Bringing Intelligence to Energy Consumption in Smart Buildings: Leveraging Fog Computing and Federated Learning

Jessica Al Achy[a], Hassan Harb[b], Abdallah Makhoul[a]
[a]*Université Marie et Louis Pasteur, CNRS, institut FEMTO-ST, F-25200 Montbéliard, France*
[b]*College of Engineering and Technology, American University of the Middle East, Kuwait*
Emails: jessica.al_achy@univ-fcomte.fr, hassan.harb@aum.edu.kw, abdallah.makhoul@univ-fcomte.fr

***Abstract*—Energy consumption prediction in smart buildings is essential for optimizing energy efficiency, reducing costs, and minimizing environmental impact. This paper proposes a novel approach combining federated learning with edge-fog-cloud computing to predict energy consumption across multiple smart buildings while preserving data privacy. The architecture involves training local models on edge devices, aggregating them at fog nodes, and performing higher-level aggregation in the cloud. Differential privacy is integrated into the federated learning process to ensure data confidentiality. The study utilizes two years of data collected from 12 houses, comprising 17,428 records, and employs LSTM neural networks for time-series prediction. The results demonstrate significant improvements in prediction accuracy (MSE: 14.55, MAE: 2.83, R^2: 0.93, Accuracy: 93.28%), along with reduced latency (1.97 seconds) and efficient execution time (5904.36 seconds). Comparative analysis explores the performance of different machine learning models in energy consumption prediction, highlighting the strengths and limitations of each approach. This work contributes to advancing sustainable energy management practices in smart building ecosystems.**

***Index Terms*—Smart Buildings, Internet of Things, Energy Consumption, Federated Learning, Edge/Fog Computing.**

I. Introduction

A. Context

Energy is fundamental to modern society, impacting individuals, institutions, and governments. Efficient energy management has become essential in addressing environmental and economic challenges. According to the European Commission [1], prioritizing energy efficiency enhances sustainability while improving access to reliable, affordable energy. In this context, Energy Management Systems (EMS) have emerged as vital tools, particularly in smart buildings, enabling optimized energy usage and promoting comfort, health, and environmental sustainability.

B. Motivation and Challenges

The integration of Internet of Things (IoT) and Artificial Intelligence (AI) into EMS has transformed smart buildings into adaptive, data-driven environments. IoT sensors enable real-time monitoring and control of energy systems, achieving up to 30-50% energy savings compared to traditional buildings [2, 3]. Meanwhile, AI enhances decision-making through predictive modeling and dynamic adaptation to user behaviors [4].

Despite these advancements, two critical challenges persist:

- **High Computational Demand:** Centralized AI-based EMS struggle to process vast IoT data streams in real time, leading to inefficiencies and latency.
- **Privacy Concerns:** Sharing sensitive user data, such as occupancy patterns, with centralized servers raises significant privacy risks.

To address these challenges, this research proposes integrating Edge/Fog Computing (EFC) and Federated Learning (FL) into the EMS architecture of smart buildings.

- **Edge/Fog Computing (EFC):** Decentralizes computation by processing data at or near its source, reducing latency and improving responsiveness. In smart buildings, EFC enhances real-time energy optimization by enabling localized decision-making with minimal dependence on centralized servers.
- **Federated Learning (FL):** Preserves data privacy by enabling collaborative model training across decentralized devices without sharing raw data. FL adapts to the heterogeneous and dynamic nature of energy consumption patterns while ensuring robust privacy protections.

Together, EFC and FL form a robust framework for efficient, privacy-conscious energy management in smart buildings.

C. Objective

This study aims to develop a scalable and privacy-preserving framework for energy consumption prediction in smart buildings. By integrating EFC and FL, the proposed system addresses computational and privacy challenges associated with centralized AI systems. The framework will be evaluated based on prediction accuracy, energy savings, scalability, and privacy preservation, offering a decentralized solution for modern EMS.

D. Contributions

The key contributions of this research are:

- **Multi-Level Federated Learning (FL):** A hierarchical FL architecture trains energy prediction models across

edge devices, fog nodes, and the cloud. This enhances scalability, reduces latency, and ensures data privacy.
- **Localized Energy Prediction with LSTM**: Time-series data is processed using LSTM models at the edge, enabling accurate predictions with minimal communication overhead.
- **Privacy with Differential Privacy (DP)**: DP ensures data privacy by adding Gaussian noise to model updates, safeguarding sensitive information even if servers are compromised.
- **Hierarchical Clustering:** A dynamic clustering method optimizes resource utilization and addresses data heterogeneity, improving training efficiency in large-scale deployments.
- **EFC-Cloud Architecture:** The integration of EFC supports real-time energy management, while cloud-level aggregation ensures stability and scalability.

