2024 IEEE 30th International Conference on Parallel and Distributed Systems (ICPADS) | 979-8-3315-1596-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICPADS63350.2024.00078

A QoE-Aware Adaptive Energy-Efficient Transmission Scheduling Method

Yankun Yuan*, Lin Wang*✉, Chonghui Xiao†, Zijuan Liu‡, Fan Dang§, Xu Wang§, Haitian Zhao‡, Xin Miao¶
*School of Information Science and Engineering, Yanshan University
†National Computer Network Emergency Response Technical Team/Coordination Center of China
‡Department of Automation and BNRist, Tsinghua University
§Global Innovation Exchange, Tsinghua University
¶School of Software and BNRist, Tsinghua University
yankun@stumail.ysu.edu.cn, wlin@ysu.edu.cn, xch@cert.org.cn,
{dangfan, xu_wang, zhaoht2022, miaoxin}@tsinghua.edu.cn

***Abstract*—In this paper, we propose a dynamic data transmission strategy for smart home environments that aims to optimize the Quality of Experience (QoE) by adaptively adjusting the data upload frequency based on the predicted trends in sensor data. Using the home wireless sensors monitoring dataset, we implement a deep learning model for accurate time series forecasting. In addition, an anomaly detection mechanism is used to identify critical events, requiring more frequent data uploads when important changes are detected. The QoE is quantified through a weighted average of several influencing factors, including data timeliness, timely upload of critical events, and transmission frequency. Our optimization objective is to maximize QoE while minimizing the number of transmissions, with an emphasis on reducing energy consumption through intelligent scheduling. The results demonstrate that our approach effectively balances data timeliness, transmission efficiency, and energy savings, leading to improved user satisfaction in smart home applications.**

***Index Terms*—QoE, LSTM Models, Energy conservation, Anomaly detection, IoT, Adaptive algorithm**

I. Introduction

With the rapid development and deployment of the Internet of Things (IoT) and network intelligence, efficient management and timely transmission of sensor data are crucial for enhancing user experience. In smart home environments, QoE depends not only on the accuracy of data collected from various sensors but also on the timely and efficient transmission of these data. To optimize QoE, the key is to balance the data upload frequency with the need to capture critical events while minimizing energy consumption. This requires an intelligent data management system that can dynamically adjust the transmission strategy based on the importance and urgency of the data.

Smart home systems are equipped with numerous sensors to monitor various environmental parameters such as temperature, humidity, light, and energy consumption. These sensors generate a significant amount of data that needs to be processed and transmitted in a manner that maximizes user satisfaction while considering energy efficiency. The challenge lies in determining the optimal timing for uploading these data without compromising the capture of critical information or user experience. Uploading too frequently can waste energy and bandwidth resources, while uploading too infrequently may result in missing important events. Therefore, an intelligent data management system needs to find the optimal balance between these two extremes.

Previous research has explored various methods for time series prediction and anomaly detection in smart home system environments. Traditional approaches [1] like ARIMA and linear regression models have been employed, but they often underperform when dealing with the inherent nonlinearity and complexity of sensor data. With the rise of deep learning, models such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have shown promising results in accurately predicting future data points [2]. These models can capture long-term dependencies and complex patterns in time series data, making them well-suited for smart home applications. Deep learning approaches not only improve prediction accuracy, but also adaptively learn data features, reducing the need for manual feature engineering.

In this paper, we propose a dynamic data transmission strategy that leverages deep learning for accurate time series prediction and anomaly detection.The dynamic data transmission strategy may decrease the accuracy of time series prediction, thereby affecting subsequent anomaly detection and user QoE. The key challenge lies in effectively addressing this issue. Our main objectives are threefold: *first*, to achieve accurate time series prediction and adjust the data upload frequency based on the predicted trends; *second*, to develop and evaluate anomaly detection algorithms to capture critical events and anomalies; and *third*, to define and optimize QoE metrics for a comprehensive evaluation and improvement of user experience. By combining these three objectives into a unified framework, our approach comprehensively addresses data management challenges in smart homes.

We use a home wireless sensors monitoring dataset [3] to train deep learning models. The model's predictions help determine the optimal timing for data transmission, balancing the need for timely updates and energy efficiency. An integrated anomaly detection mechanism ensures immediate

✉ Lin Wang is the corresponding author.

979-8-3315-1596-6/24/$31.00 ©2024 IEEE
DOI 10.1109/ICPADS63350.2024.00078

uploads when significant data changes occur, capturing critical events. This adaptive approach helps minimize unnecessary transmissions while ensuring important data is not missed. By continuously optimizing the model using historical data and real-time feedback, the system can learn and refine its decision policy over time.

