4-23-2024

Personalized Federated Learning for Load Forecasting

Sydney J. Winters
University of Alabama in Huntsville

Personalized Federated Learning for Load Forecasting

by

Sydney J. Winters

An Honors Capstone

submitted in partial fulfillment of the requirements

for the Honors Diploma

to

The Honors College

of

The University of Alabama in Huntsville

April 23, 2024

Honors Capstone Project Director: Dr. Dinh Nguyen

Sydney Winters 4/23/2024

Student Date

4/23/2024

Project Director Date

Aleksandar Milenkovic
Digitally signed by Aleksandar Milenkovic
Date: 2024.04.25 15:32:13 -05'00'

Department Chair Date

Sean Lane
Digitally signed by Sean M Lane
Date: 2024.04.30 11:27:42 -05'00'

Honors College Dean Date

HONORS COLLEGE
THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Honors College

Frank Franz Hall

+1 (256) 824-6450 (voice)

+1 (256) 824-7339 (fax)

honors@uah.edu

Honors Thesis Copyright Permission

This form must be signed by the student and submitted with the final manuscript.

In presenting this thesis in partial fulfillment of the requirements for Honors Diploma or Certificate from The University of Alabama in Huntsville, I agree that the Library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by my advisor or, in his/her absence, by the Chair of the Department, Director of the Program, or the Dean of the Honors College. It is also understood that due recognition shall be given to me and to The University of Alabama in Huntsville in any scholarly use which may be made of any material in this thesis.

Sydney Winters

Student Name (printed)

Sydney Winters

Student Signature

4/23/2024

Date

Contents

Abstract

Many more devices are becoming smart devices that collect data about the user, including smart electrical meters. This data could be very useful in predicting future trends in energy consumption, but there are growing concerns for customers' data privacy. The goal of this project is to create a federated learning system that will use smart meter data to train deep learning models to create energy consumption forecasts while preserving the privacy of the customers by keeping their data local to their system. In this project simulations of personalized federated learning are compared to federated learning and we explore some other factors of the base model, including the hidden layer parameters, that impact the performance of the model to see what produces a more accurate result.

Introduction

Federated Learning (FL) is a type of machine learning that focuses on training a machine learning model when there is a lack of data in a centralized location [1]. Rather than training a model on a single server, FL uses a client/server model to allow many client machines to train a global model on data that is local to them, and then send their trained model back to the server [1]. Once the server receives models from the clients, different aggregation methods can be used to combine these local models to a new global model which is then distributed back to all the clients for more training [1]. FL provides an advantage when working with data that can not be aggregated due to being sensitive, since the model is the only information shared between the clients and the server, rather than the training data [2]. But FL faces challenges of providing high accuracy when data is distributed heterogeneously across the clients. In situations of non-independently and identically distributed (non-iid) data, personalized federated learning (PFL) has been proposed as a solution. PFL is a subset of FL that introduces extra steps on the clientside to better adapt the global model to the data present on the client to achieve a higher accuracy [6].

There are many applications of FL where data security and privacy are a concern due to the sensitive nature of data. In this paper we focus on the application of FL in energy forecasts by using smart electric meter data that was collected from multiple different households in India [4]. This data could benefit from the use of PFL as the number of points is not evenly distributed across all clients, thus being non-iid data that may not perform the best with typical FL methods. We focus on training an LSTM, Long Short Term Memory, model to see if we can create accurate predictions and how FL and PFL impact the predictions.

Background

The Data

For these simulations, smart electric meter data was used to see if PFL could increase the accuracy of energy consumption forecasting. The dataset consists of the readings from smart meters from multiple different houses in Uttar Pradesh, India [4]. The measurements were taken at three minute intervals and include the energy consumption, the average voltage and current from the meter respective meter [4]. For our model training we use the meter labels to split the data onto our clients, using each client to simulate a household, and we use the energy consumption, measured in kWh, for the training of our models. This is so the output of the trained model is a predicted forecast value of the future energy consumption for each of the households. Even though the data is taken in 3 minute intervals, each meter, or household does not have the same number of data points, causing the data to be unevenly distributed across the clients, or non-iid. When given to the model for training batches of 10 data points are used to represent half hour increments that the model can use to make a prediction about the next data point.

