

# Exploring Lightweight Federated Learning for Distributed Load Forecasting

Abhishek Duttagupta, Jin Zhao, Shanker Shreejith

Electronic & Electrical Engineering

Trinity College Dublin, Ireland

Email: {aduttagu,zhaoj6,shankers}@tcd.ie

**Abstract**—Federated Learning (FL) is a distributed learning scheme that enables deep learning to be applied to sensitive data streams and applications in a privacy-preserving manner. This paper focuses on the use of FL for analyzing smart energy meter data with the aim to achieve comparable accuracy to state-of-the-art methods for load forecasting while ensuring the privacy of individual meter data. We show that with a lightweight fully connected deep neural network, we are able to achieve forecasting accuracy comparable to existing schemes, both at each meter source and at the aggregator, by utilising the FL framework. The use of lightweight models further reduces the energy and resource consumption caused by complex deep-learning models, making this approach ideally suited for deployment across resource-constrained smart meter systems. With our proposed lightweight model, we are able to achieve an overall average load forecasting RMSE of 0.17, with the model having a negligible energy overhead of 50 mWh when performing training and inference on an Arduino Uno platform.

*Index Terms*—Federated learning, deep neural networks, Non - i.i.d distribution, data heterogeneity

## I. INTRODUCTION

Electricity demand forecasting is a crucial function in the energy industry to allow both producers and distribution system operators to optimize the generation and distribution of electricity in real-time [1]. It aims to balance the supply and demand curve and maintain the stability of the electric grid across the entire network of producers and consumers. Smart energy meters [2] play a vital role in load forecasting by integrating communication capabilities on top of their ability to monitor energy consumption accurately [3]. Smart meters transmit demand data from consumers at specified intervals to distribution system operators, allowing them to observe the data in real time, make predictions on future demand, and adapt the grid to respond to this expected demand. This data-driven approach has enabled electric grid operators to optimize energy generation and storage to adapt to load peaks from the supply-side, and at the same time achieve demand-side load balancing through novel schemes like dynamic pricing tariff rates to achieve supply-demand equilibrium [4].

Researchers have proposed different algorithms for load forecasting that primarily enables dynamic energy pricing and usage optimization [5]. In [4], the authors apply gradient-boosted regression trees on the Spanish Energy dataset to minimize prediction error for a day-ahead forecast, showing the effectiveness of ML models in load forecasting. However, load forecasting is a challenging task due to the stochastic nature of customer usage profiles. In [6], the authors show that the inclusion of spatial and temporal parameters can have a significant impact on the accuracy of load prediction. Area-based clustering techniques have also been explored to group

customers with similar electric energy consumption patterns in a spatiotemporal window to improve the accuracy of load forecasting [7], [8].

Hybrid models combining classical and deep learning methods have also demonstrated promising results in load forecasting. In [9], the authors developed a hybrid model utilizing ARIMA, logistic regression, and deep neural networks to predict peak load days with a 70 % accuracy. Other studies reported in the literature have proposed models based on Convolutional neural networks (CNNs) with fuzzy time series, achieving superior results compared to traditional models [10], [11]. Using complex models such as a standalone Long Short Term Memory (LSTM) model [12], a combination of LSTM and CNN [13], and Gated Re-current Units (GRU) combined with CNN [14] have also shown to achieve similar prediction performance. The use of rolling updates and Bi-LSTMs was shown to reduce the computational time over other deep learning methods, while also improving the accuracy of load forecasting [15].

While most deep learning models have been primarily focused on improving forecasting performance, privacy concerns have risen with the centralized learning approach used by load forecasting schemes. In the case of centralized models, data from smart meters of consumers is monitored and accumulated at locations within the distribution grid, which is used by the model to predict peak loads or other learned parameters about the demand in specific clusters. It has been shown that non-intrusive analysis of granular load curves can provide information about appliances, usage patterns and hence personal details about consumers [16], [17]. Privacy-preserving machine learning (ML) through federated learning (FL) is a promising scheme to address the challenges of centralized learning on sensitive data [17]. With FL, individual models can be trained collaboratively and locally, removing the need to accumulate and share sensitive data across the network. While FL has shown to be effective in other domains [18], [19], its application to load forecasting is very limited. In [20], the authors propose a combination of LSTM and dense neural network within an FL framework for load forecasting and utilize clustering to improve the model's prediction performance. However, the root mean squared error (RMSE) measure of the federated approach was higher than the centralized learning technique. The work in [21] achieved better prediction performance using a global generalized model aggregated from locally trained LSTM models at the consumer end, and by incorporating spatiotemporal factors into the framework. However, the complexity of local and global models as well as their high energy consumption is a barrier to their widespread