We have conducted a detailed summary of our system. Our results demonstrate that the proposed strategy effectively balances data timeliness, transmission efficiency, and energy conservation, thereby improving user satisfaction in smart home applications. Compared to periodic data transmission methods methods, our deep learning-driven approach performs better across various QoE metrics, demonstrating its effectiveness and applicability in smart home scenarios. The insights gained from this study provide valuable guidance for the development of intelligent data management systems in smart homes, emphasizing the importance of adopting adaptive and efficient data transmission strategies. Future work can explore extending this approach to other types of IoT applications and integrating it with emerging technologies such as edge computing and federated learning to further enhance system performance and scalability.

The main contributions of this work can be summarized as follows:

- We identify the challenges of optimizing dynamic upload frequency in smart home environments resulting in reduced prediction accuracy and user QoE and proposed a dynamic data transmission strategy. This strategy combines deep learning-based time series forecasting with anomaly detection to adaptively adjust the data upload frequency, ensuring efficient and timely data transmission.
- By implementing a dual-layer LSTM network for time series forecasting, our approach provides highly accurate predictions of sensor data trends. This allows for more precise scheduling of data uploads, ensuring critical events are captured without unnecessary transmissions.Compared to traditional fixed-interval transmission strategies, our approach can reduce transmission frequency by approximately 60% on average with minimal impact on user QoE.
- We defined a set of QoE metrics that consider data accuracy, timely upload of critical events, and transmission frequency. Our approach optimizes these metrics to provide a balanced evaluation of user experience, leading to improved satisfaction in smart home applications.

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature, focusing on key contributions such as the Adaptive Data Transmission Strategy, Deep Learning Integration for Accurate Predictions, and QoE Metric Optimization. Section 3 and 4 present our methodology and experimental evaluation, detailing the development and implementation of these strategies. Section 5 concludes with a summary of the findings and suggestions for future research, particularly focusing on QoE optimization and energy-efficient transmission scheduling in smart home environments.

II. Related Works

A. Energy-Efficient Modeling and Scheduling Methods

With the advancement of hardware and software in the field of computing, energy-efficient scheduling methods [4] have been extensively researched. Before the advent of machine learning, many studies focused on mathematical methods for scheduling, with the aim of optimizing energy consumption through the design of protocols and algorithms. For example, Yao et al. [5]. proposed an inverse logarithmic algorithm that achieves energy savings in wireless sensor networks by adjusting the transmission time allocated to each sensor. In the context of collaborative execution in mobile cloud computing, Zhang et al. [6]. formulated the problem as a constrained shortest path problem on a directed acyclic graph, utilizing a Lagrangian Relaxation Aggregated Cost (LARAC) algorithm to schedule task offloading to the cloud, thereby reducing device energy consumption. Furthermore, several other studies used various mathematical and heuristic algorithms to target energy reduction in different application scenarios [7]–[9].

In recent years, extensive research in machine learning has established the use of machine learning for energy-efficient scheduling as an effective paradigm [10]. Time-series data, which are common data forms in energy-efficient scheduling processes, benefit importantly from machine learning models compared to traditional methods. Classical models such as ARIMA and linear regression are commonly used for sequence prediction [11], [12]. Although these models offer good interpretability and can effectively fit simple linear data, they struggle to model complex nonlinear relationships. Recurrent Neural Networks are representative models in deep learning, suitable for processing time series data or sequential data. Common models include RNNs, GRUs, LSTMs, and their variants [13], [14]. Original RNNs suffer from problems with vanishing and exploding gradients [15]. To achieve higher prediction accuracy, LSTM models are employed to better capture complex time series. The work of Siami et al. shows that LSTM outperforms traditional ARIMA and linear regression models [16]. Although some studies attempt to improve prediction performance by improving model architectures or applying attention mechanisms [17]–[19], these models often lack the ability to capture temporal information as effectively as recurrent neural networks, potentially becoming a bottleneck in scheduling algorithms.

B. Wireless Sensor Network Applications and QoE Design

Wireless Sensor Networks (WSNs) are increasingly used to detect various environmental parameters in a wide range of application scenarios due to their low cost, flexible deployment, and large-scale self-organizing capabilities. For example, Lombardo et al. proposed a distributed WSN deployment scheme to detect environmental temperature and humidity [20]. Tien et al. developed a WSN architecture for monitoring agricultural environmental data [21]. Ullo et al. created a WSN application for public transportation, which provides real-time traffic and environmental conditions through sensors, allowing optimal route planning to avoid public transportation congestion [22]. Using the ease of deployment and large-scale networking of WSNs, Mabrouki et al. proposed an automated weather monitoring system to detect and forecast weather data in specific areas [23]. Based on a real-world deployed IoT smart gas meter system, Wang et al. designed an ultra-low-power wireless sensor device management framework to achieve stable and accurate measurement with lower power consumption [24]. In addition, WSNs are also applied in various fields such as construction, healthcare, and manufacturing. These widely deployed devices help build digital environmental models to assist decision-making.

