Decentralized Federated Learning Framework for the Neighborhood: A Case Study on Residential Building Load Forecasting

Jiechao Gao, Wenpeng Wang, Zetian Liu, Md Fazlay Rabbi Masum Billah and Bradford Campbell
University of Virginia, VA, USA
{jg5ycn,ww2cg,zl4dc,mb2vj,bradjc}@virginia.edu

ABSTRACT

The fast-growing trend of Internet of Things (IoT) has provided its users with opportunities to improve user experience such as voice assistants, smart cameras, and home energy management systems. Such smart home applications often require large numbers of diverse training data to accomplish a robust model. As single user may not have enough data to train such a model, users intent to collaboratively train their collected data in order to achieve better performance in such applications, which raise the concern of data privacy protection. Existing approaches for collaborative training need to aggregate data or intermediate model training updates in the cloud to perform load forecasting, which could directly or indirectly cause personal data leakage, alongside with significant communication bandwidth and extra cloud service monetary cost.

In this paper, to ensure the performance of smart home applications as well as the protection of user data privacy, we introduce the decentralized federated learning framework for the neighborhood and show the study on residential building load forecasting application as an example. We present PriResi, a privacy-preserved, communication-efficient and cloud-service-free load forecasting system to solve the above problems in a residential building. We first introduce a decentralized federated learning framework, which allows the residents to process all collected data locally on the edge by broadcasting the model updates between the smart home agent in each residence. Second, we propose a gradient selection mechanism to reduce the number of aggregated gradients and the frequency of gradient broadcasting to achieve communication-efficient and high prediction results. The real-word dataset experiments show that our method can achieve 97% of load forecasting accuracy while preserving residences' privacy. We believe that our proposed decentralized federated learning framework can be widely used in other smart home applications as well.

CCS CONCEPTS

• **Computer systems organization** → *Neural networks*; • **Security and privacy** → **Privacy protections**.

KEYWORDS

Data Privacy, Federated Learning, Load Forecasting

ACM Reference Format:
Jiechao Gao, Wenpeng Wang, Zetian Liu, Md Fazlay Rabbi Masum Billah and Bradford Campbell. 2021. Decentralized Federated Learning Framework for the Neighborhood: A Case Study on Residential Building Load Forecasting. In *The 3rd International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things (AIChallengeIoT 21), November 15–17, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3485730.3493450

1 INTRODUCTION

Over the past years, IoT devices have been rapidly developed and frequently used in residential units and buildings all over the world. Smart home agents like Apple HomeKit, Google Home Assistant, and Amazon Alexa are connected with smart home devices such as TVs, cameras, smart lighting, and HVAC systems to achieve better user experience. As these devices collect various data streams such as energy, voice and image, which can enable better performance of smart home applications such as voice assistants, smart cameras, and home energy management systems by training robust models.

Home energy management systems (HEMS) is one of the most important smart home applications. As the increasing number of energy-consuming devices have been installed in residential units, the energy consumed by socialized miscellaneous electrical loads (MELs) has increased [10]. As a result, the increasing electricity monetary cost and harmful gas emissions to produce this amount of energy have become a severe problem to the society [18]. Among all the energy consumption in the building sector, energy waste is one of the biggest problems. The average energy waste per building is 20-30% of its total consumed energy [1]. Energy waste from residential buildings is even worse, accounting for at least 33% of the total energy consumption [4]. Solving building energy waste is a vital need to enable energy reduction.

Existing approaches for saving energy use real-time consumption data and machine learning techniques to implement load forecasting models to identify energy waste [7, 20]. However, single user may not have enough data to train such a model. For instance, if one user just move to a new place, he/she may need to spend months to collect energy data for training and years to train a robust model. In order to generate a robust load forecasting model, users intent to collaboratively train their collected energy data [11]. Traditional approaches require collecting data and transmitting the data to cloud platforms, then predicting the upcoming usage patterns using machine learning and deep learning methods [5, 12, 28]. Although cloud services can bring high computational power and gather more data from different users for training, they introduce

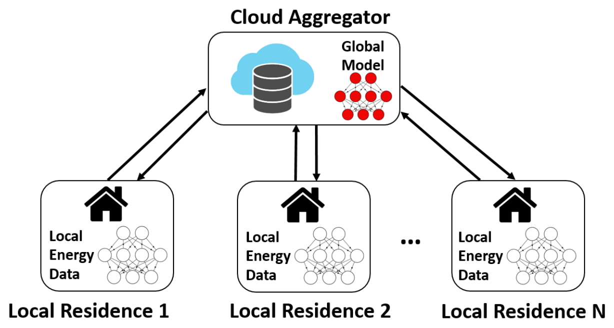

Figure 1: Federated Learning Framework.

the critical risk of potential personal data leakage. Nearly 52% of users have experienced stolen data while using cloud services [2]. As a result, the number of people willing to share their personal data fell from 41% to 31% from 2018 to 2019 [14].