deployment in constrained devices such as smart meters.

In this work, we apply an FL framework for short-term load forecasting using the openly available smart meter energy consumption dataset for London households [22]. The key objective is to show that comparable accuracy to state-of-the-art methods in load prediction can be achieved using lightweight deep neural network (DNN) models within an FL framework to restrict the need for granular data sharing and aggregation across the network. To replicate real-world non-independent and identical distribution (i.i.d.) conditions, we consider limited random connections for each round of FL, simulating scenarios where not all devices are participating in the learning process. The main contributions of this paper are summarized below.

- 1) A lightweight feed-forward model within an FL framework to achieve identical and/or superior short-term load forecasting performance, even while considering limited device connections within the cluster (non i.i.d condition).
- 2) Combining the FL flow with clustering to achieve a better global representation within each cluster, leading to improved forecasting performance.
- 3) Evaluate our approach and demonstrate that lightweight models are sufficient to achieve similar load forecasting performance when compared to other complex models.
- 4) Quantify the overheads of the proposed lightweight models on a lightweight microcontroller that mimics an actual smart meter device.

The rest of this paper is organized as follows: Section II discusses the proposed federated learning framework, and Section III illustrates the employed methodology in detail; Section IV presents the case study performed with the results analysis; and conclusions are drawn in Section V.

## II. LIGHT-WEIGHTED FL-BASED LOAD FORECASTING SCHEME

### A. Federated Learning Algorithm

In a traditional centralized scenario 1, energy consumption data from participating household smart meters is sent to a cluster head (typically at a substation level), where the global server performs model training and inference. In contrast, with an FL setup 2, each household performs its own edge training of local models based on their individual consumption data. These updated models are then sent to the global server at the substation for aggregation, creating the global model. The global model is subsequently shared with the households for evaluation. The entire cycle keeps repeating, allowing the local and global model to evolve based on changes in individual consumers' consumption and generation.

The federated algorithm flow 1 used for our process is derived, from the work in [23]. The algorithm begins by selecting a set of user devices ( $N=0,1,2,\dots,n$ ) from the total device set ( $M$ ) and initializing an initial global model ( $W$ ), which is then passed to these devices. Training is performed in parallel on these selected devices using their local data ( $P$ ), considering batch sizes ( $B$ ), learning rate ( $\eta$ ), and local epochs ( $E$ ). After training, the updated models ( $w$ ) from these devices are shared with the substation, where federated averaging occurs for server aggregation, resulting in a global model ( $W$ ). The global model is then shared back with all the devices,

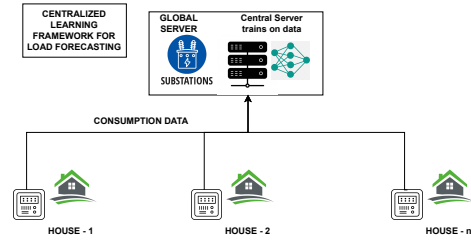


Fig. 1: Traditional centralized setup for load forecasting.

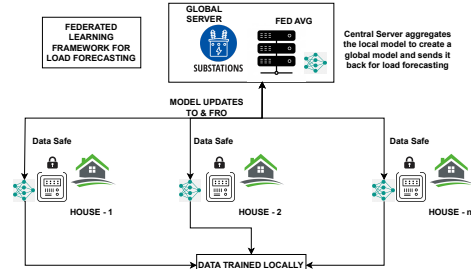


Fig. 2: Federated learning setup for load forecasting.

