Towards Federated Short-Term Load Forecasting

Alexander Wallis*, Ulrich Ludolfinger*, Sascha Hauke, Maren Martens

University of Applied Sciences Landshut, Am Lurzenhof 1, 84036 Landshut (Germany), (+49) 871 506-690, alexander.wallis@haw-landshut.de, www.haw-landshut.de

Abstract:

The ongoing paradigm shift from centrally managed power grids to distributed Smart Grids necessitates accurate and robust Short-Term Load Forecasting capabilities. With the increasing deployment of smart meters, Machine Learning based techniques show promising results. However, extensive centralized data collection and processing leads to concerns with respect to data privacy and transmission bandwidth. In this paper, we provide a solution for developing a global Machine Learning model without centralized data aggregation. This Federated Learning approach trains the model by aggregating parameter updates received from each participating household. We show that our model reaches sufficient forecast accuracy and also provides data privacy and security for every user within the Federated Learning framework.

Keywords: Federated Learning, Short-Term Load Forecasting, Smart Grid, Data Privacy

1 Introduction

A carbon-free power sector is only possible with an increasing and reliable integration of Renewable Energy Resources (RESs) into the existing power grid [1]. However, this integration results in a growth of network management tasks for the power grid operator to handle the volatile behaviour of those RESs. To guarantee a stable and reliable power supply, accurate demand forecasting is mandatory. Particularly on low-voltage grid level and for Smart Grid (SG) development, load forecasting is an important tool to secure network stability since smoothing effects are not as influential as on higher level. For Short-Term Load Forecasting (STLF), models based on Machine Learning (ML) methods have shown promising results [2]. This ongoing success of Artificial Intelligence (AI) in general is driven by two main factors: the amount of data generated by businesses, governments, and private citizens is rapidly increasing and computer processors are getting faster and cheaper. However, the accuracy of AI models highly relies not only on the amount of available data but also on the quality. Therefore, STLF continues to be a problem with much room for further improvement [3]. Even though load consumption is highly household specific, there are many characteristics that overlap between households. For example, power consumption at night is usually lower than during the day, or the power consumption rises when it is cold outside. It is therefore

* Contributed equally

conceivable to develop a global forecasting model which is adaptable to load forecasting in multiple buildings.

Traditionally, ML models are trained on gathered data which is aggregated on centralized servers. This requires a transfer of highly personal data to remote servers, e.g., high-resolution power consumption data in the context of SG development that immediately raises data security and privacy concerns – further enhanced by the General Data Protection Regulation (GDPR) of the European Union [4]. Additionally, as ML models are getting more complex a general practice is the distribution of the training process over various machines to master the computational complexity [19]. Certainly, such transactions violate the GDPR as it is not always transparent to household users what their data is used for. Beside the data privacy concerns, transferring data between clients and servers need sufficient bandwidth available for stable communication [8]. Therefore, we propose a Federated Learning (FL) approach for STLF with the following contributions:

1. Analyses of households' consumption behavior used for feature engineering and selecting a suitable model for the given problem.
2. Development of a flexible FL pipeline for simulating an arbitrary number of households within a SG.
3. Evaluation of the forecast accuracy on real-world data sets with respect to data privacy.
4. Comparison between an improved global model and a local trained model.

Furthermore, we show that by using a global model and fine tuning it for a particular household, the training time can be significantly reduced. The global model can be further improved with FL.

The remaining paper is structured as follows. In Section 2, we give a brief overview of state-of-the-art applications in STLF, especially with respect to FL. A detailed description of the problem formulation, the used data set, the various preprocessing steps as well as the development process from a single household model to a FL infrastructure is given in Section 3. The evaluation of our proposed model is presented in Section 4, followed by a concluding summary and possible starting points for further research works as well as newly opened questions (Section 5).

2 Related Work

In this section, first we give an overview of the current state-of-the-art methods for STLF with focus on ML approaches, after that we describe some FL works and how they can tackle the data privacy problem in SG development.

2.1 Short-Term Load Forecasting

For the STLF problem at household level, ML models like neural networks are valid solutions due to their capability to handle complex patterns and non-linear relationships [5]. Various types of neural networks outperform typical time-series prediction models such as Autoregressive Integrated Moving Average (ARIMA) [6]. Particularly, the Long-Short Term Memory (LSTM) emerges as the benchmark network architecture [7][8]. In our previous work, we showed that combined learning methods like Random Forest (RF) together with weather information and further feature engineering provide promising forecast accuracy [2]. All mentioned approaches assume that the training data is hold centralized. This assumption rises privacy concerns, since load data reveal sensitive information. To address this problem, a new FL paradigm, which is further described in the following, is proposed.