To address the above issues, McMahan *et al.* [16] proposed a privacy-preserving distributed machine learning framework called federated learning (FL) for IoT devices to train a global load forecasting model while keeping the training datasets local and without sharing raw data. Figure 1 shows the structure of FL framework. It allows IoT devices to collaboratively train a load forecasting model without compromising privacy. Existing approaches [6, 19, 21, 23] present a similar framework for load forecasting. First, they train a model locally and transmit the training gradient to the cloud. Second, the cloud aggregates the gradients from different devices into a combined model in the cloud. Then, the updated model is transmitted to each local device for its own load forecasting jobs. However, such a framework still faces three problems: (1) Even though the data is stored locally, the FL framework still needs cloud service to aggregate a global model. Such a model contains all the training information from different local IoT devices, which is still vulnerable to misuse such as reconstructions of the training examples by model inversion and indirectly face the potential risk of data leakage. (2) Large-scale FL training requires significant communication bandwidth for gradient transmission, which limits the scalability of IoT devices in the FL framework. (3) Using a central server to aggregate the global model incurs a monetary cost. Since the energy data is generated consistently through time, it is necessary to update the global model with a certain frequency. To keep updating the model, cloud central server needs to be on all the time, which can cause a tremendous monetary cost. Such mechanism will decrease the participants to share their trained model and lead to a lower load forecasting performance.

In this paper, we introduce a decentralized federated learning framework for the neighborhood and show the study on residential building load forecasting application as an example. We propose PriResi, a privacy-preserved, communication efficient and and cloud-service-free load forecasting system to solve the above problems in a private residential building. First, we introduce a decentralized federated learning (DFL) framework to enable distributed edge devices to collaboratively train a model. The training gradients from each local model are broadcasted and aggregated between the smart home agents owned by each residence at a certain frequency, which removes the need of using cloud central server and reduce the possibility of data leakage and monetary cost from cloud service. Second, to further improve the communication efficiency of the DFL framework while also increasing the prediction accuracy for each resident, we propose a gradient selection mechanism based on gradient threshold selection and broadcast frequency selection. This reduces the size and number of gradients aggregated by edge devices. We evaluate the proposed PriResi energy management system on the real-world Pecan Street dataset. Experimental results show that the proposed framework can achieve 97% of load forecasting accuracy while preserving residences' privacy and our gradient selection mechanism can speedup the system running time by 19%. The contributions of this paper are summarized as follows:

1. We propose a DFL framework which allows to process all collected data locally, and the model gradients are transferred among the participants network instead of using cloud service, which reduce the possibility of data leakage and monetary cost from cloud service.
2. We proposed a new gradient selection mechanism based on gradient threshold selection and broadcast frequency selection to reduce the size and number of gradients aggregated by each smart home agent which aims to improve the communication efficiency of the DFL framework as well as the prediction accuracy for each resident. Such a method overcomes the issue of traditional methods where all gradients are combined together.
3. We conduct extensive experiments on real-world dataset to demonstrate that our proposed method can achieve 97% of load forecasting which is as good as using cloud service and in the meanwhile, preserving user privacy.

2 RELATED WORK

2.1 Load Forecasting

Various works in load forecasting majorly focus on two different aspects: aggregated household-level forecasting or device-level forecasting. Due to the fact that device-level forecasting suffers from high volatility and uncertainty of device usage that is not significant on aggregated loads, their major approaches are different. Kong et al. [13] introduced a centralized density-based clustering technique to evaluate and compare the inconsistency between the aggregated load and individual loads, then they adopted long short-term memory (LSTM) and designed a load forecasting framework for individual residential households. However, they require all data to be transmitted to a central hub, which can cause privacy concern on sensitive usage data. Luo et al. [15] proposed a HVAC energy demand prediction system based on artificial neural network, and adopted a cybersecurity framework to protect data in the fog computing environment. However, such method focuses on centralized building scenario with one HVAC system, which can't handle the residential building problem.