As numerous wireless sensors are deployed, the QoE for users of wireless network systems should also be considered. QoE reflects the subjective perception and acceptability of users, and some studies infer users' subjective feelings through objective parameters to design QoE metrics. Yasuhiro et al. proposed a scalable IoT QoE modeling framework that qualitatively models physical parameters such as devices, networks, computation and user interfaces according to application characteristics [25]. Redowan et al. proposed a fog computing method that ranks and deploys fog applications to the corresponding fog instances according to user expectations to maximize QoE [26]. Amulya et al. argued that in IoT environments, the quality of interactions between things should be more considered, thus redesigning QoE as QoT to reflect the quality of interaction in IoT [27]. Chen et al. proposed a network multi-layer collaboration method to model Qoe in wireless video streaming transmission in emotion-aware intelligent systems and further reduced transmission energy consumption and improved QoE [28]. In summary, QoE needs to consider different influencing factors in different scenarios to better optimize and improve the user experience.

III. Methodology

A. Overall Design

To achieve a scheduling method that balances user experience quality and sensor data upload frequency, selecting appropriate time points for sensor data upload is a feasible and effective strategy. Considering the limited computational capabilities of most sensors, all decisions should be made by the data center and then communicated to the sensors. Therefore, predicting future trends in sensor data and incorporating user requirements into scheduling decisions is essential.

The key challenge lies in obtaining accurate real-time sensor data, as this is crucial to accurately predict future sensor data trends. We aim to reduce the frequency of sensor data uploads when data trends are stable, minimizing unnecessary transmissions. In contrast, when data trends show important changes, it is crucial to ensure timely uploads and adjust our prediction models accordingly. This approach ensures that even if real-time sensor data is not continuously available, prediction accuracy is maintained while reducing the data transmission frequency. From the user's perspective, reducing the number of sensor uploads extends sensor lifespan and reduces maintenance frequency. Timely data uploads during significant trend changes ensure that users can monitor abnormal changes in real time. In addition, the scheduling process considers user requirements for real-time information, ensuring that users can access critical information in the shortest possible time.

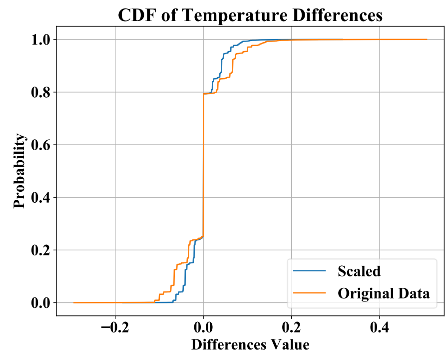


Fig. 1: A sample of data differences.

In our system design, to ensure the quality of user experience as much as possible, we have introduced a solution that integrates LSTM, differential anomaly detection, and historical data interpolation. Fig. 2 illustrates our overall design: wireless sensors collect sensing data in their respective environments, and the central device trains sensor data predictors in different scenarios. Based on the predicted results of data trends and user restrictions on data upload time, the central device determines the optimal time for wireless sensors to establish wireless connections with the central device to upload data. When sensors do not upload data, Planner temporarily substitutes the predicted values of the predictor for the uploaded data of the sensors until the sensors upload real data and then updates the historical data through interpolation. Even if the data predictor predicts a gentle trend in future data changes, Planner still considers the user's settings for data timeliness and the minimum upload frequency to select the appropriate upload time points to timely update the data predictor and historical data.

Another key insight is that, when the data trend is relatively gentle, the sensor's data will not undergo drastic fluctuations or very small changes in the next few time steps. As shown in Fig. 1 , this scenario accounts for a considerable proportion of the data uploaded by sensors, ensuring the precision of the data prediction and historical data updates, and providing a basis for the algorithm to adaptively adjust the upload frequency based on data trends.

B. Details of Predictor

In designing our data predictor, we focus on capturing variations in time series data to achieve high-precision data predictions. Since most sensors used for environmental awareness naturally produce time series data, Long Short-Term Memory (LSTM) networks are an ideal choice. LSTM networks leverage the strengths of recurrent neural networks to

