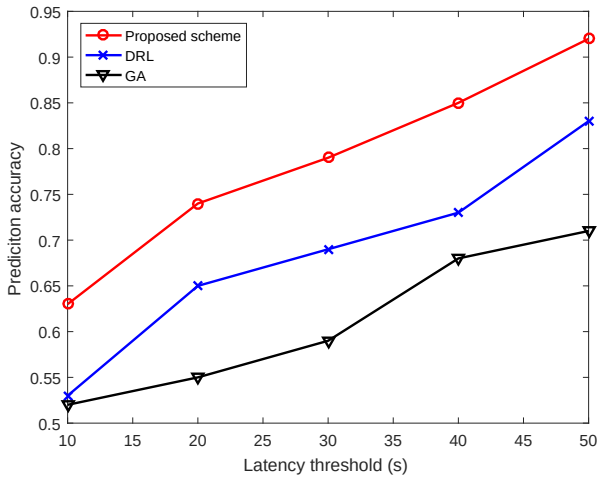


**Figure 2.** Prediction accuracy versus the training latency threshold with $K = 20$ and $N_D = 10$.

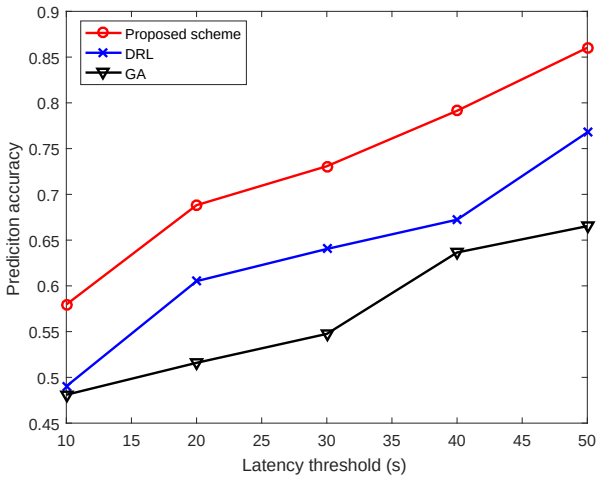


**Figure 3.** Prediction accuracy versus the training latency threshold with $K = 10$ and $N_D = 20$.

$K = 10$ and $N_D = 10$ for three different schemes: the proposed scheme, deep reinforcement learning (DRL), and genetic algorithm (GA). The latency threshold ranges from 10s to 50s, From this figure and table, we can see that the three schemes all show a clear trend where the prediction accuracy improves as the latency threshold increases. Specifically, the proposed scheme consistently outperforms both the DRL and GA schemes in terms of prediction accuracy across all latency thresholds. For example, at a latency threshold of 10 seconds, the proposed scheme achieves a prediction accuracy of about 0.6, while the DRL scheme reaches approximately 0.55, and the GA scheme is slightly lower at around 0.5. As the latency threshold increases to 50 seconds, the proposed scheme's accuracy climbs to almost 0.85, whereas DRL reaches around 0.75, and GA stays significantly lower, around 0.65. The results in this figure and table demonstrate the effectiveness of the proposed scheme, particularly as the latency threshold increases, showing that it provides a higher prediction accuracy than both DRL and GA, which shows more gradual improvement in performance.

Fig. 2 and Table II display the relationship between the training latency threshold and prediction accuracy of three different schemes, where $K = 10$, $N_D = 20$, and the latency threshold varies from 10s to 50s. From Fig. 2 and Table II, we can see that as the latency threshold increases, the prediction accuracy improves for all three schemes, where the proposed scheme consistently achieves the highest accuracy compared to DRL and GA across all latency thresholds. For instance, at a latency threshold of 10 seconds, the proposed scheme achieves the accuracy of about 0.6, DRL reaches approximately 0.55, and GA is slightly lower at around 0.5. At higher latency thresholds, such as 50 seconds, the proposed scheme achieves the prediction accuracy of nearly 0.95, while DRL reaches about 0.85 and GA stays at around 0.7. This comparison indicates that, although all schemes benefit from increased latency, the proposed scheme shows superior performance, particularly as the latency threshold rises. It exhibits the most significant improvement in the prediction accuracy over the other two schemes, especially as the latency threshold exceeds 30 seconds.