### Algorithm 1 Federated learning algorithm

---

**Require:** Total device set  $M$ , selected device set  $N$ , batch size  $B$ , learning rate  $\eta$ , local epochs  $E$

**Ensure:** Global model  $W$

- 1: Initialization: Randomly initialize global model  $W$
- 2: **while** not converged **do**
- 3:   **for** each device  $n$  in selected device set  $N$  **do**
- 4:     Retrieve local data  $P$  on device  $n$
- 5:     Train local model  $w_n$  on data  $P$  for  $E$  epochs with batch size  $B$  and learning rate  $\eta$
- 6:     Share local model  $w_n$  with central server
- 7:   **end for**
- 8:   Aggregate models  $w_n$  from all devices on a central server using federated averaging
- 9:   Update global model  $W$  as the average of all local models  $w_k$
- 10:   Share global model  $W$  with all devices
- 11: **end while**

---

which update their models accordingly, and the process is repeated until convergence is reached. By Convergence, we refer to the point where no noticeable improvement in the global loss function (GLF) occurs with increasing federated training rounds, indicating that the GLF has reached a minimum.

Adopting the representation in [24] for our case, an FL cluster with  $n$  clients communicating with the global server(substation for our case), if each client has a  $k$  fixed number of data points in its respective dataset, the input/output relation for a given local dataset  $S$  for a client  $i$  can be represented by the equation eq. 1. Here  $x$  and  $y$  represent the input and output vectors, respectively.

$$S_i = \{(x_i, y_i)\}_{i=1}^{k_i} \quad (1)$$

The learning process can hence be captured using the model  $f(\cdot; \theta)$  that maps an input  $x$  to the predicted label  $y$ . Here,  $\theta$  represents the trainable parameter of the model  $f$ . Assuming a loss function of  $L$  for this model capturing the error in the prediction  $f(x; \theta)$  given the true label  $y$ , the local objective of the client  $i$  is given by the equation eq. 2 and the global server

objective in an FL setting can be represented by the equation eq. 3.

$$L(\theta; \mathcal{S}_i) = \frac{1}{k_i} \sum_{i=1}^{k_i} l(f(x_i; \theta), y_i) \quad (2)$$

$$\arg \min_{\theta} L(\theta) = \frac{1}{n} \sum_{i=1}^n L(\theta; \mathcal{S}_i) \quad (3)$$

### B. Performance Metrics

With errors varying according to household average consumption level, MAE (calculated using eq. 4) and RMSE (given by eq. 5) are not ideal metrics to accurately assess short-term load forecasting at small energy consumption units. Mean absolute percentage error (MAPE, given by eq. 6) provides a more reliable measure by capturing the absolute percentage deviation between predicted and actual values. Using MAPE with MAE and RMSE allows for a comprehensive evaluation and comparison of prediction accuracy independent of a household's average consumption.

$$MAE = \frac{1}{N} \sum_{i=1}^N |X(i) - Y(i)| \quad (4)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N [X(i) - Y(i)]^2} \quad (5)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{X(i) - Y(i)}{X(i)} \right| \times 100 \quad (6)$$

As before, in the case of equations eq. 4 to eq. 6, X represents the actual value and Y represents the predicted value, with N as the number of data points.

## III. DISTRIBUTED LOAD FORECASTING METHODOLOGY

For our simulation and testing, we utilized the smart meter energy consumption dataset from London Households [22]. This dataset comprises energy consumption data recorded from smart meters installed in 5,567 households across London. The data covers the period from November 2011 to February 2014, with half-hourly energy consumption data, containing four columns: a unique tag, tariff type (standard or dynamic pricing), time-stamp, and half-hourly electric consumption (in kWh).