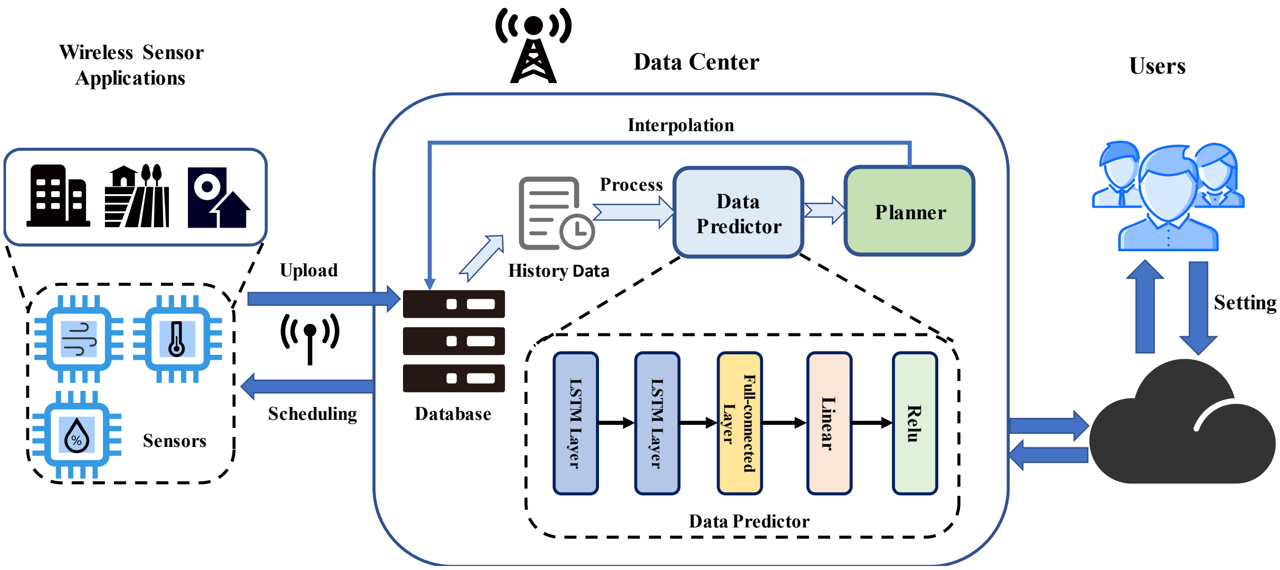


Fig. 2: System overview.

process sequential inputs, dynamically training the model to adjust parameters, and capture sequence variations in the data.

As illustrated in Fig. 2, our predictive model consists of two LSTM layers, a fully connected layer, and a linear layer, delivering predictions through a ReLU activation function. Specifically, the hidden size of the LSTM layers is set to 50, with a dropout regularization rate of 0.2 to prevent overfitting. During the training process, we employ the Adam optimization algorithm and use the Mean Squared Error (MSE) as the loss function, with an initial learning rate set to 0.001.

In the fully connected layer, we set the number of neurons to (hidden size + output size) // 2 to balance the complexity of the inputs and outputs with computational efficiency. This architectural design aims to fully exploit the time series processing capabilities of LSTM, ensuring that the model accurately captures the dynamic changes in sensor data, thus providing high-quality predictions.

C. Planner & QoE Design

In the Planner section, Algorithm 1 illustrates the operational workflow of the entire algorithm. The data center database stores all the data uploaded by wireless sensors and the data temporarily filled by the algorithm. For each application scenario, the data center trains the predictors for the sensors. The Planner predicts future time steps based on inputs from a historical time window and computes thresholds using the difference values over a recent period and an anomaly detection sensitivity. The threshold calculation involves the absolute mean difference value of the data differentials over a recent period multiplied by the standard deviation of these differentials. The authenticity of this data trend change is crucial for users who focus on the evolution of data trends rather than the magnitude of data changes. Such changes in data trends may signify the occurrence of abnormal events. To

Algorithm 1 Predict & Plan

Input: History Data D, time windows w, predict model m, max interval I_{max}, anomaly detection sensitivity s, filtering parameter f
Output: If the next time step upload

```
Scaled data D
Initialize time windows w ⇐ D
while true do
    if Received the real data then
        if Exist values in D that were replaced with predicted values then
            Interpolate these values in D with real data
        end if
    end if
    Calculate the next time step values with w and m
    Calculate the diff threshold with w and s
    if current diff(x_{i+1}^{predict} − x_i) > threshold then
        if current diff and w satisfied f then
            return True
        end if
    else
        if Time interval satisfied I_max or user's setting then
            return True
        end if
    else
        Update D and w with x_{i+1}^{predict}
    end if
    To next time step
    return False
end while
```

address this, a filtering parameter is used to filter out minute changes based on the standard deviation, ensuring that they do not disrupt other processes.

$$\begin{aligned} &\arg\max_{I} - \left(\alpha N_{\text{total}} + \beta \sum (T_{\text{upload}} - T_{\text{anormal}})\right), \\ &\text{subject to} \quad I < I_{\max}. \end{aligned} \tag{1}$$

The formula 1 illustrates the QoE objectives that we aim to optimize. Here, N_{total} denotes the total number of uploads, T_{upload} represents the closest upload step to the anomaly time step $T_{anormal}$, and α and β signify the weights that affect user QoE. I signifies the interval between data uploads, and I_{max} is determined by user settings and model errors. QoE design aims to reflect the user experience as quantitatively as possible. In the context of wireless sensor applications, users prefer to minimize data transmission frequencies due to associated maintenance overhead. Simultaneously, users want prompt access to all critical data to mitigate potential losses caused by environmental anomalies detected by sensors. In addition, users can customize the maximum data upload interval to ensure a minimal data acquisition frequency.