2.2 Energy Data Privacy

With more smart devices brings into residential homes, and the widely use of load forecasting systems, various sensors will be able to collect data of all kinds. This creates a new issue as these fine-grained data can be possibly used to identify consumers' specific activity or behavior patterns, thereby giving rise to serious privacy concerns. Only a few works in building load forecasting systems have considered data privacy [8, 15, 26]. Most of their work aims

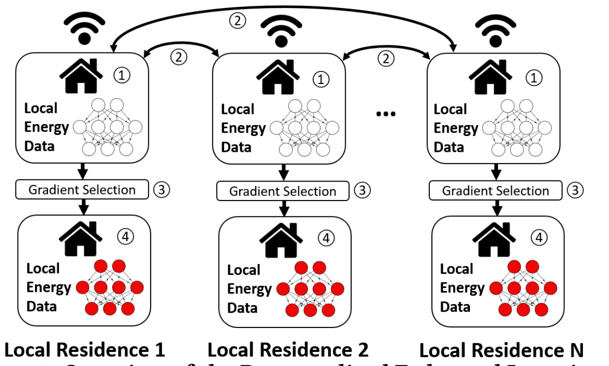

Figure 2: Overview of the Decentralized Federated Learning Framework: ① Local energy data collection and training. ② Gradient broadcasting at certain time frequency β, ③ Gradient threshold α calculation using gradient selection mechanism to generate updated model. ④ Local testing

to deal with the tradeoffs between energy data privacy and energy costs, for example, in [26], they developed an online control algorithm using Lyapunov optimization technique to balance the problem between cutting down electricity bill and keeping the privacy of load requirement and electricity bill processes. Our work differs from previous work as we maximize the energy privacy while not sacrificing the overall performance on energy reduction, where all data collected on the edge by various sensors will be processed locally, and only policy gradient that contain zero personal data is transferred in the network.

2.3 Federated Learning

A recent trend in maintaining user privacy is using federated learning approach, where each agent process its own collected data locally, and only transmit the calculated gradient to a center node. This reduces the amount of data transmission and preserves the user privacy. There have been several papers using federated learning. [25] trains the machine leaning model by distributing data across multiple edge nodes instead of sending them to a center node. Their proposed approach adaptively finds the best trade-off between local parameter update and global parameter aggregation under a given resource budget. Edge Federated Learning is also proposed to separate the process of model training into multiple mobile devices [27]. However, all previous FL frameworks require cloud central server as a global model aggregator, which can indirectly leak the local training data and require extra monetary cost.

3 SYSTEM DESIGN

3.1 System Overview

We present PriResi, a decentralized federated learning (DFL) framework for load forecasting. Our proposed DFL framework removes the requirement for cloud services. Figure 2 shows the structure of the proposed DFL framework. First, each agent collects the device load data from every IoT device deployed in its local residence and trains a load forecasting model with a default machine learning method. Second, the agents broadcast the gradients of each device at certain time frequency β, so that each agent has the gradient information from the same kind of device in other residences. Third, in the first cycle, every agent will broadcast their gradient to each other, then each agent calculates a gradient threshold α using gradient selection mechanism located in each agent for each device. If the gradient of certain device exceeds the threshold, the gradient will be selected as the aggregated gradient. If the gradient of a certain device is smaller than the top α of the total gradients, such gradient of certain device will be labeled as held gradient and not be used as aggregated gradient and will not be broadcasted in the second cycle. Finally, after each agent updates its local model by aggregating the qualified gradient for each device, the agent predicts the future power draw for each device. In the meantime, the devices are still recording load data for the next training phase, and this process happens at same interval, by default once per hour.

Specifically, our building load forecasting **problem** is as follows: *Given a multi-tenant building with multiple residents, the system should be able to accurately forecast load consumption for all residences in the building while preserving user privacy, and making efficient use of the communication channel.*

To handle such problem, we propose a residential building load forecasting system with the following three steps:

(1). For each smart home agent in each residence, the system starts with the same structured training models which allow the agent to train their own model locally for each load. The training gradients generated from each locally trained model then are recorded for each load. (Section 3.2.1)

(2). The system then decides the gradient threshold and broadcast frequency to reduce the size and number of gradients aggregated by each agent. (Section 3.2.2)

(3). Upon receiving gradients, each agent integrates the gradients to update their local model to improve the prediction accuracy for each device in the residence for load forecasting. (Section 3.2.3)

3.2 Decentralized Federated Learning

3.2.1 Local Training Process. In our system, we consider a residential building which includes N residences. Each residence n has an agent such as Google Home or Amazon Alexa which is connected to the IoT devices in certain residence n, where $n \in \{1, 2, ...N\}$. We denote A_n as the agent in the system which represents that it's residence n. For each agent A_n, we have the same default training model initially, such as Long Short Term Memory (LSTM). We denote D_{Xn} as different IoT devices in different residences, X refers to the type of certain IoT device. Since each IoT device D_{Xn} has their own local dataset (i.e., sensing time-series data from IoT nodes), we train the model separately for each device on the connected agent.