II. Related Work

Energy consumption prediction in smart buildings has gained attention due to privacy concerns, high communication costs, and scalability challenges in centralized models [5, 6].

M. Umair et al. [7] propose a Markov-chain-based model integrated into a fog-based IoT architecture, achieving a 36% reduction in electricity usage. Coalition game theory has been explored for optimizing energy allocation while balancing generation and demand [8]. A Smart Home Energy Management System (SHEMS) combining Non-Intrusive Appliance Load Monitoring (NIALM) and ANN enhances residential energy management with fog-cloud analytics [9]. Federated learning (FL) has been adopted for load forecasting, ensuring data privacy. Studies compare FL strategies like FedSGD and FedAVG, demonstrating FedAVG's superior accuracy and communication efficiency [10]. Privacy-preserving FL approaches employ Min–Max scaling, LSTM, and Differential Privacy (DP) to secure residential energy data [11]. Hybrid FL frameworks integrating clustering, anomaly detection, and hyperparameter tuning enhance both prediction accuracy and privacy [12]. FL-based residential load forecasting has also been applied to large-scale datasets, optimizing customer demand management [13]. Additionally, ANN-based FL frameworks effectively address data silos in short-term energy forecasting [14].

Despite advancements, most studies lack comprehensive multi-level FL architectures integrating edge, fog, and cloud layers, and privacy mechanisms are often not optimized for resource-constrained environments. Our work addresses these gaps by proposing a multi-layer FL framework leveraging LSTM, edge-fog-cloud integration, and DP to improve energy consumption predictions in smart buildings.

III. Proposed Framework

This section presents a multi-tiered federated learning framework designed for the prediction of energy consumption in smart buildings. The architecture integrates EFC, FL, and DP to address critical challenges such as scalability, data privacy, real-time responsiveness, and computational efficiency. The framework, shown in Figure 1, includes five key layers: Perception, Network, Fog, Cloud, and Application. Each layer has a unique function that contributes to effective, privacy-preserving energy management.

A. Initial Data Collection and Clustering

At the perception layer, IoT devices, such as smart meters, sensors, and smart plugs, continuously collect minute-level data on key parameters, including energy consumption, temperature, humidity, time, etc. These edge devices perform lightweight preprocessing to ensure data quality and minimize transmission overhead. Preprocessing involves two key techniques: Min-Max Scaling and Principal Component Analysis (PCA).

The Min-Max Scaling technique ensures that the collected data is normalized to a standard range [0,1], as shown in equation (1):

$$x' = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{1}$$

where x is the original value, x_{min} and x_{max} are the minimum and maximum values of the feature, respectively, and x' is the scaled value. This normalization step ensures uniform feature distribution, reduces bias, and accelerates convergence during training.

Next, PCA is applied to reduce the dataset's dimensionality, extracting principal components that account for the most variance. The transformation is mathematically represented in the equation (2):

$$Z = XW \tag{2}$$

where X represents the standardized dataset, W is the matrix of principal components, and Z represents the transformed dataset. PCA reduces computational overhead while preserving essential information for model training.

To further enhance efficiency, K-means clustering is employed to group edge devices based on their energy consumption patterns. The clustering algorithm minimizes the intra-cluster variance using the following objective function (3):

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n} \|x_{ij} - c_i\|^2 \tag{3}$$

where x_{ij} represents the j^{th} data point in cluster i, and c_i is the centroid of cluster i. Devices within the same cluster exhibit similar energy consumption behaviors, enabling more targeted federated learning. These steps collectively ensure that only processed and optimized data is transmitted to the fog nodes, reducing both communication overhead and computational complexity.