2.2 Federated Learning

The basic idea behind FL was first published by a developer team from *Google* in 2016 to reduce uplink communication costs [9]. Meanwhile, this concept is used in various applications and sectors [10]. In the context of SG development, FL is used to predict the energy demand of electric vehicles [11]. Furthermore, based on a reinforcement learning model, FL is able to manage local home energy management systems [12]. In [13], the authors propose a FL approach for load forecasting in combination with edge computing devices. They showed that the accuracy highly depends on the computation power of the edge computing device but reaches sufficient predictions. Similar results are provided by the authors in [14] for ultra-short load forecasting.

Especially the later papers encouraged our work. Since the previous mentioned FL approaches for STLF are working with single value predictions, e.g., load forecasting for a whole day or next hour, the problem of predicting the next day's load curve is still unsolved. So, in the following, we propose a combination of the suggested ML model for STLF with a FL approach with respect to data privacy and reduction of data transmission bandwidth.

3 Methodology

In the following, we give a formal description of the problem. The goal of FL is to learn a model with parameters $\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}$, in which the training data is distributed over a set of clients — in this case single households $\mathcal{H}$. Contrary to the traditional approach with a centralized data aggregation and training process, within a FL framework every household trains its own model and sends only the parameters from its specific model (see Figure 1) to the server. There, the households' parameters are aggregated, e.g., averaged. After that, the global model is updated and distributed to the selected households.

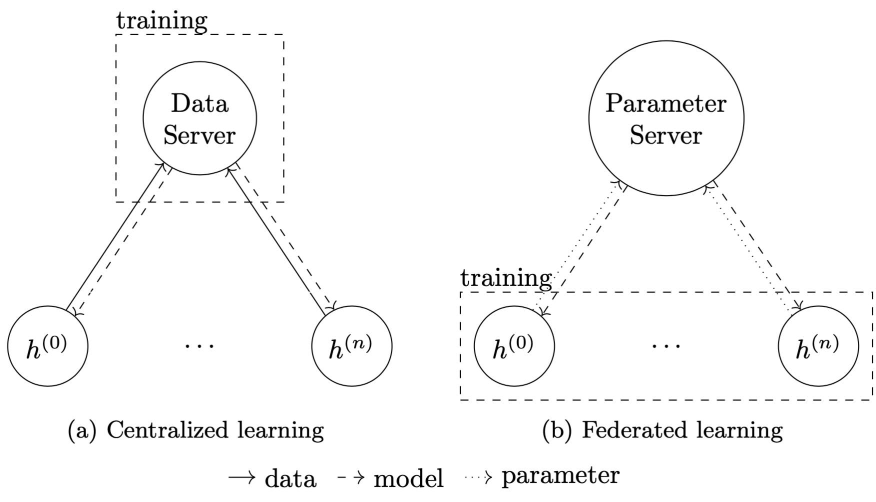


Figure 1: Difference between (a) centralized learning, where the data is aggregated on the server, and (b) federated learning, where the parameters are aggregated on the server.