Based on the above considerations, the Planner selects future transmission times while taking into account QoE and user settings. During periods of sensor data nontransmission, the Planner temporarily substitutes sensor data with predictions from the data predictor. Recognizing potential cumulative errors in model predictions, the Planner sets a maximum upload interval based on user preferences and deviation from model predictions. Beyond this interval, the Planner schedules data uploads from sensors. Upon receiving actual sensor data uploads, the data center interpolates the data replaced by the predictor between actual values, promptly updates model parameters, and ensures adaptive system adjustments over time to maintain data prediction accuracy and optimize user QoE.

IV. Evaluation

A. Dataset and Experiment Settings

Our study uses a dataset collected between January and May 2016 from a residential building in Belgium, using a Zigbee sensor network [3]. The dataset includes data from nine temperature and humidity sensors placed at various locations within the building and an outdoor wind speed sensor. In addition, it includes outdoor weather data from the nearest airport. Each sensor in the dataset transmitted data every 10 minutes. For training the predictor and conducting simulation experiments, we selected data from eight temperature sensors, eight humidity sensors, and one outdoor wind speed sensor, resulting in a total of 19,736 data points per sensor. The last 10% of the data was set aside for simulation experiments, and the remaining data was divided into training, validation, and testing sets in a 60%, 20%, and 20% ratio, respectively. In the simulation phase, the algorithm was executed sequentially over time. The maximum upload interval was set to 6 time steps, the sensitivity coefficient to 3, and the historical time window size to 10 time steps. The predictor's forecast values were replaced with actual data values to serve as the ground truth for evaluating the algorithm's performance.

In the following sections, we will gradually present the performance of our predictor and the experimental results, along with an analysis of the algorithm's effectiveness considering QoE. Our results demonstrate that, compared to the fixed periodic data transmission mode typically used in most current wireless sensor scenarios, our algorithm provides a superior energy-saving data transmission scheme while taking user QoE into account.

B. Evaluation of Predictor Performance

This section discusses the prediction performance of sensor data predictors across different types and scenarios. We use historical data from sensors over ten time steps as input, which is standardized using Z-score before prediction. Our primary goal is to schedule wireless sensor data uploads to meet user QoE based on trends in each sensor data.

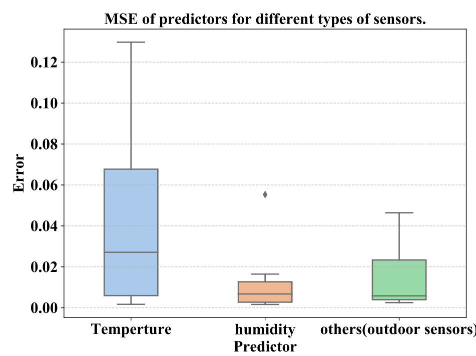
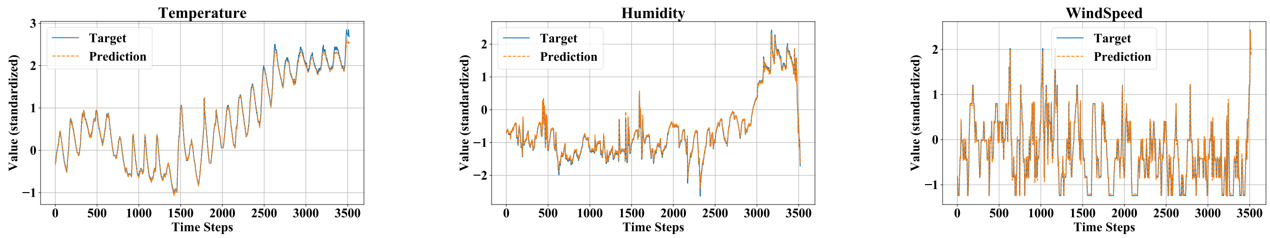


Fig. 3: MSE of different predictor.