The Base Model

For the base or client model in our simulations we are using a Long-Short Term Memory (LSTM) model. This is because these are a type of Recurrent Neural Network (RNN) that have been optimized for time-series problems [5]. This is done by adding blocks into the hidden layers of the models that remember the current state of the values for a certain period of time, given the model a form of memory [5]. Specifically, we are using a basic LSTM from Pytorch with an input size of 10 data points, which is our batch size and we changed the parameters of the hidden layer to try to improve the accuracy of the model. The hidden layers in an LSTM do calculations on the input to determine what the output should be [7]. By default, the LSTM model we started with had a singular hidden layer with 50 parameters [9]. We then changed the number of parameters to two to see how this impacted the outcome as the number of parameters in the hidden layer impacts the complexity of the model's forecasting [9].

The FL and PFL Methods

One of the most widely used FL algorithms is Federated Averaging or FedAvg, which implements aggregation by averaging the weights of each of the client models based on the number of local training rounds performed [3]. Once the model is averaged it is sent back to the clients for more local training [3]. FedBN is similar to FedAvg but it adds an extra layer of batch normalization to the model that is not transmitted back to the global model [8]. This batch normalization layer is not transmitted as keeping it local to the client allows the model to be more personalized to the client and its data [8]. For the non-batch normalization layers, they are still averaged using the FedAvg algorithm at the global level [8].

Simulations

In this paper we run multiple simulations using the smart meter data to see what creates the best forecasting model. For these simulations, we ran a centralized model first, that had no federated learning, and used the entire dataset to train a singular model. We then ran both FedAvg and FedBN with three and five clients. For these simulations, that data was split as follows:

Client 1: 14.8%

Client 2: 29 %

Client 3: 19.13 %

Client 4: 19.13%

Client 5: 17.89 %

The client split was based on the data per meter, as each client was assigned the data for a singular household. For every federated learning and personalized federated learning simulation, we ran 50 global rounds and 3 local rounds. We then ran each of these sets of simulations on the base model that had 50 parameters in the hidden layer and the one that had 2 parameters in the hidden layer.

50 Hidden Layer Parameters

We first ran all our simulations using a base LSTM with 50 hidden layer parameters, which was the default for this model. We first ran simulations for FedAvg that included the centralized LSTM model that used the entire dataset to train the model, as well as a three-client system and a five-client system. We then repeated the three and five client simulations for the FedBN algorithm to see how personalization changed the output. Figures 1 and 2 show the loss and accuracy for FedAvg and Figures 3 and 4 show the metrics for FedBN.

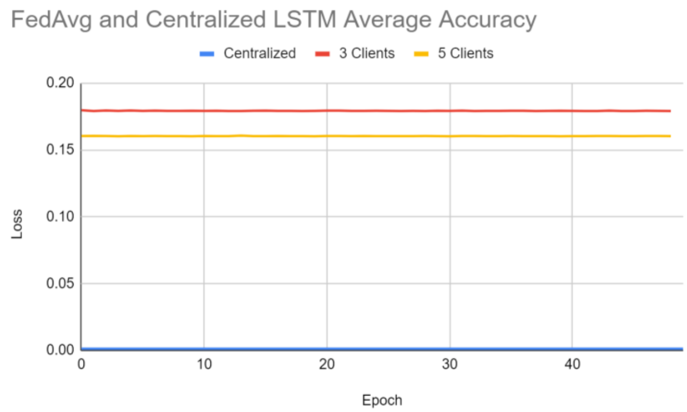

Figure 1. The Average Loss of FedAvg Systems and Centralized LSTM with 50 Hidden Layer Parameters.