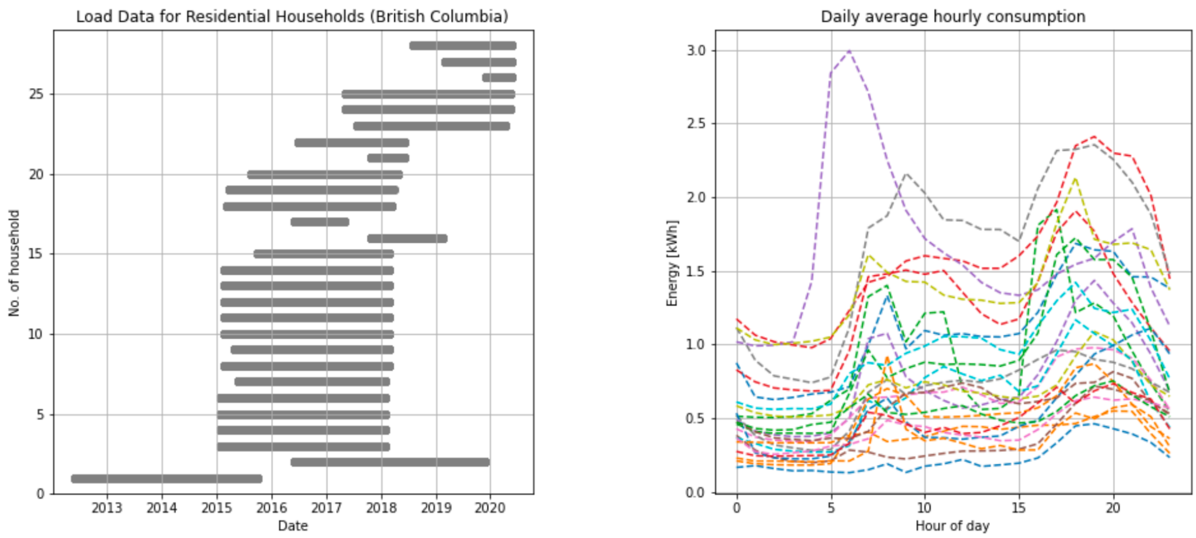
This whole procedure is an iterative process. In every round $t \geq 0$, the parameter server distributes the current model $\mathbf{W}_t$ to households $h \in \mathcal{H}_n$, where $\mathcal{H}_n \subset \mathcal{H}$ is a randomly selected subset of size n. Through some training epochs, every household $h \in \mathcal{H}_n$ generates a household-specific, updated model $\mathbf{W}_t^{(h)}$ and sends its update $\mathbf{H}_t^{(h)} \coloneqq \mathbf{W}_t^{(h)} - \mathbf{W}_t$ back to the server, where the global update is computed by an incremented average as follows [15]:

$$\mathbf{W}_{t+1} = \mathbf{W}_t + \mathbf{H}_t, \text{with } \mathbf{H}_t \coloneqq \frac{1}{|\mathcal{H}_n|} \sum_{h \in \mathcal{H}_n} \mathbf{H}_t^{(h)} \tag{1}$$

In the following, we show how this FL approach is implemented for our task with a summary of the used data set and executed pre-processing steps.

3.1 Data Set & Pre-Processing

For our work, we used the *HUE* data set [16]. For 28 residential households in Vancouver, BC, this publicly available data set contains individual hourly energy consumption records, each representing roughly three years. In Figure 2(a) the date ranges for the households are displayed. It can be seen, that for 16 households (with IDs $3-15$ and $18-20$) consumption data is available for the same time window of approximately three years. For those households the daily average load profile is shown in Figure 2(b).



(a) Time range per household (b) Daily average load profiles

Figure 2: Overview of the HUE data set with (a) covered date range per household and (b) daily average load profiles.

The data coverage per household is between $97\ \%$ and $100\ \%$. Since most ML models are not able to handle missing values in their input data, gaps with less than three hours are filled through linear interpolation. The remaining missing data points are filled by average values from the same hour for a given weekday within a specific month. Furthermore, categorical information, e.g., month, weekday, needs to be transformed to be interpreted correctly by ML models. The typical approach to keep dimensionality low, is to encode a value x as a cyclic feature [2] composed of the sine and cosine components

$$x_{sin} = \sin\left(\frac{2 \times \pi \times x}{T}\right) \text{ and } x_{\cos} = \cos\left(\frac{2 \times \pi \times x}{T}\right) \tag{2}$$

with $T \in \{7, 12, 24\}$ for weekday, month, and hour of day. Furthermore, we added weather information between the given date range (see Figure 2(a)) for the Vancouver, BC area. Values for temperature, relative humidity, and atmospheric pressure were taken from the *World Weather Online API* [17]. The columns of the resulting data set are summarized in Table 1.

Table 1: Data set description after pre-processing steps.

	Data Set Columns						
Name	Energy	Weekday	Month	Hour	Temperature	Humidity	Pressure
Unit	kWh	(float, float)	(float, float)	(float, float)	°C	%	kPa

Completing the pre-processing, the numerical values are scaled by subtracting the respective mean and dividing by the standard deviation. This procedure leads to a final set of household specific data sets $\mathcal{D} = \{\mathcal{D}^{(h)} \mid h \in \mathcal{H}\}$.

3.2 Model Development

Before explaining the FL, we describe the used ML model. First, we formalize the problem our model needs to solve, then we propose one concrete model for FL. Since we are interested in STLF, the model's prediction is a total of 24 energy consumption values for every hour of the following day. For the STLF problem, a special kind of Neural Network — the LSTM — is the state-of-the-art ML model. This network architecture is able to represent non-linearity and especially handle seasonality due to its internal self-loops that are used for information storage. There are five main elements within a LSTM: 1) input gate, 2) forget gate, 3) output gate, 4) cell, and 5) state gate [18]. For our model architecture, we use multiple LSTM layers (see Figure 3). This is recommended by [19] and also achieved the best performance in our tests. By adding an additional fully connected layer before the last output layer, we could further improve our model architecture.