B. Local Training within Each Cluster

Once edge devices are clustered, each device independently trains a Long Short-Term Memory (LSTM) model for time-series energy prediction. LSTMs are well-suited for capturing long-term dependencies in sequential data, making them ideal

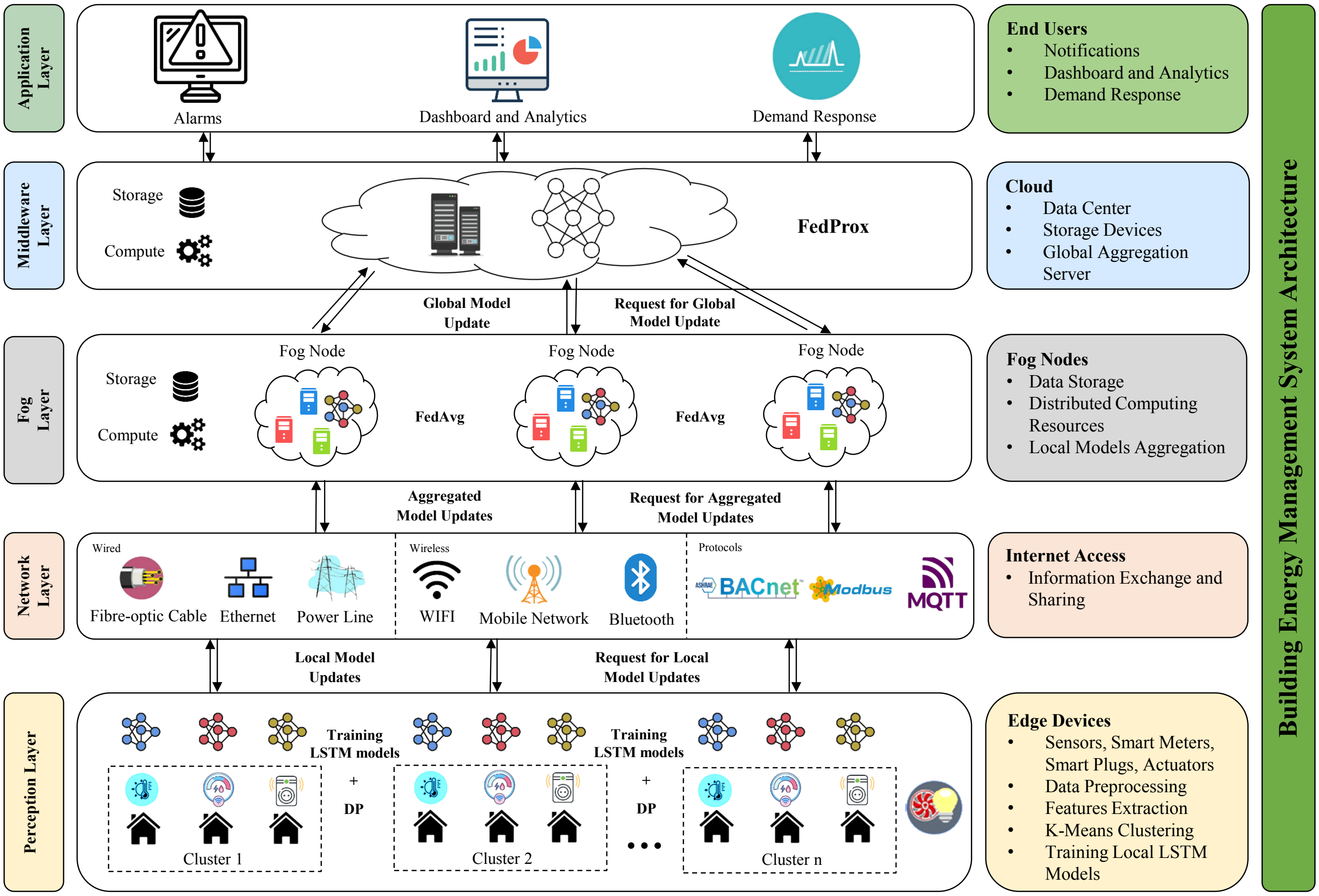

Figure 1: Smart Building Energy Management System Architecture.

for modeling minute-level energy consumption patterns. These models process historical data to predict future energy usage effectively by retaining relevant information over extended sequences while filtering out irrelevant details.