Fig. 4 shows the prediction performance of predictors for different types of sensors on the test dataset. The results demonstrate that our predictor architecture effectively captures the development trends of various data types and provides accurate predictions. Besides, we observe that while most environmental monitoring data exhibit minimal short-term variation, long-term patterns are complex, potentially impacting recursive predictions over multiple time steps. Fig. 3 illustrates the Mean Squared Error (MSE) of predictors on standardized data for different types of sensors. In the figure, "others" denotes sensors that include outdoor weather data such as wind speed. For most sensor data, predictors achieve an acceptable accuracy. However, temperature sensors exhibit larger errors compared to other sensors. A plausible explanation is that temperature sensors are placed in various locations within the home, such as bathrooms and ironing rooms, where temperature variations can be more abrupt, so affecting the overall predictor performance. Overall, the effectiveness of predictors in different types and scenarios suggests that scheduling wireless sensor data uploads based on trends in data variation is feasible to improve user QoE.



(a) Temperature predictor (b) Humidity predictor (c) Windspeed predictor

Fig. 4: Performance of the predictor samples for temperature, humidity, and windspeed on the test dataset.

C. *Simulated performance of scheduling*

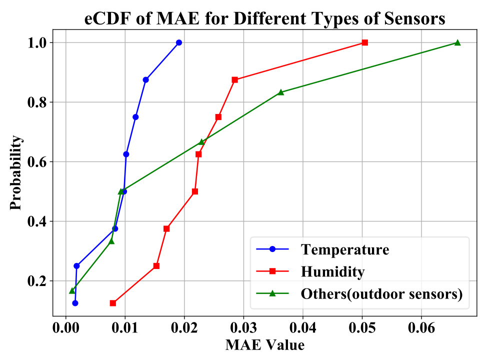


Fig. 5: eCDF of MAE for different predictor.

This section presents the simulation results of our scheduling algorithm on real datasets. Fig. 6 compares the simulated data curves obtained by our scheduling planner with the actual data curves and shows the selected data upload time points of wireless sensors under QoE constraints. The algorithm predicts future data trends based on historical data. When the planner detects that the data differ from previous trends, it increases the upload frequency of the wireless sensor. Meanwhile, when the data trend is stable, the planner periodically schedules the wireless sensor to upload data to meet the predictor's correction and the user's personalized settings. The results indicate that, for most time points, even without uploading data, the simulated data curve by the data center algorithm can still fit the actual data trend well. The selected upload time points effectively capture moments with important data trend changes. When the data trend slope tends to remain unchanged, the algorithm can detect very slight trend changes and schedule sensor uploads to confirm the real situation. Fig. 5 shows the eCDF of MAE between the simulated data curves and actual data curves for different scenarios and types of wireless sensors. The data is denormalized back to actual environmental sensor data such as temperature, humidity, and wind speed. The results show that 80% of different types of sensors have an MAE of less than 0.04, with a maximum error of less than 0.07. For most application scenarios, this error is almost negligible. Besides, the results demonstrate that for users, the algorithm can reduce the energy consumption of wireless sensors without significantly altering the accuracy of the original data. Users can access historical data at any time through the data center, even if the sensor did not upload real data at that time, which demonstrates the effectiveness of our algorithm.

D. *Evaluation of QoE*

TABLE I: QoE metrics for different type of sensors.

Type	**Avg**	**Max**	**Min**	**Median(mins)**
Temperature	1043.75	1579	377	5.0
Humidity	1272.75	1512	989	6.25
others	1196.83	1590	655	6.7

This section evaluates the QoE, consistent with the QoE optimization objectives defined in Chapter 3. We identify the primary objective parameters that influence the user experience, such as the total number of uploads and the time interval to detect anomalous data. In our algorithm design, we account for user-defined transmission frequency settings, and prior experimental results demonstrate the data center's capability to generate usable data even when sensors do not transmit real-time data. Therefore, objective parameters are considered important in shaping the quality of the user experience. Table 1 presents the experimental results of our algorithm's efforts in optimizing user QoE.

The QoE experiments were conducted on a dataset with a time-step interval of 10 minutes and a total of 1973 time steps. Here, "Avg", "Max", and "Min", respectively, denote the average, maximum and minimum reductions in uploads compared to a fixed interval upload strategy. The "median" represents the median upload delay for anomalous data trends. The results indicate that our algorithm importantly reduces upload frequencies, thereby reducing energy consumption for wireless sensors. Even the minimal reductions in upload frequency are noteworthy. The average median upload delay demonstrates our ability to upload data within a median delay of less than one time step compared to a fixed-interval transmission strategy, which justifies the reduction in energy consumption. Furthermore, because of our model's accurate predictive capabilities, users can accurately access data for