For example, in Figure 2, the TV in residence one, defined as D_{TV1}, TV in residence two, defined as D_{TV2}, the lighting in residence one, defined as D_{light1}, etc. We train the model for each device in each residence on the connected agent locally and separately. Thus every device has a certain gradient, for example, a gradient G_{TV1} is calculated for the TV in residence one in a certain time period. In each resident's home, an agent will record the gradient for all the resident's devices. As an instance, for residence one, the agent A_1 has the gradient information G_{TV1}, G_{light1}, G_{HVAC1} and all the other devices that are connected to A_1.

Given a local dataset recorded by an IoT device D_{Xn}, our goal is to predict the future energy consumption for this certain device for the following hour. An LSTM model is used in training set to learn the usage pattern of the device, and the testing set is used to predict the estimated energy consumption after the gradient aggregation. For each device D_{Xn}, we first predict the energy consumption V_{Xn}

in every minute for the next hour. Then, we calculate and record the gradient for each device D_{Xn}.

3.2.2 *Gradient Selection Mechanism.*

Traditional large-scale FL training requires significant communication bandwidth to broadcast gradients in high frequency. However, in our DFL framework, the gradients are broadcasted across all agents instead of cloud servers. This in effect mitigates the requirement of frequent updates and large communication bandwidth problem.

To reduce the high-bandwidth requirement, in our DFL approach we first apply gradient quantization to quantify gradients to low-precision values. Then we set α as the threshold of the gradient to determine which gradient can be aggregated to other agents. For gradient quantization, since we are using LSTM as the default training model in the DFL framework, the fully connected (FC) layer in LSTM can be defined as $out = f(W * in + b)$, where in is the input, b is the bias, W is the weight, f is the nonlinear mapping, and out is the output. So for each specific neuron i, the above equation can be simplified as $out_i = ReLU(\sum_{j=0}^{n-1} W_{ij} in_j)$ where ReLU [17] is the activation function in LSTM model, and i, j represent the position information of the weight element in the weight matrix W. In order to quantize the gradients to low-precision values, we compress the corresponding weight matrix into a sparse matrix, and hence the gradient is the derivative of the sparse weight matrix. The corresponding formula is given as follows:

$$out_i = ReLU(\sum_{j \in X_i \cap Y} Sparse\left[I_{ij}\right] in_j) \tag{1}$$

where $\sum_{j \in X_i \cap Y} Sparse\left[I_{ij}\right]$ represents the compressed weight matrix. Such a method reduces the size of the gradient by sparsing the weight matrix W to reduce communication bandwidth requirement.

For the number of aggregated gradient, after each agent A_n trains the model locally for each device D_{Xn}, the agent will broadcast the calculated gradients to the other agents labeled with D_{Xn}. Since each agent has the gradient value of each device from all the residences after the broadcast, the top α of the total gradients can be determined respectively. In order to improve the communication efficiency of the DFL framework as well as the prediction accuracy for each resident, we must ensure that there is no information loss in the above scheme. In order to prevent the information loss, there are two circumstances for each gradient. First, if the gradient of a certain device is larger than or equal to the top α of the total gradients, it will be selected as aggregated gradients. Such a device will use all the selected gradients as aggregated gradients to update their model. Second, if the gradient of a certain device is smaller than the top α of the total gradients, such gradient of certain device will be labeled as held gradient and not be used as aggregated gradient. Such a device will use all the selected gradients as aggregated gradients plus their own held gradient to update their model. In this process, only in the first cycle each agent will all broadcast their gradient, from second cycle and further iteration, only the gradient of a certain device is larger than or equal to the top α of the total gradients will be broadcasted. In our experiments, we will determine the hyperparameter α that can achieve the best result.

To avoid high frequency gradient broadcasting, we set β hours as the gradient broadcast rate to reduce the frequency of broadcasting gradients. In our experiment we will determine the hyperparameter β that has the best result.

3.2.3 *Gradient Aggregation.*

After each agent A_n determines the selected gradients from other agents for each device D_{Xn}, the agent will update the model with such gradients G_{Xn}. To do so, we use DSGD for iterative updates, and the loss function to be optimized is defined as follows:

$$F(w) = \frac{1}{D_{Xn}} \sum_{n \in D_{Xn}} f(n, w) \tag{2}$$

where $F(w)$ is the loss function for the updated model, $f(n, w)$ is the loss function for the previous model and w are the weights of the model.