To ensure privacy preservation, Differential Privacy (DP) is applied at the client-side before transmitting model updates to fog nodes. Gaussian noise is added to the gradients, controlled by the privacy budget (ϵ) and failure probability (δ). The noise addition can be mathematically expressed in the equation (4):

$$M(D) = f(D) + N(0, \sigma^2) \tag{4}$$

where $M(D)$ represents the perturbed output, $f(D)$ is a deterministic function applied to the dataset D, and $N(0, \sigma^2)$ represents Gaussian noise with mean 0 and variance σ^2. This approach ensures that individual data contributions remain private, even if the aggregated model updates are exposed.

C. Network Layer

The network layer connects the perception layer (edge devices) with the fog and middleware layers, enabling seamless data transmission and machine-to-machine (M2M) communication for efficient data routing, using various technologies based on speed, range, and reliability.

Wired Technologies:

- **Fibre-Optic Cable:** Ultra-fast, high-bandwidth transmission.
- **Ethernet:** Reliable, high-speed connectivity for local networks.
- **Power Line Communication (PLC):** Data transmission via existing electrical wiring.

Wireless Technologies:

- **WiFi:** Flexible, moderate-range wireless communication.
- **Mobile Networks (e.g., 4G, 5G):** Wide-area connectivity across large regions.
- **Bluetooth:** Short-range communication for low-power devices.

Communication Protocols:

- **BACnet:** For building automation and device interoperability.
- **Modbus:** Reliable industrial data exchange.
- **MQTT:** Lightweight protocol for IoT remote data transmission.

By integrating these technologies and protocols, the network layer ensures efficient, reliable, and secure communication, supporting the edge-fog computing architecture.

D. Aggregation at the Fog Layer

In the fog layer, fog nodes serve as intermediate aggregators, processing and consolidating model updates received from edge devices within their respective clusters. Each fog node employs the Federated Averaging (FedAvg) algorithm to combine model parameters efficiently by averaging the weights from multiple edge devices. The aggregation process is mathematically expressed as:

$$w_{t+1} = \frac{1}{N} \sum_{i=1}^{N} w_i^{(t)} \tag{5}$$

where w_{t+1} represents the aggregated model weights, N is the number of participating devices, and $w_i^{(t)}$ represents the local model weights from device i.

Federated Averaging (FedAvg) is ideal for environments with numerous edge devices, as it minimizes communication overhead by averaging model updates instead of transferring raw data. This lightweight design enhances scalability, making it suitable for resource-constrained environments. Compared to Secure Aggregation (SecAgg), FedAvg is easier to implement, prioritizing efficiency and real-time performance over stronger privacy guarantees. While SecAgg encrypts model updates for privacy, it introduces more computational and communication overhead.

Fog nodes aggregate model updates from edge devices using FedAvg before transmitting them to the cloud. This hierarchical aggregation minimizes data sent to the cloud, optimizing resource utilization and reducing latency. The fog layer acts as a strategic intermediary, balancing computational efficiency, scalability, and communication optimization, ensuring effective federated learning in distributed environments.

E. Cross-Cluster Optimization at the Cloud Layer

At the cloud layer, the models from multiple fog nodes are aggregated and optimized using the Federated Proximal (FedProx) algorithm. FedProx extends Federated Averaging (FedAvg) by introducing a proximal term (μ) to regulate how closely local models follow the global model. This addresses heterogeneity across data and systems, ensuring stability and convergence even with non-IID data. The optimization problem is formulated as:

$$\min_{w} \left(f_i(w) + 2\mu \|w - w_{\text{global}}\|^2 \right) \tag{6}$$

where $f_i(w)$ represents the local loss function for the local model, w denotes the local model weights, and w_{global} refers to the global model weights. The term (μ) controls the regularization strength.

After aggregation, the global model is fine-tuned for specific clusters to improve accuracy and adaptation. It is then redistributed to fog nodes and edge devices for further training until convergence, ensuring satisfactory performance for energy load prediction. This hierarchical process of local training, fog-level aggregation, and cloud-level refinement ensures scalability, accuracy, and adaptability in diverse environments.

F. Application Layer for Analytics

At the application layer, the globally trained model provides actionable insights through dashboards and APIs, enabling proactive energy management and operational efficiency. This layer delivers real-time analytics on energy usage trends and forecasts, offers alerts and notifications to implement demand response strategies, and features user dashboards that present intuitive interfaces for building managers. These capabilities empower users to make informed decisions, optimize energy consumption, and enhance overall building performance.