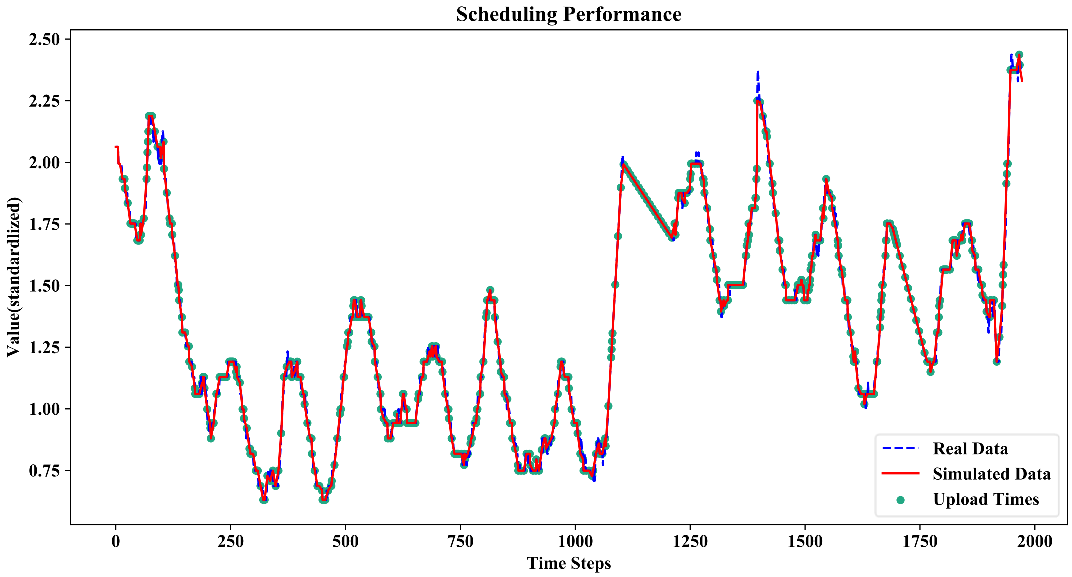


Fig. 6: A sample of scheduled data collection.

time steps with anomalous trends, even if they were not uploaded. In summary, the experimental findings demonstrate that our approach effectively reduces sensor data upload frequencies while meeting user QoE requirements, thus validating the efficacy of our method.

V. Conclusion

This study identifies that traditional fixed interval upload strategies in low-power wireless sensor networks [29], [30] incur high maintenance costs and inefficient utilization of transmitted data, which diverge importantly from user requirements. To address this issue, we propose a sensor upload scheduling method for wireless sensor networks aimed at reducing the frequency of sensor data uploads to lower the energy consumption associated with data transmission, ultimately enhancing user QoE. Additionally, we introduce a series of methods to ensure consistency between the reduced upload frequency system and the original system's data acquisition. Simulation experiments demonstrate that our approach offers a plug-and-play solution without altering the existing wireless sensor network architecture compared to traditional fixed-interval upload strategies. The algorithm is deployed in the data center without the need to modify the software or hardware of the edge device. Our method importantly reduces the energy consumption of sensor network devices and the experimental results indicate minimal impact on the integrity of the data center, ensuring user QoE.

Although our approach effectively reduces the overall energy consumption of the sensor network and maintains user QoE, the reduction in the data transmission frequency poses the risk of missing sudden events. We propose automatic correction mechanisms to mitigate this issue, yet further optimization is necessary. Finally, this study presents an effective method to reduce energy consumption in wireless sensor networks while meeting user QoE requirements. Future work can build upon this study to further enhance user satisfaction.

Acknowlegment

This work is supported in part by Tsinghua University - Architectural Design and Research Institute Joint Research Center for Synergy and Wisdom Creation of Architecture and the NSFC under Grant No. 62302259 and 62202263.

References

[1] Hongjun Wang, Kaiying Wang, Hui Zhao, and Youjun Yue. Prediction of user behavior in smart home based on improved arima model. In *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 298–302, 2018.

[2] Bahram Gharabaghi Brian S. Freeman, Graham Taylor and Jesse Thé. Forecasting air quality time series using deep learning. *Journal of the Air & Waste Management Association*, 68(8):866–886, 2018. PMID: 29652217.

[3] M. GAGAN CHANDRA. house wireless sensor monitoring. https://www.kaggle.com/datasets/gaganmaahi224/appliances-energy-consumption, 2021.

[4] Zhuqing Xu, Junzhou Luo, Zhimeng Yin, Shuai Wang, Ciyuan Chen, Jingkai Lin, Runqun Xiong, and Tian He. Leveraging imperfect-orthogonality aware scheduling for high scalability in lpwan. *IEEE Transactions on Mobile Computing*, 2024.

[5] Yingwei Yao and G.B. Giannakis. Energy-efficient scheduling for wireless sensor networks. *IEEE Transactions on Communications*, 53(8):1333–1342, 2005.

[6] Weiwen Zhang, Yonggang Wen, and Dapeng Oliver Wu. Energy-efficient scheduling policy for collaborative execution in mobile cloud computing. In *2013 Proceedings IEEE INFOCOM*, pages 190–194, 2013.