In the gradient aggregate phase, the agents use Federated Averaging (FedAVG) algorithm [26] to obtain an updated model w_{t+1} for the next iteration, thus we have:

$$w_{t+1} = w_t - \eta \frac{1}{Nb} \sum_{n=1}^{N} \sum_{x \in B_n, k} \nabla f(x, w_t) \tag{3}$$

where η is the learning rate, B_n, k is the data sample for the k_{th} round of training, and each local dataset size of b.

All the collaborated agents repeat the above process until the model reaches convergence. Then we use the updated model to predict the energy consumption for the certain device for the next hour.

4 PERFORMANCE EVALUATION

4.1 Datasets and Experiment Settings

4.1.1 *Energy Consumption.*

We apply the proposed framework PriResi to the real-world Pecan Street dataset [3] for performance demonstration. The Pecan Street dataset contains minute-level energy consumption for home appliances such as bedrooms lights, dryers, TVs, HVAC, etc. in 1641 residences in Texas from 2013 to 2017. Since the recorded data has some missing data points, also the devices in each residence are not exactly the same, we randomly select 100 residences that have the lights, TVs and HVAC load records as our default experiment dataset. In our experiment, we first use 80% of the energy consumption dataset as the training set to calculate the gradients and aggregate the gradients from other agents to get an updated model. Second, we use the other 20% of the dataset as testing data. For the RL hyperparameters, we set the learning rate as 0.001, discounted rate as 0.9, memory capacity as 2000, and target replace iteration as 100. We also experiment with different numbers of residences joining the system and different system running days to test the performance and robustness of our PriResi system.

4.2 Compared Methods

To evaluate PriResi, we choose the following three methods for comparison. We train the models with the same Pecan Street dataset with parameters as mentioned in the individual papers.

(1) Smart Energy Management and Demand Reduction [22] (SEM) studies the usage behavior of consumers from their historical data and predicts the demand for energy every hour for the individual consumer using seasonal auto regressive integrated moving average (SARIMA).

(2) Cloud forecasting system for monitoring and alerting [9] (CFS) proposes an efficient cloud-based energy information management system home appliances that monitors the load of residences, predict energy consumption based on a machine learning method called SARIMA-MetaFA-LSSVR.

(3) Forecasting Multi-Appliance Usage and Energy Management [24] (FMA) developes a prediction algorithm based on a graphical model

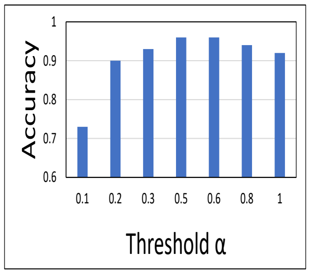

Figure 3: The accuracy of DFL with different threshold α.

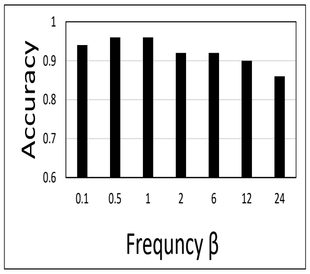

Figure 4: The accuracy of DFL with different broadcast frequncy β.

that captures the everyday habits and the interdependency between appliances by exploiting their periodic features.

4.3 Performance Metrics

(1) *Hyperparameters selection.* We measure the DFL framework hyperparameter α and β to determine the threshold for gradient selection and broadcast frequency which have the best accuracy.

(2) *Prediction accuracy.* To measure the prediction accuracy of energy consumption, we measure the prediction accuracy as below: $Ac_n = 1 - \frac{|V_n - RV_n|}{RV_n}$ where Ac_n is the prediction accuracy of n^{th} prediction, V_n is the predicted value of n^{th} prediction and RV_n is the true value of n^{th} prediction.

(3) *Communication efficiency.* To analyze the communication efficiency of the system, we calculate the running time speedup percentage as below: $Speedup = (T_1 - T_n) \div T_1$ where T_1 is the running time for system 1^{st} time running, T_n is the running time for system n^{th} time running. Since we fix the communication overhead of each round in the gradient selection and aggregation section, so we can compare the running time of the system to compare the communication efficiency.

4.4 Experimental Results

4.4.1 Hyperparameters Selection of the DFL Framework. Figure 3 shows the accuracy of the DFL framework with different threshold α. We use $\alpha \in \{0.1, 0.2, 0.3, 0.5, 0.6, 0.8, 1\}$ to determine the best threshold of the proposed framework. We observe that $\alpha = 0.5$ and 0.6 have the best prediction accuracy for load forecasting, which means 50% or 60% of the total gradients for each device should be aggregated to update the model. Therefore, to achieve a good trade-off between the communication efficiency and accuracy, we choose $\alpha = 0.5$ as the best threshold of our DFL framework.