Fig. 3 and Table III illustrate the relationship between the training latency threshold and prediction accuracy for three schemes, where $K = 10$, $N_D = 20$, and the latency threshold ranges from 10 to 50 seconds. From this figure and table, we can find that as the latency threshold increases, all schemes demonstrate an improvement in the prediction accuracy, but the

**Table 2.** Prediction accuracy of several schemes versus the latency threshold with $K = 20$ and $N_D = 10$.

| Latency Threshold (s) | Proposed Scheme | DRL | GA |
|---|---|---|---|
| 10 | 0.63 | 0.53 | 0.52 |
| 20 | 0.74 | 0.65 | 0.55 |
| 30 | 0.79 | 0.69 | 0.59 |
| 40 | 0.85 | 0.73 | 0.68 |
| 50 | 0.92 | 0.83 | 0.71 |

**Table 3.** Prediction accuracy of several schemes versus the latency threshold with $K = 10$ and $N_D = 20$.

| Latency Threshold (s) | Proposed Scheme | DRL | GA |
|---|---|---|---|
| 10 | 0.5796 | 0.4903 | 0.4810 |
| 20 | 0.6882 | 0.6052 | 0.5159 |
| 30 | 0.7308 | 0.6403 | 0.5475 |
| 40 | 0.7914 | 0.6723 | 0.6365 |
| 50 | 0.8602 | 0.7678 | 0.6653 |

proposed scheme outperforms both DRL and GA at all latency thresholds. For example, at the latency threshold of 10 seconds, the proposed scheme achieves the prediction accuracy of around 0.6, DRL reaches approximately 0.55, and GA is slightly lower at 0.5. As the latency threshold increases to 50 seconds, the proposed scheme achieves the prediction accuracy of about 0.85, while DRL reaches around 0.75, and GA remains lower at approximately 0.65. This indicates that while increasing the latency threshold benefits all three schemes, the proposed scheme demonstrates the most significant improvement in the prediction performance, providing superior accuracy across the board. Notably, as the latency threshold rises, the proposed scheme shows a sharper increase in accuracy compared to DRL and GA, suggesting its higher effectiveness at managing the training process under longer latency conditions.

## 5. Conclusions

This paper presented an innovative solution for building temperature estimation using wireless federated learning, addressing the critical issue of latency constraint. The proposed approach employed a hierarchical federated learning architecture within a wireless network, consisting of one BS, multiple APs and UEs. Each UE performed local model training and transmitted its updates to the AP, which aggregated the information and forwarded it to the BS for global model consolidation. The system was designed to minimize both latency and energy consumption, ensuring an accurate temperature prediction in a resource-efficient environment. Simulation results validated the performance of the proposed method, demonstrating its superiority over the competing schemes, such as DRL and GA. Specifically, at the latency threshold of 10s, the proposed scheme achieved a prediction accuracy of approximately 0.60, outperforming DRL (0.50) and GA (0.48). These results highlighted the effectiveness of the federated learning-based approach, particularly in latency-sensitive environments, and demonstrated its potential for real-time applications in smart buildings, where efficient wireless communication was paramount.

## References


[1] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin *et al.*, "MEC in 5G networks," *ETSI white paper*, vol. 28, no. 28, pp. 1–28, 2018.

[2] S. Guo and X. Zhao, "Multi-agent deep reinforcement learning based transmission latency minimization for delay-sensitive cognitive satellite-uav networks," *IEEE Trans. Commun.*, vol. 71, no. 1, pp. 131–144, 2023.

[3] H. Hui and W. Chen, "Joint scheduling of proactive pushing and on-demand transmission over shared spectrum for profit maximization," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 107–121, 2023.

[4] L. F. Abanto-Leon, A. Asadi, A. Garcia-Saavedra, G. H. Sim, and M. Hollick, "Radiorchestra: Proactive management of millimeter-wave self-backhauled small cells via joint optimization of beamforming, user association, rate selection, and admission control," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 153–173, 2023.

[5] L. Yin and B. Clerckx, "Rate-splitting multiple access for satellite-terrestrial integrated networks: Benefits of coordination and cooperation," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 317–332, 2023.

[6] S. Arya and Y. H. Chung, "Fault-tolerant cooperative signal detection for petahertz short-range communication with continuous waveform wideband detectors," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 1, pp. 88–106, 2023.