First, we examined the heterogeneity of the consumption data and analyzed it to understand fluctuations in the energy consumption patterns of consumers across the year and to determine any seasonal variations or long-term trends. Figure 3 illustrates the monthly total consumption trend for a random household across the year, indicating that January, February, and December tend to have higher consumption compared to other months. Similarly, figure 4 shows the daily consumption of one high-consumption consumer and one medium-consumption consumer, capturing the significant variations and heterogeneity present in the dataset. We use the average consumption for each household across the entire dataset to identify outliers with very high or very low average half-hourly

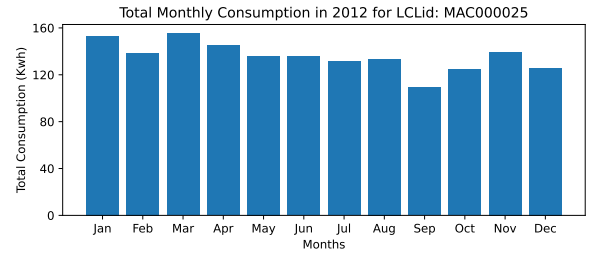


Fig. 3: Energy profile yearly trend of a random household

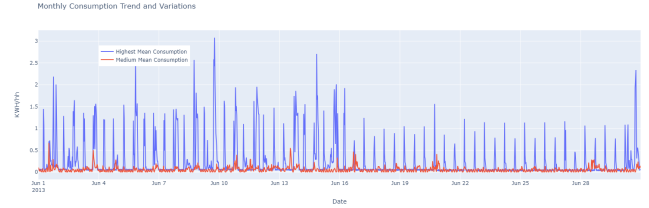


Fig. 4: Comparison of daily consumption over a month: high consumption household vs average consumption household

consumption. Based on our observations, households with an average energy consumption per half-hourly (hh) interval less than 0.09 kWh/hh (nearly unused) or greater than 1.35 kWh/hh are marked as outliers and filtered out of the dataset. Eliminating these outliers reduces the number of households from 5547 to 4672 and removes any bias that may be incurred when training the model with the outlier data points.

### A. Clustering of smart meter-based energy consumption data

Clustering the training data has shown to provide better generalisability for deep learning models when used for load prediction [7], [8]. With federated models, clustering is particularly effective as it allows clients to be organized into smaller clusters based on similar consumption and their locality. This also improves the aggregation phase in the substation where local models from similarly performing households have a better probability of achieving an unbiased aggregation to arrive at the new global model. For our approach, we use statistical information about energy consumption to organize households into 18 groups using K-Means clustering. We use mean energy consumption, median consumption, total energy consumption, maximum recorded consumption, and minimum recorded consumption parameters to guide the clustering process, similar to the approach in [21]. Clustering also takes into account the area locality for the grouping to ensure that they can be grouped under a global server that serves this area (e.g., a substation). So, in our setup, each cluster would indicate an area within London, comprising households of similar electric consumption levels with a substation (global server), feeding to all these households. Figure 5 illustrates our constructed framework up to the cluster level.

### B. Federated Training and Testing Data sets Creation

The entire dataset, indexed by household tags, is subsequently partitioned into training and test data. For every household, data from Nov 2011 to Feb 2013 was used for training and Validation purposes, while, the data from Feb 2013 to Feb 2014, was used for testing purposes, thereby achieving a 60:40 split between training and test data.

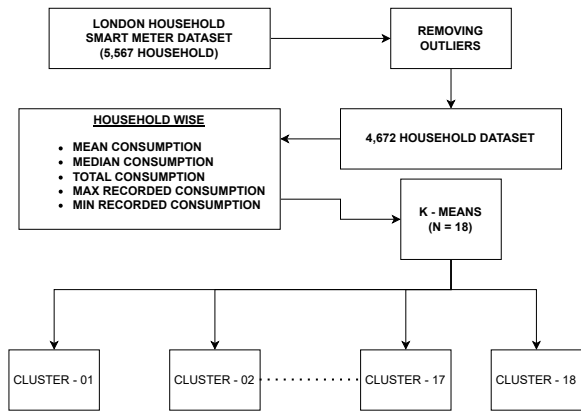


Fig. 5: Top to Cluster level Simulation Architecture

1) *Input Window and Output Prediction:* To account for the time-series nature of load forecasting, we use a sliding window scheme to capture the short history of recent consumption data, which forms the input to the deep learning model. For our experiments, we evaluated different window sizes using training error at a fixed number of epochs to arrive at a size of 336, which captures an entire week of consumption at a half-hour resolution. We infer that by considering a longer time span that includes weekdays and weekends, we can capture and analyze the trends and patterns that emerge throughout a complete week.