Figure 4 shows the accuracy of DFL framework with different broadcast frequencies β. We employ $\beta \in \{0.1, 0.5, 1, 2, 6, 12, 24\}$ to adjust the best frequency of the proposed framework. We observe that $\beta = 0.5$ and 1 have the best prediction accuracy for load forecasting, which means the gradients should be broadcasted every 30 minutes or 1 hour. Due to the same reason as in Figure 3, we choose $\beta = 1$ as the best frequency of our DFL framework.

4.4.2 Load forecasting Accuracy. Figure 5 shows the cumulative distribution function (CDF) of the load forecasting result. The result follows: SEM<FMA<CFS≈PriResi. The reason is that, for SEM, it uses SARIMA as load prediction method. Such method can only deal with short-term prediction, which is good at the time when the load is in certain pattern in a short term. However, for each time the load can't be defined as certain pattern, the prediction accuracy of SEM will be much lower than average. So the accuracy of SEM is lower than the others. For FMA, it uses the graphical model to capture the load changing for each device. Such method can still achieve high prediction accuracy when facing unexpected load patterns. However, FMA only considers the devices in one resident, which will face the shortage of data. If a certain residence only has the energy data from a device within a short time range (i.e., a month), the accuracy for that certain device will drop dramatically. So the accuracy of FMA is lower than CFS and PriResi. For CFS, it applies cloud service and gathered all the data from each device into the cloud for prediction which can improve the average prediction accuracy for all devices. Their proposed method SARIMA-MetaFA-LSSVR can also handle the unexpected load for different devices. For PriResi, we apply FDL framework with the gradient selection mechanism, which can enhance not only the average prediction accuracy, but the accuracy for each device in each resident.

Figure 6 shows the load forecasting accuracy in a day at different times. The result follows: SEM<FMA<CFS≈PriResi due to the same reason as explained in Figure 5. We also observe that, the accuracy from 2 AM to 6 AM and from 12 PM to 16 PM are higher than the other time in a day. The reason is that, in such time frame, residences usually have the same energy usage patterns for each device. From 8 AM to 10 AM and evening time, the energy usage in different residences is vary depending on the date.

Figure 7 shows the prediction accuracy while we accumulatively train the DFL framework with different number of days. We set the number of residences as 100 in this experiment. The result follows: SEM<FMA<CFS≈PriResi due to the same reason as explained in Figure 5. Since we accumulatively train the DFL framework, for each hour, each agent has the aggregated gradient for each device. The updated gradient will be used for the next training period, which will improve the prediction accuracy over time. On the other hand, from day 1 to day 30, the growth of the accuracy is higher than from day 70 to 100. The reason is that, the aggregated gradient tends to approach the best value for the load forecasting.

Figure 8 shows the prediction accuracy with different number of residences participating the DFL framework. We set the number of days as 365. For the number of residences under 100, the result follows: SEM<FMA<CFS≈PriResi due to the same reason as in Figure 5. For the number of residences above 100, the result follows: SEM<FMA<CFS<PriResi. The reason is that, for SEM, FMA and CFS, they use all the gradients or data to train the model together. Such methods can indeed improve the average accuracy when the number of total participants is small. When the number of residences goes up, the number of different kinds of load patterns also goes up. In this case, using all the gradients or data to train the model may cause prediction accuracy drop to some devices. For PriResi, we proposed gradient selection mechanism which allows only the qualified gradients (i.e., larger than certain threshold) to be aggregated as the updated training model, which can lower the possibility of prediction accuracy drop.

Figure 9 and Figure 10 compare different sharing methods in the DFL framework with different number of days and different number of residences respectively. We try sharing the real data, sharing all the gradients and sharing gradient with our proposed selection mechanism. Figure 9 has the same experiment as Figure 7

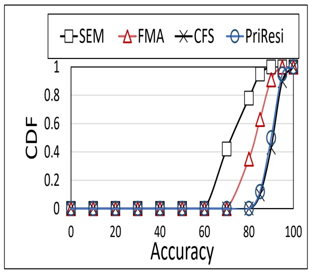

Figure 5: CDF for Load forecasting accuracy.

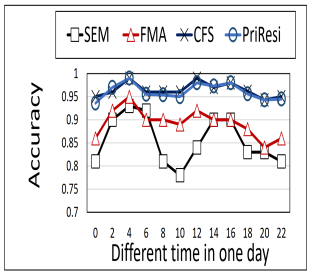

Figure 6: Load forecasting accuracy in a day.

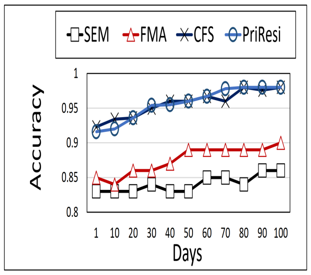

Figure 7: Prediction accuracy with different number of days.