[7] Xiaoyong Tang and Zhuojun Fu. Cpu–gpu utilization aware energy-efficient scheduling algorithm on heterogeneous computing systems. *IEEE Access*, 8:58948–58958, 2020.

[8] Saurabh Kumar and Hyungwon Kim. Energy efficient scheduling in wireless sensor networks for periodic data gathering. *IEEE Access*, 7:11410–11426, 2019.

[9] J. Ma, W. Lou, Y. Wu, X.-Y. Li, and G. Chen. Energy efficient tdma sleep scheduling in wireless sensor networks. In *IEEE INFOCOM 2009*, pages 630–638, 2009.

[10] D. Praveen Kumar, Tarachand Amgoth, and Chandra Sekhara Rao Annavarapu. Machine learning algorithms for wireless sensor networks: A survey. *Information Fusion*, 49:1–25, 2019.

[11] Chenghao Liu, Steven C.H. Hoi, Peilin Zhao, and Jianling Sun. Online arima algorithms for time series prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), Feb. 2016.

[12] Nelson Fumo and M.A. Rafe Biswas. Regression analysis for prediction of residential energy consumption. *Renewable and Sustainable Energy Reviews*, 47:332–343, 2015.

[13] Nipun Ramakrishnan and Tarun Soni. Network traffic prediction using recurrent neural networks. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 187–193, 2018.

[14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[15] Alexander Rehmer and Andreas Kroll. On the vanishing and exploding gradient problem in gated recurrent units. *IFAC-PapersOnLine*, 53(2):1243–1248, 2020. 21st IFAC World Congress.

[16] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1394–1401, 2018.

[17] Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. Segrnn: Segment recurrent neural network for long-term time series forecasting. *arXiv preprint arXiv:2308.11200*, 2023.

[18] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):11106–11115, May 2021.

[19] Ziyue Dang, Fan Dang, and Yankun Yuan. Lstm-driven scheduling for energy-efficient crop monitoring in wireless networks. In *2023 20th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 41–47, 2023.

[20] Luca Lombardo, Simone Corbellini, Marco Parvis, Ahmed Elsayed, Emma Angelini, and Sabrina Grassini. Wireless sensor network for distributed environmental monitoring. *IEEE Transactions on Instrumentation and Measurement*, 67(5):1214–1222, 2018.

[21] Tien Cao-hoang and Can Nguyen Duy. Environment monitoring system for agricultural application based on wireless sensor network. In *2017 Seventh International Conference on Information Science and Technology (ICIST)*, pages 99–102, 2017.

[22] S. Ullo, M. Gallo, G. Palmieri, P. Amenta, M. Russo, G. Romano, M. Ferrucci, A. Ferrara, and M. De Angelis. Application of wireless sensor networks to environmental monitoring for sustainable mobility. In *2018 IEEE International Conference on Environmental Engineering (EE)*, pages 1–7, 2018.

[23] Jamal Mabrouki, Mourade Azrour, Driss Dhiba, Yousef Farhaoui, and Souad El Hajjaji. Iot-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts. *Big Data Mining and Analytics*, 4(1):25–32, 2021.

[24] Ziteng Wang, Chun Hu, Dezhi Zheng, and Xinlei Chen. Ultralow-power sensing framework for internet of things: A smart gas meter as a case. *IEEE Internet of Things Journal*, 9(10):7533–7544, 2022.

[25] Yasuhiro Ikeda, Shouji Kouno, Akihiro Shiozu, and Katsunori Noritake. A framework of scalable qoe modeling for application explosion in the internet of things. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 425–429, 2016.

[26] Redowan Mahmud, Satish Narayana Srirama, Kotagiri Ramamohanarao, and Rajkumar Buyya. Quality of experience (qoe)-aware placement of applications in fog computing environments. *Journal of Parallel and Distributed Computing*, 132:190–203, 2019.

[27] Amulya Karaadi, Lingfen Sun, and Is-Haka Mkwawa. Multimedia communications in internet of things qot or qoe? In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 23–29, 2017.

[28] Xinlei Chen, Yulei Zhao, and Yong Li. Qoe-aware wireless video communications for emotion-aware intelligent systems: A multi-layered collaboration approach. *Information Fusion*, 47:1–9, 2019.

[29] Qiling Xu, Kai Sun, Shuai Wang, Pengpeng Chen, and Zhimeng Yin. Slora: A systematic framework for synergic interference resilience in lpwan. In *2023 IEEE 31st International Conference on Network Protocols (ICNP)*, pages 1–11. IEEE, 2023.

[30] Luoyu Mei, Zhimeng Yin, Shuai Wang, Xiaolei Zhou, Taiwei Ling, and Tian He. Ecrlora: Lora packet recovery under low snr via edge–cloud collaboration. *ACM Transactions on Sensor Networks*, 20(2):1–25, 2024.