[7] H. Yao, X. Li, and X. Yang, "Physics-aware learning-based vehicle trajectory prediction of congested traffic in a connected vehicle environment," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 102–112, 2023.
[8] Z. Li, J. Xie, W. Liu, H. Zhang, and H. Xiang, "Joint strategy of power and bandwidth allocation for multiple maneuvering target tracking in cognitive MIMO radar with collocated antennas," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 190–204, 2023.
[9] Y. Sun, D. Wu, X. S. Fang, and J. Ren, "On-glass grid structure and its application in highly-transparent antenna for internet of vehicles," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 93–101, 2023.
[10] J. Zhang, M. Wu, Z. Sun, and C. Zhou, "Learning from crowds using graph neural networks with attention mechanism," *IEEE Trans. Big Data*, vol. 11, no. 1, pp. 86–98, 2025.
[11] J. Du, J. Xu, A. Sun, J. Kang, Y. Hu, F. R. Yu, and V. C. M. Leung, "Profit maximization for multi-time-scale hierarchical DRL-based joint optimization in MEC-enabled air-ground integrated networks," *IEEE Trans. Commun.*, vol. 73, no. 3, pp. 1591–1606, 2025.
[12] M. Zhao, Z. Yang, Z. He, F. Xue, and X. Zhang, "Multi-round stackelberg game-based pricing and offloading in containerized MEC networks," *IEEE Trans. Green Commun. Netw.*, vol. 9, no. 1, pp. 191–206, 2025.
[13] S. Aggarwal, N. Kumar, N. Mohammad, and A. Barnawi, "Blockchain-based content retrieval mechanism in ndn-enabled V2G networks," *IEEE Trans. Ind. Informatics*, vol. 21, no. 1, pp. 118–127, 2025.
[14] Y. Xu, S. Li, K. Feng, B. Sun, X. Yang, L. Kou, Z. Zhao, and G. Q. Huang, "Multiperspective temporal pooling convolutional neural networks for fault diagnosis of mechanical transmission systems," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–12, 2025.
[15] X. Zhang, B. Yang, Z. Yu, X. Cao, G. C. Alexandropoulos, Y. Zhang, M. Debbah, and C. Yuen, "Reconfigurable intelligent computational surfaces for MEC-assisted autonomous driving networks: Design optimization and analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 1, pp. 1286–1303, 2025.
[16] Y. Gong, H. Yao, Z. Xiong, C. L. P. Chen, and D. Niyato, "Blockchain-aided digital twin offloading mechanism in space-air-ground networks," *IEEE Trans. Mob. Comput.*, vol. 24, no. 1, pp. 183–197, 2025.
[17] M. Zhang and D. Vasal, "Large-scale mechanism design for networks: Superimposability and dynamic implementation," *IEEE Trans. Mob. Comput.*, vol. 24, no. 3, pp. 1278–1292, 2025.
[18] C. Qiao, M. Li, Y. Liu, and Z. Tian, "Transitioning from federated learning to quantum federated learning in internet of things: A comprehensive survey," *IEEE Commun. Surv. Tutorials*, vol. 27, no. 1, pp. 509–545, 2025.
[19] S. Pejoski, M. Poposka, and Z. Hadzi-Velkov, "Optimized scheduling transmissions for wireless powered federated learning networks," *IEEE Commun. Lett.*, vol. 29, no. 3, pp. 640–644, 2025.
[20] M. Alishahi, P. Fortier, M. Zeng, T. Huynh-The, X. Li, and Q. Pham, "Latency minimization for STAR-RIS-aided federated learning networks with wireless power transfer," *IEEE Internet Things J.*, vol. 12, no. 7, pp. 8508–8522, 2025.
[21] X. Fang, M. Ye, and B. Du, "Robust asymmetric heterogeneous federated learning with corrupted clients," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 4, pp. 2693–2705, 2025.
[22] H. Zhang, C. Li, W. Dai, Z. Zheng, J. Zou, and H. Xiong, "Stabilizing and accelerating federated learning on heterogeneous data with partial client participation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 1, pp. 67–83, 2025.
[23] L. Chen, D. Zhao, L. Tao, K. Wang, S. Qiao, X. Zeng, and C. W. Tan, "A credible and fair federated learning framework based on blockchain," *IEEE Trans. Artif. Intell.*, vol. 6, no. 2, pp. 301–316, 2025.
[24] Z. Wang, N. Dong, J. Sun, W. Knottenbelt, and Y. Guo, "Zero-knowledge proof-based gradient aggregation for federated learning," *IEEE Trans. Big Data*, vol. 11, no. 2, pp. 447–460, 2025.
[25] X. Chen, T. Du, M. Wang, T. Gu, Y. Zhao, G. Kou, C. Xu, and D. O. Wu, "Towards optimal customized architecture for heterogeneous federated learning with contrastive cloud-edge model decoupling," *IEEE Trans. Computers*, vol. 74, no. 4, pp. 1123–1137, 2025.
[26] Y. Jia, Z. Huang, J. Yan, Y. Zhang, K. Luo, and W. Wen, "Joint optimization of resource allocation and data selection for fast and cost-efficient federated edge learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 11, no. 1, pp. 594–606.