Layer	Parameters	Activation
LSTM	168×10	tanh
LSTM	168×64	tanh
Dense	64	ReLU
Dense	24	-

Figure 3: LSTM architecture used for our FL model.

Like most other ML techniques, it requires a training and testing phase. Let $\mathbf{X} = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ be a vector representing the input data of a particular day d and household h, where for our LSTM model $\mathbf{X}$ contains the hourly data (see Table 1) of the seven days until d. Further let $\mathbf{Y} = (y_1, \dots, y_{24})^T \in \mathbb{R}^{24}$ be the vector, which shows the actual load value for household h for the 24 hours of the day following d. Then the training set, for household h, is $\mathcal{D}_{\text{train}}^{(h)} = \{(\mathbf{X}_1, \mathbf{Y}_1), \dots, (\mathbf{X}_m, \mathbf{Y}_m)\}$, where the $(\mathbf{X}_i, \mathbf{Y}_i)$ are the pairs of $\mathbf{X}$- and $\mathbf{Y}$-vectors for every day d in the considered time-period.

For every household h, we train a local LSTM model as shown in Figure 3. For every day in the considered time-period, we have one training round, where the process updates weights $\mathbf{W}_t^{(h)}$ using the training data $\mathcal{D}_{\text{train}}^{(h)}$. The update aims to minimize the Mean Squared Error (MSE) between the predictive values $\widehat{\mathbf{Y}}$ and the actual values $\mathbf{Y}$, i.e., $\sum_{i=0}^{24}(\hat{y}_i - y_i)^2$, for the energy loads of the 24 hours of the following day.

In the following, this LSTM architecture is also used for our FL approach.

3.3 Federated Learning Infrastructure

Our FL approach is an iterative process, which in every communication round tunes a global model by aggregating weight updates provided by the involved households. This process is detailed more specifically in the following Algorithm 1.

Algorithm 1 Federated Learning Procedure

Require: Rounds T, Households $\mathcal{H}$, Data $\mathcal{D}$, clients per round n
  Initialize $t = 0$ and random global weights $\mathbf{W}_{t=0}$
  while $t < T$ **do**
    $\mathcal{H}_n \subsetneq \mathcal{H}$ with $|\mathcal{H}_n| = n$ ▷ Select random subset of households
    Broadcast weights $\mathbf{W}_t$ to all $h \in \mathcal{H}_n$
    for $h \in \mathcal{H}_n$ **do**
      $\mathbf{W}_t^h \leftarrow \text{train}(\mathcal{D}_{\text{train}}^{(h)})$ ▷ Train local LSTM model
      $\mathbf{H}_t^h \leftarrow \mathbf{W}_t^h - \mathbf{W}_t$
      Send update $\mathbf{H}_t^h$ to server
    end for
    Calculate $\mathbf{W}_{t+1}$ ▷ See Equation (1)
    $t \leftarrow t + 1$
  end while
  return $\mathbf{W}_T$

For implementation, we chose the TensorFlow Federated Framework by Google's TensorFlow [20] in combination with Keras for model development [21]. In every round t, a random subset of three households is selected from $\mathcal{H}$. These households receive the current global model $\mathbf{W}_t$ from the server and train their local model with their own data $\mathcal{D}_{\text{train}}^{(h)}$. The resulting model updates are aggregated by the server based on the calculation in Equation 1. In the following Section 4, we specify the chosen parameters and show the model performance for various test scenarios.

4 Evaluation

The previously described FL framework is implemented in Python 3.6.9 with the help of TensorFlow 2.4.1. The model was trained on the Nvidia Geforce RTX 2080 graphic card. For our test setup, we selected 16 households from the *HUE* data set, that contain load data from the same period. The period covers three years from 2015 to 2018. We trained our global FL model for 2.500 epochs in total. For each training epoch, we simulated the availability of three households. This means, we selected randomly three out of the 16 households in each training epoch for FL. This is to simulate that households join and leave the training network over time.

For evaluation we use forward chaining since this approach is closest to reality. For this purpose, we used a new household from the *HUE* data set (ID 24) that was still unknown to the global model. We first evaluated the global model and a yet untrained model (with randomly initialized parameters) on the first month of the data set. For every following month, we updated both models using the respectively previous month. The evaluation results can be seen in Figure 4.