IV. Implementation and Experimental Setup

A. Technical Details

The study was conducted on an HPE ProLiant ML150 Gen9 Server with a 6-core Intel Xeon CPU (1.7 GHz), 64 GB RAM, a 240 GB SSD for fast data access, and an 8 TB HDD for long-term storage. The server ran Windows Server 2012 R2, ensuring stability for computational processes.

All techniques were implemented in Python, utilizing TensorFlow for neural network training. This setup enabled efficient processing of large datasets and complex models, supporting the simulation of edge, fog, and cloud layers in federated learning for smart building energy prediction.

B. Dataset Description

The Pecan Street dataset was used for simulations. It is a well-known benchmark dataset in the field of energy consumption prediction in smart buildings. This dataset is widely used for evaluating and benchmarking total energy consumption. It includes minute-interval appliance-level customer electricity use from nearly $1,000$ houses and apartments in Pecan Street's multi-state residential electricity use research, as well as ERCOT market operations. For prediction purposes, we took 2 years worth of continuous electricity usage data of 12 houses comprising $17,428$ raw data from the Pecan Street dataset. Using one house of electrical usage was very low to predict so we chose a bunch of houses usage and used a summation of it as the electricity usage. The dataset contains 10 features for each record, which provide information about weather data from weather underground and energy usage data. Table I shows the feature description along with their measurement units.

Table I: Dataset description.

Feature	Unit
DateTime	timestamp
TotalUsage	Kwh
Month	1-12
Temperature	Fahrenheit
Humidity	%
Hour of the day	0-23
Minute of the day	0-3599
Day of week	0-6
Weekend/Weekday	0/1
Holiday	0/1

V. Performance Evaluation

A. Result Discussion

The proposed federated learning-based energy consumption prediction model demonstrated high accuracy, efficiency, and scalability. Key performance metrics include an MSE of 14.55, an MAE of 2.83, and an R^2 value of 0.93, indicating strong predictive power. The Actual vs Predicted plot confirmed the model's ability to capture temporal energy usage patterns accurately, with a prediction accuracy of 93.28%. Low error metrics suggest good generalization and minimal prediction deviations. From a computational perspective, the model achieved real-time predictions with an average latency of 1.97 seconds per instance. However, the total training time of 5904.36 seconds highlights the communication overhead of federated learning. While this decentralized approach enhances privacy and reduces data transfer costs, it also increases computational requirements. The edge-fog architecture facilitated localized model training at edge devices, with aggregation at fog nodes, improving efficiency and scalability. This setup supports real-world deployment by enabling privacy-preserving, distributed energy forecasting without compromising performance.

B. Comparative Analysis

In this section, we compare the performance of the LSTM model with RNN and CNN across key evaluation metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R^2), Accuracy, Average Latency, and Execution Time.

1) Performance Comparison Overview: To comprehensively evaluate the performance of the different models, we compare key metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R^2), Accuracy, Average Latency, and Execution Time. These metrics provide insights into each model's predictive accuracy, computational efficiency, and suitability for real-time deployment in smart building energy management systems.

The table II below summarizes the performance of RNN, CNN, and LSTM models on the test dataset.

This comparison highlights how each model performs under the same experimental conditions, offering a clear perspective on their strengths and limitations.

2) Evaluation: The LSTM model outperforms the other techniques across most critical evaluation metrics. It achieved the lowest Mean Squared Error (14.55) and Mean Absolute Error (2.83), reflecting its exceptional accuracy in predicting energy consumption. Additionally, the R^2 value of 0.93 indicates that the LSTM model captures 93% of the variance in the observed data, showcasing its robustness in modeling time-series dependencies. Its accuracy of 93.28% further solidifies its effectiveness for energy prediction tasks.

The CNN model demonstrated the fastest execution time (5541.73 seconds) and the lowest average latency (1.71 seconds), making it computationally efficient. However, its predictive performance (MSE: 26.16, MAE: 3.79) falls behind the LSTM model, which offers better prediction accuracy.

The RNN model showed the weakest predictive performance among the three, with the highest Mean Squared Error (40.81) and Mean Absolute Error (4.46), as well as the lowest R^2 value (0.81) and accuracy (81.14%). Its execution time (6718.88 seconds) and latency (2.65 seconds) were also higher compared to CNN, making it less efficient for real-time applications.

Overall, the LSTM model remains the optimal choice for energy consumption prediction in smart buildings, balancing predictive accuracy, latency, and computational efficiency.