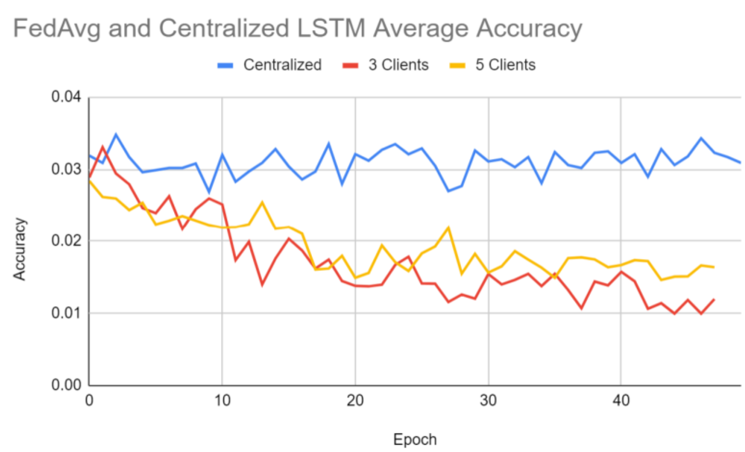

Figure 2. The Average Accuracy of FedAvg Systems and a Centralized LSTM with 50 Hidden Layer Parameters.

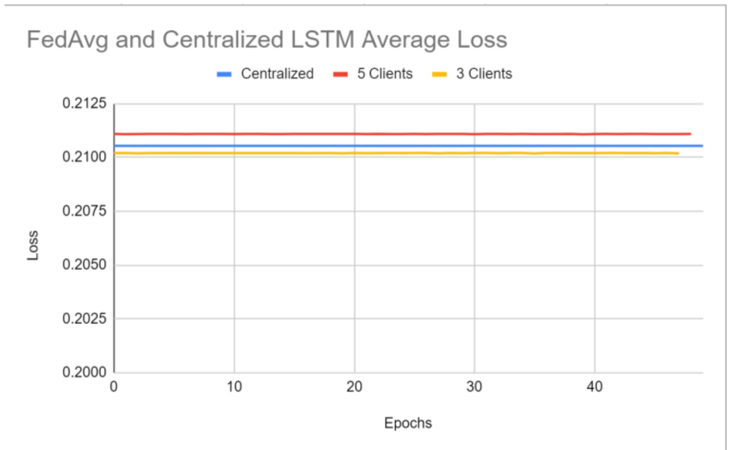

Figure 3. The Average Loss of FedBN Systems and a Centralized LSTM with 50 Hidden Layer Parameters.

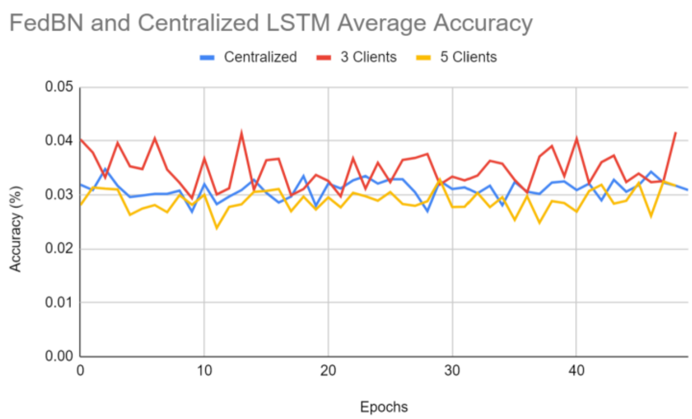

Figure 4. The Average Accuracy of FedBN Systems and a Centralized LSTM with 50 Hidden Layer Parameters.

Looking at the average accuracy in the figures above, we can see that the maximum accuracy for FedAvg and FedBN was less than 0.5%. This could be due to the complexity of the hidden layer. Since we had 50 parameters in the hidden layer, the model should be able to detect more complex patterns, but it increases the time necessary to train the model [8]. Since the accuracy was very low to start and decreased or stayed near a consistent value over the 50 epochs, we then decreased the number of hidden layer parameters to see how that impacted the model performance. We decrease the hidden layer parameters because the model is not properly learning the data patterns and may be overfitting causing inaccurate results.