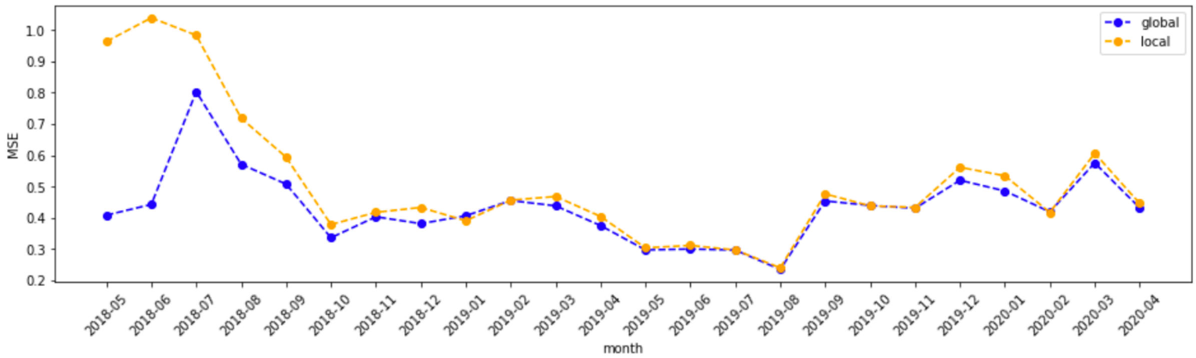


Figure 4: Comparison of the Mean Squared Errors (MSEs) between a global and a local model. Both models are tested incrementally on every new month.

It can be seen, that especially at the beginning the global model performs a lot better than a newly trained model. After some time, sufficient data is available, thus the new model also provides similarly good predictions. This shows, that with a global LSTM model trained by FL, the training period, until good predictions can be made, can be significantly shortened. For illustration, Figure 5 shows three days of 24 h predictions on the test household.

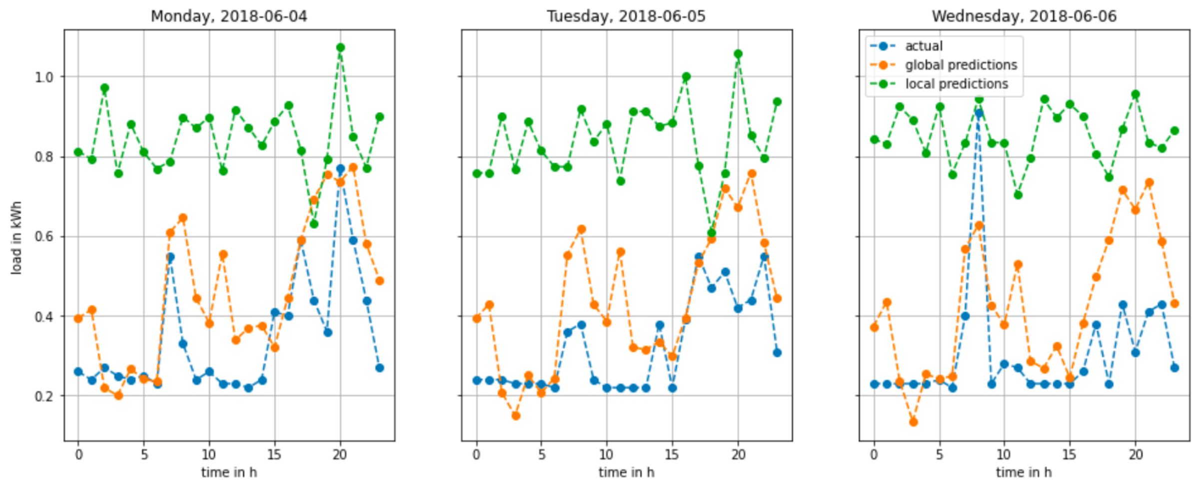


Figure 5: The next 24 h predictions for the test household (ID 24), after one month of training. The global predictions are the forecasting values with the fine-tuned global model. The local predictions are based on a newly generated model.

These results are generated by a fine-tuned global FL model and a newly trained local model on the first month for the test household. In Figure 5 it can be observed that the predictions of the fine-tuned FL model are closer to the actual load measurements than those of the newly trained local model.