3) Prediction Visualizations: To further illustrate the differences in prediction performance, the following figures compare the Actual vs. Predicted energy consumption values for each model. These visualizations offer deeper insights into how effectively each approach captures temporal dependencies and reduces prediction deviations.

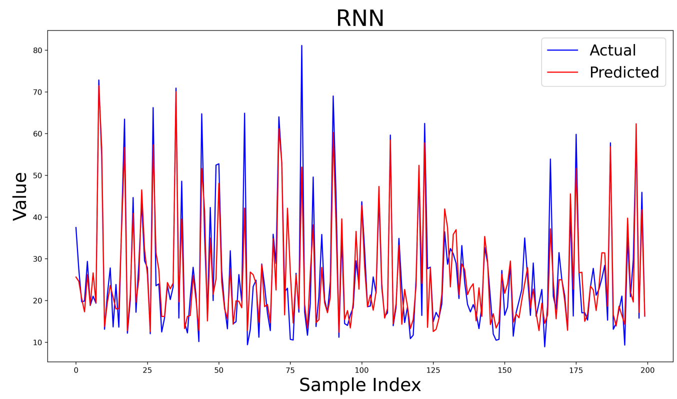

Figure 2: Actual vs. Predicted - RNN.

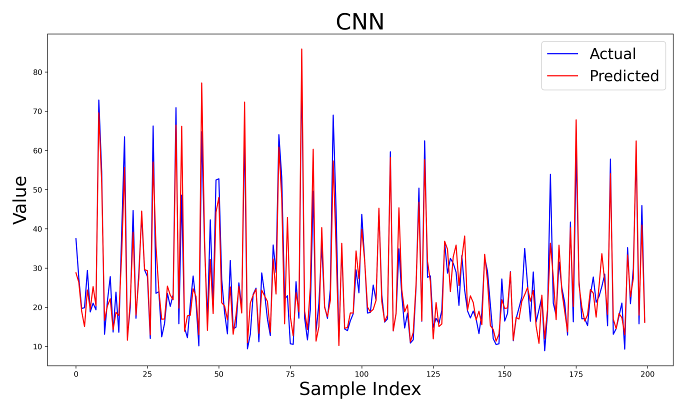

Figure 3: Actual vs. Predicted - CNN.

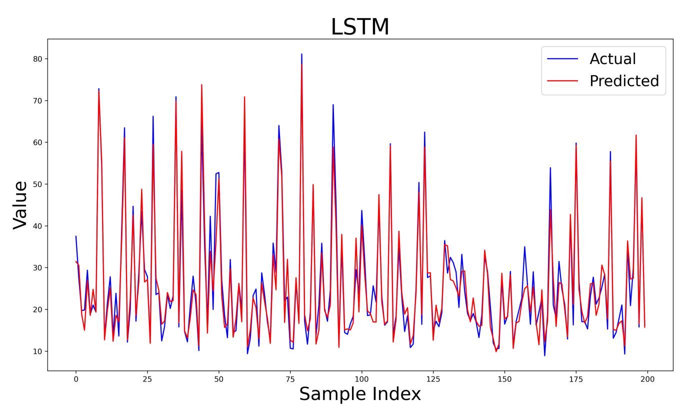

Figure 4: Actual vs. Predicted - LSTM.

As clearly illustrated in Figure 4, the LSTM model stands out by closely following the actual energy consumption trends

Model	MSE	MAE	R^2	Accuracy	Average latency	Execution time
RNN	40.81	4.46	0.81	81.14	2.65	6718.88
CNN	26.16	3.79	0.88	87.91	**1.71**	**5541.73**
LSTM	**14.55**	**2.83**	**0.93**	**93.28**	1.97	5904.36

Table II: Model Performance Comparison Across Key Metrics.

with minimal deviation. This remarkable alignment highlights its ability to capture both long-term trends and short-term fluctuations effectively. In stark contrast, Figure 2 demonstrates that the RNN model exhibits substantial gaps between actual and predicted values, underscoring its limitations in capturing precise temporal dependencies and long-term patterns.

The CNN model (Figure 3), while showing moderate alignment, falls short in its ability to model sequential dependencies, which is a critical factor in energy consumption prediction. Its predictions exhibit more deviation compared to LSTM, reinforcing the latter's superior performance in minimizing prediction errors.