2) *Model architecture and TensorFlow Federated Wrappers:* For building our training and inference flows (simulation), we utilise the Python-based TensorFlow-Federated (TFF) library and its associated frameworks. TFF wrapper converts the windowed input/output data into a Python dictionary for use with the TFF framework. For our evaluation, we simulate the learning of each individual household and its aggregation at the cluster using the TFF libraries. Unique tags associated with each household are used to access individual data streams and clusters during the training and evaluation phase.

For our model, we chose to evaluate feed-forward neural networks for their simplicity and computational efficiency for each of the clients. After evaluating multiple architectures, we settled on a 4-layer model with 2 hidden layers. The model follows [16,8,4,1] shape with rectified linear unit (ReLU) activation functions at each layer. We used RMSE and MAE as the loss functions and metrics for our training and all three (RMSE, MAE and MAPE) for evaluation. The learning rate was set to 0.01 at each client site. The final model architecture has 5569 trainable parameters, making it ideal for training and performing inference on low-cost microcontroller-based computing nodes (like smart meters) while having sufficient parameters to learn crucial patterns and trends in the data stream.

#### IV. CASE STUDY

##### A. Case Settings

We use the following hyper-parameters for training the model: batch size = 12, global server learning rate = 1 and number of federated rounds = 20. We present a selection of results from individual clusters representing the spectrum of consumption levels that are available in the dataset in addition

to the aggregate performance across all clusters. For every cluster, two types of households are selected for evaluating our model performance:

- 1) Households that received all global model updates.
- 2) Households that did not receive some updates or all updates during training.

This differentiation factors in non-i.i.d conditions that may be present in a real-world condition. Additionally, within the group of households that did not receive some updates, we further divided them into inter-cluster examples, including high-consumption and low-consumption examples for observing how the forecasting accuracy's varies across different households.

##### B. Results

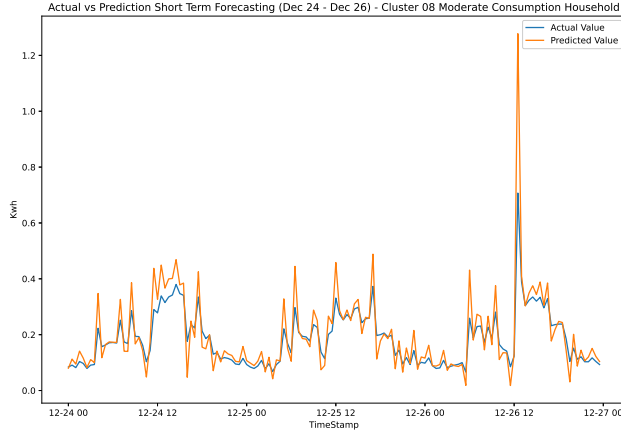
Table I presents the RMSE values averaged across different types of households within a specific cluster from our experiments. It can be observed that the model offers better (average) prediction performance for households that received all the updates (i.i.d condition) compared to those that missed out on some of the updates. Additionally, this demonstrates that in the worse scenario when a house skip updates, the approach is able to provide reasonably accurate predictions.

We also examined the performance across festive periods where we observe significant differences in daily consumption while analyzing the dataset. The prediction results around a three-day window around Christmas from the 24<sup>th</sup> of December to the 26<sup>th</sup> of December is plotted in figure 6 for different clusters. The general consumption data is well-aligned with the model's prediction, as can be seen from the figure across all clusters. While the sudden peaks are not accurately predicted by the model, the predicted consumption is on the lower side than the actual consumption, making it a reliable baseline for distribution system operators and grid operators.

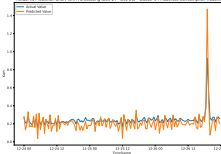
1) *Performance compared with Centralized Setup:* We evaluate the use of our lightweight model in a purely centralized setup to see the effectiveness of distributed learning in comparison to a global learning approach using a model of similar complexity. For the centralized setup, we assume that all consumption data is made available to the central server and that all clients participate at a consistent rate (set to 3× higher) compared to our federated learning scheme. The clustering approach is maintained in the centralized setup to ensure fairness in both cases. Both setups use 20 rounds of training - FL using training locally at the clients, while the centralized model trains for 20 epochs at the central server.