5 Conclusions & Future Work

In this paper, we showed how a centralized ML model can be trained for STLF with respect to data privacy and security (see Section 3). We selected a proper ML model and architecture (Section 2.1) and developed a framework based on the FL approach described in Section 2.2. Based on this FL pipeline, we trained a global forecasting model through 2.500 communication rounds in total, whereby in every round a subset of three randomly chosen households out of 16 were used for the training process. Conclusively, we evaluated the accuracy of the developed global model with a new household (Section 4). We observed that within the first six months the global model reaches a significantly higher accuracy than the local one. This shows that the time for a local model development can be shortened until enough data is collected. In this paper, we demonstrated how FL can be used for STLF with respect to data privacy and furthermore for bypassing the data collection time for single households.

For future work it will be interesting to test various ML models beside the proposed LSTM as a global FL model. During the development process, the high variance of load profiles between households countered the forecasting accuracy when the number of clients within the FL framework increased. Thus, we assume that a classification of the different households within a SG would lead to further forecasting accuracy by clustering similar households for the FL process. Additionally, an adaptive local fine-tuning composed of one or more chained local ML models combined with ongoing FL process is an interesting research field.

Literature

[1] J. D. Jenkins, M. Luke, and S. Thernstrom, "Getting to Zero Carbon Emissions in the Electric Power Sector," *Joule*, vol. 2, no. 12, pp. 2498–2510, 2018.

[2] A. Wallis, A. Haber, and S. Hauke, "Menschliches Wissen und Künstliche Intelligenz in einem Prognosemodell für Erneuerbare Energien," 2020.

[3] A. K. Srivastava, A. S. Pandey, and D. Singh, "Short-term load forecasting methods: A review," in *2016 International Conference on Emerging Trends in Electrical Electronics & Sustainable Energy Systems (ICETEESES)*, Mar. 2016, pp. 130–138.

[4] European Commission, "2018 reform of EU data protection rules." May 2018, Accessed: Jun. 17, 2019. [Online]. Available: https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf.

[5] S. Kulkarni, S. P. Simon, and K. Sundareswaran, "A spiking neural network ({SNN}) forecast engine for short-term electrical load forecasting," *Appl. Soft Comput.*, vol. 13, no. 8, pp. 3628–3635, Aug. 2013.

[6] M. Y. Cho, J. C. Hwang, and C. S. Chen, "Customer short term load forecasting by using ARIMA transfer function model," in *Proceedings 1995 International Conference on Energy Management and Power Delivery EMPD '95*, 1995, vol. 1, pp. 317–322 vol.1.

[7] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, "Short-Term Residential Load Forecasting Based on Resident Behaviour Learning," *{IEEE} Trans. Power Syst.*, vol. 33, no. 1, pp. 1087–1088, Jan. 2018.

[8] Y. Wang, D. Gan, M. Sun, N. Zhang, Z. Lu, and C. Kang, "Probabilistic individual load forecasting using pinball loss guided {LSTM}," *Appl. Energy*, vol. 235, pp. 10–20, Feb. 2019.

[9] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon,

"Federated Learning: Strategies for Improving Communication Efficiency," 2016.

[10] S. Abdulrahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A Survey on Federated Learning: The Journey From Centralized to Distributed On-Site Learning and Beyond," *{IEEE} Internet Things J.*, vol. 8, no. 7, pp. 5476–5497, Apr. 2021.

[11] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, M. D. Mueck, and S. Srikanteswara, "Energy Demand Prediction with Federated Learning for Electric Vehicle Networks," Dec. 2019.

[12] S. Lee and D.-H. Choi, "Federated Reinforcement Learning for Energy Management of Multiple Smart Homes with Distributed Energy Resources," *{IEEE} Trans. Ind. Informatics*, p. 1, 2020.

[13] A. Taik and S. Cherkaoui, "Electrical Load Forecasting Using Edge Computing and Federated Learning," Jun. 2020.

[14] J. Li, Y. Ren, S. Fang, K. Li, and M. Sun, "Federated Learning-Based Ultra-Short term load forecasting in power Internet of things," Aug. 2020.

[15] 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*, 2017, pp. 1273–1282.

[16] S. Makonin, "HUE: The Hourly Usage of Energy Dataset for Buildings in British Columbia." Harvard Dataverse, 2018.

[17] "World Weather Online." .

[18] A. Almalaq and G. Edwards, "A Review of Deep Learning Methods Applied on Load Forecasting," Dec. 2017.

[19] G. Hinton *et al.*, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012, doi: 10.1109/MSP.2012.2205597.

[20] TensorFlow Developers, "TensorFlow." Zenodo, 2021, doi: 10.5281/ZENODO.4724125.

[21] F. Chollet and others, "Keras." GitHub, 2015, [Online]. Available: https://github.com/fchollet/keras.