When comparing the performance of FedAvg and FedBN we can see that FedBN does outperform FedAvg slightly in both accuracy and loss. This can be seen when looking at both the three and five client systems as shown below.

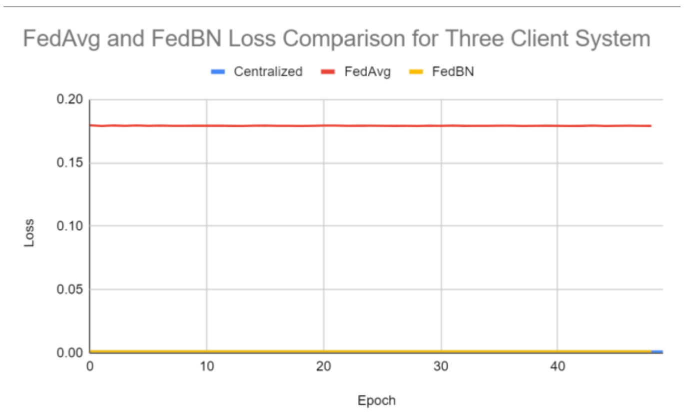

Figure 5. The Loss Comparison between FedAvg and FedBN for Three Client System.

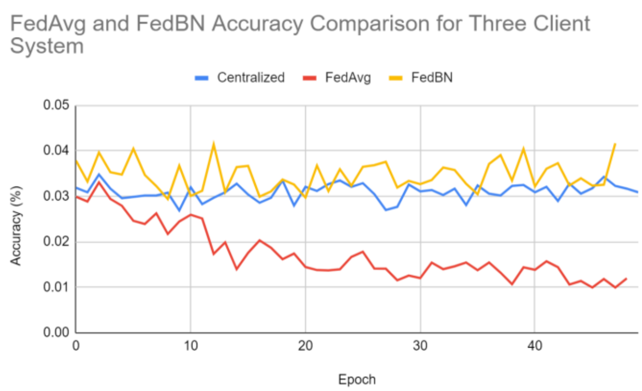

Figure 6. The Accuracy Comparison between FedAvg and FedBN for Three Client System.

FedBN performs better than the centralized LSTM and the FedAvg three system client in both accuracy and loss. When looking at the five client systems, we can see that FedBN still outperforms the FedAvg system, but the centralized model has a similar accuracy to the FedBN system. This does not happen with the FedBN or the centralized model, showing that averaging alone may have a negative effect on the accuracy of the model, but that when combined with batch-normalization is able to limit the decreasing accuracy.

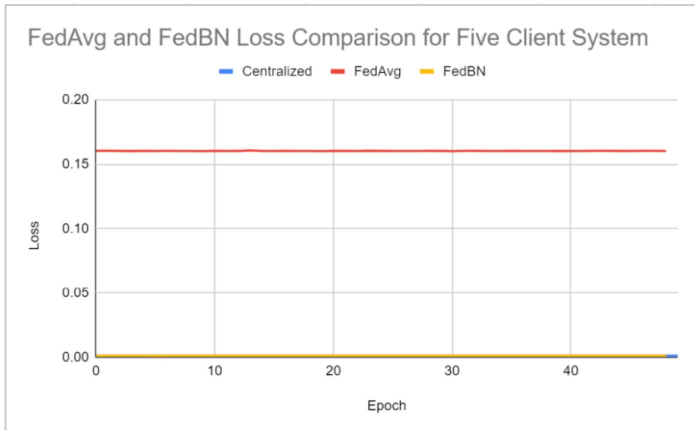

Figure 7. The Loss Comparison between FedAvg and FedBN for Five Client System.