TABLE I: Monthly load forecasting RMSE: comparison between different cluster 08 households

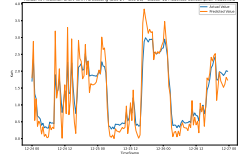
Month	Client Type (RMSE)		
	Mod(All)	High(Non-i.i.d)	Low(Non-i.i.d)
January	0.0636	0.2507	0.0778
February	0.0633	0.2581	0.0665
March	0.0623	0.2875	0.0760
April	0.0544	0.1531	0.0797
May	0.0570	0.0866	0.0740
June	0.0621	0.0583	0.0752
July	0.0638	0.1690	0.7425
August	0.0662	0.2217	0.7119
September	0.0528	0.1417	0.0730
October	0.0611	0.1896	0.0695
November	0.0539	0.2505	0.0684
December	0.0650	0.2484	0.0787



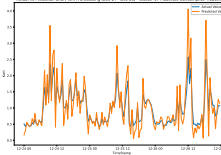
(a) Cluster 08



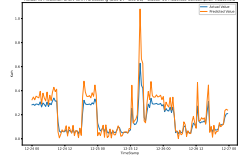
(b) Cluster 04



(c) Cluster 18



(d) Cluster 07



(e) Cluster 09

Fig. 6: Short-term daily forecasting (24<sup>th</sup> – 26<sup>th</sup> of December) across a selection of clusters

Figure 7 shows the average prediction error (RMSE) for the entire year for a single cluster (cluster 04) across different months. The results show that the federated model achieves better performance at the same model complexity.

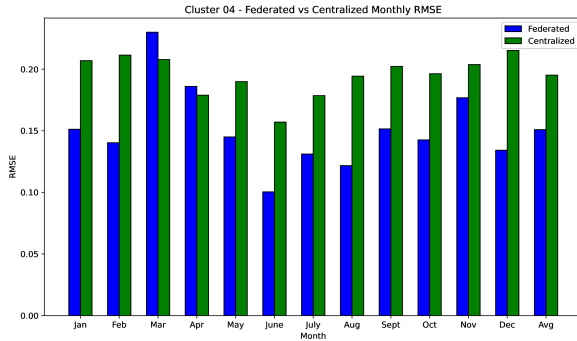


Fig. 7: RMSE of centralized vs federated prediction for cluster 04

2) *Performance comparison against other research works:* We compare our load forecasting model with the federated architecture used in [21], using average monthly RMSE across various clusters. We include households with high, low and moderate electric energy consumption. For our simulation, the number of participating clients was set to be less than 50% of the clients used in the competing method [21] for each cluster. Although an increase in the number of participating nodes does not guarantee improvement in prediction performance, it should potentially lead to a better global model based on the consumption profile of the additionally included nodes. Additionally, the work in [21] uses an LSTM-based model with 7505 parameters compared to our 4-layer feed-forward model with only 5569 learnable parameters (and thus lower communication costs during the model aggregation phase).

The table II) shows the results of our simulation capturing the average monthly RMSE of the models across all clusters over the entire year in comparison with the results reported in [21]. Despite using 50% fewer participating nodes in

TABLE II: Comparison of average RMSE of all clusters between our 4-layer feed-forward model and the LSTM-based FL model in [21]

Month	Our Proposed Model (RMSE)	M. Savi Model (RMSE)
January	0.2053	0.1463
February	0.1626	0.153
March	0.2176	0.1535
April	0.1800	0.1345
May	0.1657	0.1304
June	0.1434	0.1259
July	0.1685	0.1057
August	0.1526	0.1106
September	0.1581	0.1302
October	0.1554	0.1335
November	0.1729	0.1471
December	0.1728	0.1328

TABLE III: Comparison of Participation Ratio, Number of Rounds, and Average MAPE

Model	Household Ratio	Fed. Rounds	Avg MAPE
Taik's Model	36%	5	34.14%
Our Proposed Model	10.50%	20	22.01%

each cluster and 20% fewer computational requirements for the local model, our model is able to achieve an average RMSE of 0.17 compared to 0.14 of the LSTM model. While RMSE does not fully capture the prediction performance in this case, we believe that the trade-off is in line with the lower computational complexity, communication overhead and participation ratio in the network.