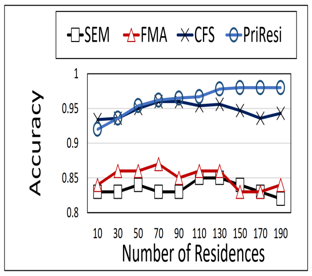

Figure 8: Prediction accuracy with different number of residences.

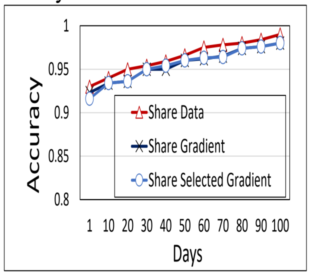

Figure 9: Sharing method in DFL with different number of days.

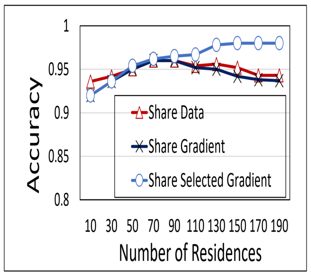

Figure 10: Sharing method in DFL with different number of residences.

and Figure 10 has the same experiment as Figure 8. For Figure 9, the result follows: Share Data≈Share Gradient≈Share Selected Gradient due to the same reason as explained in Figure 7. For Figure 10, the result follows: Share Data≈Share Gradient≈Share Selected Gradient for the number of residences under 100, due to the same reason as explained in Figure 5. For the number of residences above 100, the result follows: Share Data≈Share Gradient<Share Selected Gradient due to the same reason as explained in Figure 8.

4.4.3 Communication Efficiency Comparison. Figure 11 shows the running time speedup percentage versus different number of training days to compare the communication efficiency. We set the number of residences as 100 in this experiment. We observe that the running time speedup percentage of DFL with gradient selection mechanism can achieve 19% at 100 days comparing with the 1st time system running which is much higher than without gradient selection mechanism. The reason is that gradient selection mechanism can reduce the number of gradients exchanged between the smart home agents and speed up the local model convergence time.

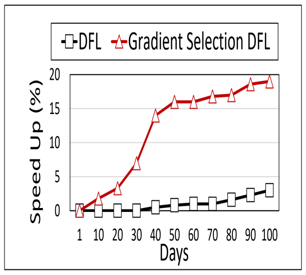

Figure 11: Communication efficiency with different number of days.

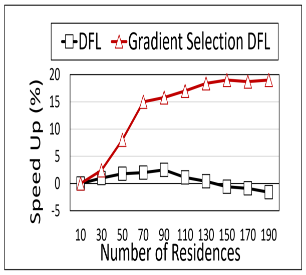

Figure 12: Communication efficiency with different number of residences.

Figure 12 shows the running time speedup percentage versus different number of residences to compare the communication efficiency. We set the number of days as 365 in this experiment. We observe that the running time speedup percentage of DFL with gradient selection mechanism can achieve 19% with 90 residences comparing with the 1st time system running which is much higher than without gradient selection mechanism due to the same reason as mentioned in Figure 11. However, for DFL without gradient selection mechanism, we observe that the running time is slower than the 1st time system running after the number of residences is larger than 130. The reason is that, number of residences in the system can directly increase the number of gradients. Without the gradient selection mechanism, the DFL model will calculate all the gradients that are broadcasted from each residence, which make the local model convergence time longer.

5 CONCLUSION

In this paper, we propose PriResi, a privacy-preserved, communication efficient and cloud-service-free federated learning system to achieve accurate load forecasting in a residential building. First, we propose a DFL framework that doesn't require cloud service to aggregate gradients and update the prediction model. Such proposed framework is privacy-preserved and can still achieve high prediction accuracy while saving the monetary cost from cloud service. Second, we propose a gradient selection mechanism which includes three steps: gradient quantization, selecting aggregated gradient and gradient broadcasting frequency to solve the communication efficient problem. Experimental results validate that our proposed system can achieve 97% prediction accuracy which is comparable to cloud service without compromising residences' private data leakage. We also achieve higher accuracy when facing more participants in our system. We believe that our proposed decentralized federated learning framework can be used not only in energy load forecasting scenario, but potentially in other smart home applications as well. In our future work, we will further study how to utilize our decentralized federated learning framework to solve collaborative training problems in privacy-sensitive scenarios.

6 ACKNOWLEDGEMENT

This work was supported by the Department of Energy (DOE) under grant DE-EE0008225. We wish to thank the anonymous reviewers for their detailed comments and feedback.