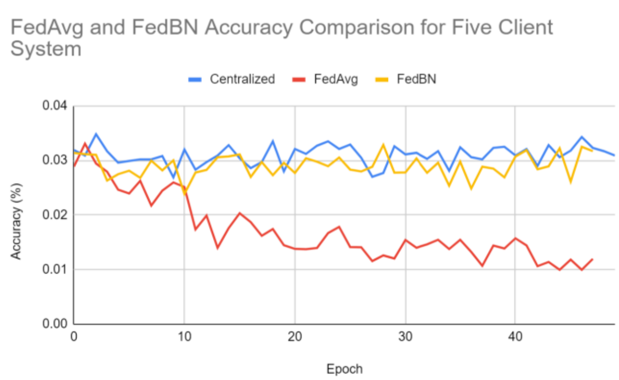

Figure 8. The Accuracy Comparison between FedAvg and FedBN for Five Client System.

2 Hidden Layer Parameters

Since 50 hidden layer parameters seem too complex for our dataset, we then changed the number of hidden layer parameters to 2. This is to see if we could stop the possible overfitting of the data. We first ran simulations for FedAvg that included the centralized LSTM model that used the entire dataset to train the model, as well as a three-client system and a five-client system. For the three-client system, the first three clients from the above list were used. Figure 9 and Figure 10 show the average loss and accuracy for each of the systems per epoch.

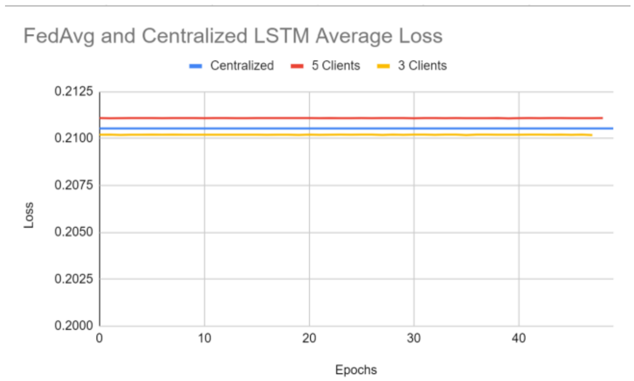

Figure 9. The Average Loss for the two FedAvg Simulations and the Centralized LSTM with 2 Hidden Layer Parameters.

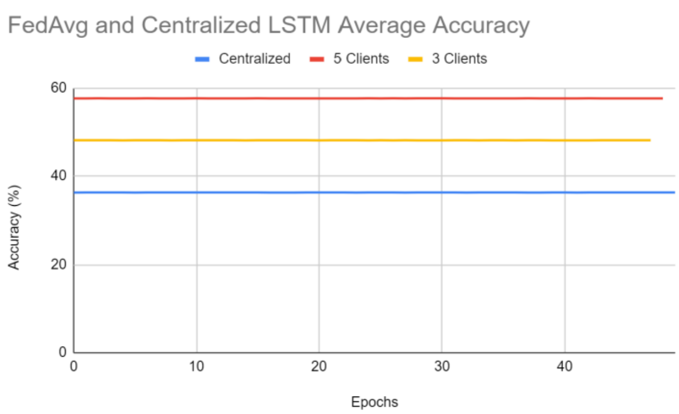

Figure 10. The Average Accuracy for the two FedAvg Simulations and the Centralized LSTM with 2 Hidden Layer Parameters.

Looking at the above results we can see that there is no change in the loss, as well as no change in the accuracy, meaning that there was no learning happening in the model. We then repeated the same simulations but using FedBN rather than FedAvg, whose results are shown below.