We also compare our results against the work in [25], where the model uses a participation ratio of 36% compared to the 10% in our method. Similar to the previous work, the work in [25] also uses an LSTM-based model with customized local model improvement at each round, with only 5 federated rounds compared to our 20. Our results, shown in table III, show that our approach achieves significantly better average prediction accuracy (in terms of MAPE) compared to the results reported in [25]. Additionally, we observe that reducing the number of federated rounds from 20 to 5 increases the average error across all clusters to increase by nearly 4% in our case, which is still better than the results achieved by [25].

### 3) Energy consumption of the model on an IoT platform:

To quantify the energy overhead in performing model training and inference at each client (smart meter), we implement the training and inference on an Arduino Uno R4 WiFi platform, that uses an 8-bit processor. The model was coded using Arduino C and compiled to the device using the standard compiler flow. For evaluation, we simulate the condition that the model is training on the data for one day and predicted the half-hourly consumption for the following day, with the output transferred via serial port to a standard laptop. To measure the energy consumed, we monitor the power consumed by the Arduino device using a calibrated USB energy monitor and record the consumption of the model under idle, training and prediction phases. We observed that the model consumes 50 mWh (milliwatt-hour) when averaged over the entire day on top of the idle power consumption of 440 mWh, when performing training and inference, resulting in an overhead of 11% on the Arduino platform. The result indicates that integration of our model will not incur significant energy consumption overhead in a similarly capable (or higher) smart meter hardware platform, making our approach ideally suited for this application.

## V. CONCLUSION

In this work, we propose a federated learning-based approach for distributed load forecasting that uses a lightweight fully connected neural network at its core and test it on a real-world openly available energy consumption dataset. By combining this lightweight model with clustering and randomized client selection, we emulate real-world conditions in an electric distribution grid. Our approach learns locally at each client's smart meter with only the model updates passed up to the central entity, reducing the privacy risks associated with granular aggregation of energy consumption data. Our simulations show that the light-weight model when combined with clustering is able to generate accurate load forecasting, and provides comparable results against competing FL-based load prediction approaches, trading off inference accuracy ( $\approx 21.4\%$  higher RMSE) with model complexity ( $\approx 25.8\%$  lower parameters) in one case, while achieving  $\approx 35.6\%$  better aggregate MAPE in another. We also show that training and inference using our model on an Arduino platform incurs an energy overhead of only 50 mWh when averaged over multiple cycles, which is a fraction of the standalone power consumption for the platform, making it ideally suited for load forecasting in a decentralized setting. In the future, we aim to show that decentralized learning can aid in improving the smart-grids stability and resilience without requiring aggregation of privacy-sensitive data.