These observations highlight the LSTM model as the most robust and reliable approach for energy consumption prediction, offering a balance of high accuracy and computational efficiency that the other models fail to match.

VI. Conclusion

This paper introduced a privacy-preserving energy consumption prediction framework for smart buildings, combining Federated Learning (FL) with a hierarchical Edge-Fog-Cloud architecture and Differential Privacy (DP) mechanisms. The proposed approach addressed critical challenges, including high computational demand, data privacy concerns, and communication overhead, by decentralizing the training process across edge devices, fog nodes, and cloud servers. Our findings validate the effectiveness of decentralized and privacy-aware energy prediction systems, offering a scalable solution for modern energy management in smart buildings.

Acknowledgment

This work has been achieved in the frame of the EIPHI Graduate school (contract "ANR-17-EURE-0002").

References

[1] Paul Wilkinson, Kirk R Smith, Sean Beevers, Cathryn Tonne, and Tadj Oreszczyn. Energy, energy efficiency, and the built environment. *The lancet*, 370(9593):1175–1187, 2007.

[2] Jennifer King and Christopher Perry. *Smart buildings: Using smart technology to save energy in existing buildings*. Amercian Council for an Energy-Efficient Economy Washington, DC, USA, 2017.

[3] Georgios Martinopoulos, Konstantinos T Papakostas, and Agis M Papadopoulos. A comparative review of heating systems in eu countries, based on efficiency and fuel cost. *Renewable and Sustainable Energy Reviews*, 90: 687–699, 2018.

[4] Joey Li, Munur Sacit Herdem, Jatin Nathwani, and John Z Wen. Methods and applications for artificial intelligence, big data, internet of things, and blockchain in smart energy management. *Energy and AI*, 11:100208, 2023.

[5] Neco Villegas, Luis Diez, Idoia De la Iglesia, Marco González-Hierro, and Ramón Agüero. Energy-aware optimum offloading strategies in fog-cloud architectures: A lyapunov based scheme. *IEEE Access*, 2023.

[6] Aya Sayed, Yassine Himeur, Abdullah Alsalemi, Faycal Bensaali, and Abbes Amira. Intelligent edge-based recommender system for internet of energy applications. *IEEE Systems Journal*, 16(3):5001–5010, 2021.

[7] Muhammad Umair, Muhammad Aamir Cheema, Bilal Afzal, and Ghalib Shah. Energy management of smart homes over fog-based iot architecture. *Sustainable Computing: Informatics and Systems*, 39:100898, 2023.

[8] Mohammed A Saeed, Abdelfattah A Eladl, Bilal Naji Alhasnawi, Saad Motahhir, Anand Nayyar, Mohd Asif Shah, and Bishoy E Sedhom. Energy management system in smart buildings based coalition game theory with fog platform and smart meter infrastructure. *Scientific Reports*, 13(1):2023, 2023.

[9] Yung-Yao Chen, Ming-Hung Chen, Che-Ming Chang, Fu-Sheng Chang, and Yu-Hsiu Lin. A smart home energy management system using two-stage non-intrusive appliance load monitoring over fog-cloud analytics based on tridium's niagara framework for residential demand-side management. *Sensors*, 21(8):2883, 2021.

[10] Mohammad Navid Fekri, Katarina Grolinger, and Syed Mir. Distributed load forecasting using smart meter data: Federated learning with recurrent neural networks. *International Journal of Electrical Power & Energy Systems*, 137:107669, 2022.

[11] Joaquín Delgado Fernández, Sergio Potenciano Menci, Chul Min Lee, Alexander Rieger, and Gilbert Fridgen. Privacy-preserving federated learning for residential short-term load forecasting. *Applied energy*, 326:119915, 2022.

[12] Nastaran Gholizadeh and Petr Musilek. Federated learning with hyperparameter-based clustering for electrical load forecasting. *Internet of Things*, 17:100470, 2022.

[13] Keon-Jun Park and Sung-Yong Son. Residential load forecasting using modified federated learning algorithm. *IEEE Access*, 2023.

[14] Junyang Li, Chaobo Zhang, Yang Zhao, Weikang Qiu, Qi Chen, and Xuejun Zhang. Federated learning-based short-term building energy consumption prediction method for solving the data silos problem. In *Building Simulation*, volume 15, pages 1145–1159. Springer, 2022.