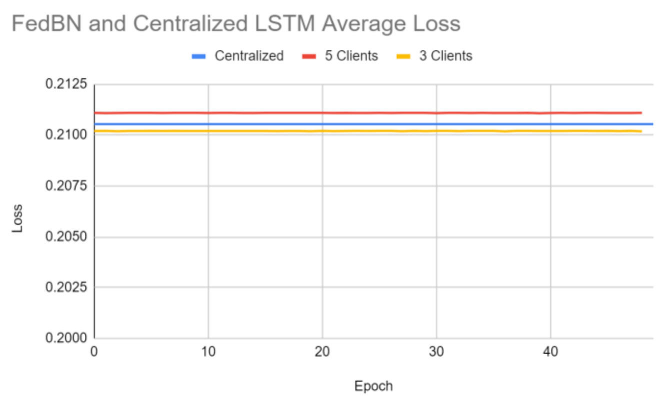

Figure 11. The Average Loss for the two FedBN Simulations and the Centralized LSTM with 2 Hidden Layer Parameters.

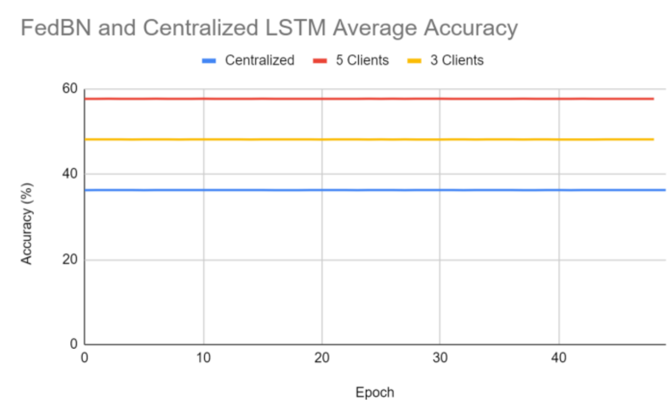

Figure 12. The Average Accuracy for the two FedBN Simulations and the LSTM with 2 Hidden Layer Parameters.

The lack of learning in these simulations may indicate that the number of parameters in the hidden layers in the model are not enough to allow the model to properly learn any correlation between the data. If the model is not able to recognize or save patterns from the data then it is unable to improve.

When looking at the performance of FedAvg compared to FedBN we can see that the personalization has no impact on the results of the LSTM model. Below we can see that the loss for the three client systems is the same for the centralized, FedAvg and FedBN, although the accuracy of the FL systems is greater than the accuracy of the centralized system.

Figure 13. The Loss Comparison between FedAvg and FedBN for Three Client System.

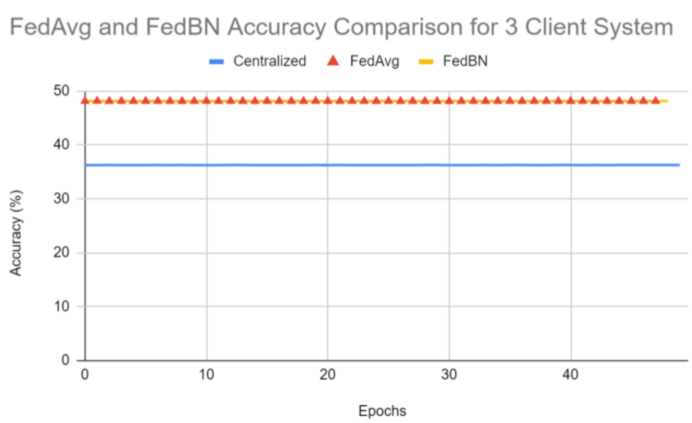

Figure 14. The Accuracy Comparison between FedAvg and FedBN for Three Client System.

When looking at the comparison for the five client systems we can see that the loss for the FL algorithms were the same and they were slightly higher than the loss for the centralize system. We can also see that the accuracy of the FL systems was the same and was much better than the accuracy of the centralized system.

Figure 15. The Loss Comparison between FedAvg and FedBN for Five Client Systems.