## REFERENCES

- [1] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *International journal of forecasting*, vol. 30, no. 4, pp. 1030–1081, 2014.
- [2] A. Arif, M. Al-Hussain, N. Al-Mutairi, E. Al-Ammar, Y. Khan, and N. Malik, "Experimental study and design of smart energy meter for the smart grid," in *2013 International Renewable and Sustainable Energy Conference (IRSEC)*. IEEE, 2013, pp. 515–520.
- [3] E. Spanò, L. Niccolini, S. Di Pascoli, and G. Iannacconeluc, "Last-meter smart grid embedded in an internet-of-things platform," *IEEE Transactions on smart grid*, vol. 6, no. 1, pp. 468–476, 2014.
- [4] G. Díaz, J. Coto, and J. Gómez-Aleixandre, "Prediction and explanation of the formation of the spanish day-ahead electricity price through machine learning regression," *Applied Energy*, vol. 239, pp. 610–625, 2019.
- [5] A. R. Khan, A. Mahmood, A. Safdar, Z. A. Khan, and N. A. Khan, "Load forecasting, dynamic pricing and dsm in smart grid: A review," *Renewable and Sustainable Energy Reviews*, vol. 54, pp. 1311–1322, 2016.
- [6] A. Tascikaraoglu and B. M. Sanandaji, "Short-term residential electric load forecasting: A compressive spatio-temporal approach," *Energy and Buildings*, vol. 111, pp. 380–392, 2016.
- [7] K. Bandara, C. Bergmeir, and S. Smyl, "Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach," *Expert systems with applications*, vol. 140, p. 112896, 2020.
- [8] F. L. Quilumba, W.-J. Lee, H. Huang, D. Y. Wang, and R. L. Szabados, "Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 911–918, 2014.
- [9] H. Saxena, O. Aponte, and K. T. McConky, "A hybrid machine learning model for forecasting a billing period's peak electric load days," *International Journal of Forecasting*, vol. 35, no. 4, pp. 1288–1303, 2019.
- [10] G. Sideratos, A. Ikonomopoulos, and N. D. Hatzigiorgiou, "A novel fuzzy-based ensemble model for load forecasting using hybrid deep neural networks," *Electric Power Systems Research*, vol. 178, p. 106025, 2020.
- [11] H. J. Sadaei, P. C. d. L. e Silva, F. G. Guimarães, and M. H. Lee, "Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series," *Energy*, vol. 175, pp. 365–377, 2019.
- [12] G. Memarzadeh and F. Keynia, "Short-term electricity load and price forecasting by a new optimal lstm-nn based prediction algorithm," *Electric Power Systems Research*, vol. 192, p. 106995, 2021.
- [13] X. Guo, Q. Zhao, D. Zheng, Y. Ning, and Y. Gao, "A short-term load forecasting model of multi-scale cnn-lstm hybrid neural network considering the real-time electricity price," *Energy Reports*, vol. 6, pp. 1046–1053, 2020.
- [14] M. Sajjad, Z. A. Khan, A. Ullah, T. Hussain, W. Ullah, M. Y. Lee, and S. W. Baik, "A novel cnn-gru-based hybrid approach for short-term residential load forecasting," *Ieee Access*, vol. 8, pp. 143 759–143 768, 2020.
- [15] S. Wang, X. Wang, S. Wang, and D. Wang, "Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting," *International Journal of Electrical Power & Energy Systems*, vol. 109, pp. 470–479, 2019.
- [16] G. Eibl and D. Engel, "Influence of data granularity on smart meter privacy," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 930–939, 2015.
- [17] L. AlAbdulkarim and Z. Lukszo, "Impact of privacy concerns on consumers' acceptance of smart metering in the netherlands," in *2011 International Conference on Networking, Sensing and Control*, 2011, pp. 287–292.
- [18] B. Hu, Y. Gao, L. Liu, and H. Ma, "Federated region-learning: An edge computing based framework for urban environment sensing," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.
- [19] S. R. Pfohl, A. M. Dai, and K. Heller, "Federated and differentially private learning for electronic health records," *arXiv preprint arXiv:1911.05861*, 2019.
- [20] N. Gholizadeh and P. Musilek, "Federated learning with hyperparameter-based clustering for electrical load forecasting," *Internet of Things*, vol. 17, p. 100470, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2542660521001104>
- [21] M. Savi and F. Olivadese, "Short-term energy consumption forecasting at the edge: A federated learning approach," *IEEE Access*, vol. 9, pp. 95 949–95 969, 2021.
- [22] UK Power Networks: SmartMeter Energy Consumption Data in London Households, "<https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households>," 2014.
- [23] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.
- [24] Z. Shen, J. Ye, A. Kang, H. Hassani, and R. Shokri, "Share your representation only: Guaranteed improvement of the privacy-utility tradeoff in federated learning," in *The Eleventh International Conference on Learning Representations*, 2023.
- [25] A. Taïk and S. Cherkaoui, "Electrical load forecasting using edge computing and federated learning," in *ICC 2020-2020 IEEE international conference on communications (ICC)*. IEEE, 2020, pp. 1–6.