REFERENCES

[1] [Accessed in JUL. 2021]. Energy Star: Save Energy. *https://www.energystar.gov/buildings/* ([Accessed in JUL. 2021]).

[2] [Accessed in JUL. 2021]. Help Net Security. *https://www.helpnetsecurity.com/2020/01/28/accessing-cloud-services/* ([Accessed in JUL. 2021]).

[3] [Accessed in JUL. 2021]. Pecan Street dataset. *https://www.pecanstreet.org/dataport/about/* ([Accessed in JUL. 2021]).

[4] [Accessed in JUL. 2021]. RESIDENTIAL BUILDINGS FACTSHEET. *http://css.umich.edu/factsheets/* ([Accessed in JUL. 2021]).

[5] Y. Agarwal, B. Balaji, and R. Gupta. 2010. Occupancy-driven energy management for smart building automation. In *Proc. of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*.

[6] U. Matchi Aivodji, S. Gambs, and A. Martin. 2019. IOTFLA: A secured and privacy-preserving smart home architecture implementing federated learning. In *2019 IEEE Security and Privacy Workshops (SPW)*.

[7] M. Al Faruque and K. Vatanparvar. 2015. Energy management-as-a-service over fog computing platform. *IEEE internet of things journal* (2015).

[8] H. Chang, W. Chiu, H. Sun, and C. Chen. 2018. User-centric multiobjective approach to privacy preservation and energy cost minimization in smart home. *IEEE Systems Journal* (2018).

[9] J. Chou and N. Truong. 2019. Cloud forecasting system for monitoring and alerting of energy use by home appliances. *Applied Energy* (2019).

[10] US DOE. 2015. An assessment of energy technologies and research opportunities. *Quadrennial Technology Review. United States Department of Energy* (2015).

[11] Yaochen Hu, Di Niu, Jianming Yang, and Shengping Zhou. 2019. FDML: A collaborative machine learning framework for distributed features. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2232–2240.

[12] A. Javed, H. Larijani, A. Ahmadinia, and D. Gibson. 2016. Smart random neural network controller for HVAC using cloud computing technology. *IEEE Transactions on Industrial Informatics* (2016).

[13] Z. Kong, W.and Dong and Y. Jia. 2017. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Transactions on Smart Grid* (2017).

[14] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang. 2019. Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. *IEEE Transactions on Industrial Informatics* (2019).

[15] X. Luo, L. Oyedele, and A. Ajayi. 2019. Development of an IoT-based big data platform for day-ahead prediction of building heating and cooling demands. *Advanced Engineering Informatics* (2019).

[16] B. McMahan, E. Moore, D. Ramage, and S. Hampson. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*.

[17] P. Petersen and F. Voigtlaender. 2018. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks* (2018).

[18] Bipartisan Policy. 2020. Annual Energy Outlook. (2020).

[19] Y. Saputra, D. Hoang, and E. Nguyen, D.and Dutkiewicz. 2019. Energy demand prediction with federated learning for electric vehicle networks. In *2019 IEEE Global Communications Conference (GLOBECOM)*.

[20] M. Soliman, T. Abiodun, and T. Hamouda. 2013. Smart home: Integrating internet of things with web services and cloud computing. In *2013 IEEE 5th international conference on cloud computing technology and science*.

[21] A. Taik and S. Cherkaoui. 2020. Electrical Load Forecasting Using Edge Computing and Federated Learning. In *2020 IEEE International Conference on Communications (ICC)*.

[22] R. Tom, S. Sankaranarayanan, and J. Rodrigues. 2019. Smart Energy Management and Demand Reduction by Consumers and Utilities in an IoT-Fog-Based Power Distribution System. *IEEE Internet of Things Journal* (2019).

[23] N. Tran and A. Bao, W.and Zomaya. 2019. Federated learning over wireless networks: Optimization model design and analysis. In *2019 IEEE INFOCOM*.

[24] N. Truong, J. McInerney, and L. Tran-Thanh. 2013. Forecasting multi-appliance usage for smart home energy management. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*.

[25] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan. 2019. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. *IEEE Journal on Selected Areas in Communications* (2019).

[26] L. Yang, X. Chen, J. Zhang, and H. Poor. 2014. Optimal privacy-preserving energy management for smart meters. In *2014-IEEE INFOCOM*.

[27] Y. Ye, S. Li, F. Liu, Y. Tang, and W. Hu. 2020. EdgeFed: Optimized Federated Learning Based on Edge Computing. *IEEE Access* (2020).

[28] K. Zhou, C. Fu, and S. Yang. 2016. Big data driven smart energy management: From big data to big insights. *Renewable and Sustainable Energy Reviews* (2016).