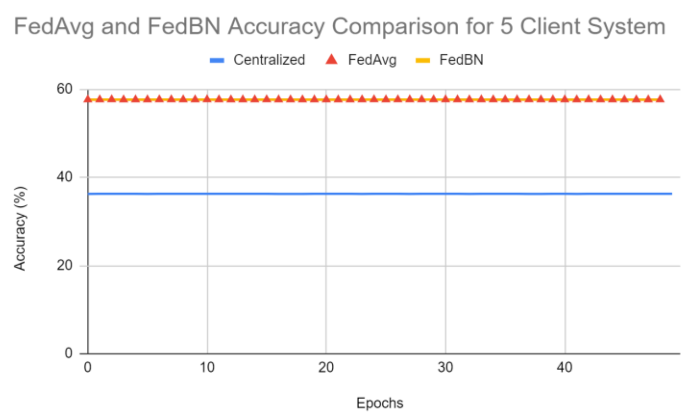

Figure 16. The Accuracy Comparison between FedAvg and FedBN for Five Client Systems.

Conclusion

Federated learning is a system that allows the use of data that is located in different sources and may not be able to be centralized. In this project we focused on how federated learning could be used to allow smart meter data to be used to create forecasts for future energy consumption. We focus on experimenting on federated learning and personalized federated learning to see the impacts on the performance of an LSTM model. We found that personalized federated learning does not particularly help increase the accuracy of the model but in some situations will keep the accuracy from decreasing or perform equal to the federated learning algorithms. We also found the number of parameters in the hidden layer of the LSTM play a significant role in the performance of the model as the accuracy significantly increased when the number of parameters decreased, but then the accuracy did not change between rounds, indicating that there was no training happening on the model.

These experiments showed that while federated and personalized federated learning had an effect on the performance of the LSTM model, the accuracy of the clients was not to a point that could be used for energy forecasting. The highest accuracy of any of the simulated clients was 85%, and the lowest was less than 0.02%. There are many factors that go into the performance of the model, a few of which were explored in this paper such as hidden layer parameters and federated learning. While we did not achieve a high performing model, we showed that it could be possible but that a more complex base model may be necessary. In this project we only used one hidden layer and one method of personalization, but data forecasting is a complex problem, and more research could explore using more complex base LSTM models that can be fine-tuned to the data as well as other personalized federated learning algorithms to see if a usable forecasting model could be created.

References

[1] K. Martineau, “What is federated learning?,” *IBM Research Blog*, Feb. 09, 2021. https://research.ibm.com/blog/what-is-federated-learning

[2] P. M. Mammen, *Federated Learning: Opportunities and Challenges*, Jan. 2021. [Online]. Available: https://arxiv.org/pdf/2101.05428.pdf

[3] H. Mcmahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” Jan. 2023. Available: https://arxiv.org/pdf/1602.05629.pdf

[4] “CEEW - Smart meter data Mathura 2021.tab - Council on Energy, Environment and Water (CEEW) Dataverse,” *Harvard.edu*, 2021. https://dataverse.harvard.edu/file.xhtml?fileId=5425312&version=2.0 (accessed Apr. 20, 2024).

[5] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, “Deep Learning with Long Short-Term Memory for Time Series Prediction,” *IEEE Communications Magazine*, vol. 57, no. 6, pp. 1–6, 2019, doi: https://doi.org/10.1109/mcom.2019.1800155.

[6] A. Tan, H. Yu, L. Cui, and Q. Yang, “IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS 1 Towards Personalized Federated Learning,” Mar. 2022. Available: https://arxiv.org/pdf/2103.00710.pdf

[7] “How do you choose the optimal number of hidden layers and units for an LSTM model?,” *www.linkedin.com*. https://www.linkedin.com/advice/1/how-do-you-choose-optimal-number-hidden-layers-units#:~:text=Hidden%20layers%20and%20units%20are (accessed Apr. 20, 2024).

[8] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, “FedBN: Federated Learning on Non-IID Features via Local Batch Normalization,” *openreview.net*, Jan. 28, 2023. https://openreview.net/forum?id=6YEQUn0QICG

[9] A. Tam, “LSTM for Time Series Prediction in PyTorch,” *Machine Learning Mastery,* Apr. 08, 2023. https://machinelearningmastery.com/lstm-for-time-series-prediction-in-pytorch/