PREDICT AIR PRESSURE USING MODEL-AGNOSTIC FEDERATED LEARNING IN AN EMPIRICAL GRAPH

Aalto University, Espoo, Finland

ABSTRACT

Accurate air pressure prediction is important for scheduling planes in airports. Early works in weather prediction mainly focused on optimization on a single server machine and left out the effect of neighborhoods on a certain data point. In order to monitor and predict air pressure information better, this work proposes a prediction machine using Model-Agnostic Federated Learning in an Empirical Graph. We constructed a networked graph, including local data points as nodes. Through training, validation and testing, we achieved the Federated Learning model by updating the GTVMin regularization on every single node. The experimental result shows a slight improvement in performance in comparison with the traditional local training for each node, which provides a new idea for the weather forecast in general.

Index Terms— Federated Learning, Networks, Personalized Machine Learning, Trustworthy AI

1. INTRODUCTION

Air pressure helps airplanes to produce lift and stay up in the air. The differences in air pressure between above and below the airplane creates a force to lift the wing up. Therefore, understanding air pressure is crucial to take off and land a plane safely, especially in the context of airport monitoring. In practice, high-precision air pressure forecasting can assist airplane scheduling tasks in airports more efficiently.

Previous works in studying air pressure mainly employed the Deep Learning method. The study conducted by Huang Zi-Qi et al. in 2020 combined a long-short term memory (LSTM) model and the multilayer perceptron (MLP) model to train a weather monitoring and prediction system for city buses [1]. With a similar approach, Chen G. et al in 2022 used a long-short term memory (LSTM) neural network to process time-series data within an empirical mode decomposition [2]. Earlier works showed reliable performance and good prediction results. However, all weather data from different stations must be gathered under the same database and server in order to process the training. This requirement leads to a couple of consequences. First, the centralizing data might cause a threat of privacy and security issues. Second, running the training process with all data at the same time is costly and inefficient. Third, gathering data all together has ignored the fact that the physical location has some certain effects to the data differences between two places.

Recently, the Federated Learning method in a networked graph proposes a new opportunity to study the air pressure condition while taking into account the affection of neighborhoods [3]. Therefore this project uses a Federated learning method in a network graph in order to achieve a high-precision Air Pressure prediction from Temperature and Humidity.

The remainder of this paper is organized as follows. Section 2 forms a clear understanding of the problem foundation that led to this work. Section 3 describes the use of Federated Learning methodology to solve the initial problem. Section 4 presents results and its validation for the final choice. Finally, a discussion of the limitations and possible future improvements are highlighted in Section 5.

2. PROBLEM FORMULATION

The dataset in this project is a record of Air Pressure from the top ten largest airports in Finland during a year period from 1st May 2022 to 30th April 2023 [4] [5]. The data were acquired from the Finnish Meteorological Institute (FMI) through their open data observation at ilmatieteenlaitos.fi from a much larger and wider dataset with a total of 12 features [5]. In this project we aim to highlight the crucial affection of humidity and air temperature in Air pressure prediction. A brief summary of the dataset is described in Table 1.

We constructed an empirical graph G, which carries a total of 10 nodes (vertices) V = $\{1,..10\}$ as representatives of 10 airport data points D(i) for i = $\{1,...10\}$. Each node connects to a certain number of similar neighbours through edges. We add undirected *edges* between every pair i, i' $\in$ V of two similar nodes using `kneighbors graph()` method from `scikit-learn` library [6] [3] [7] based on their coordinates through param `coord`. Then we calculated the edge *weight* Aii' using the Euclidean distance between nodes by the parameter `coord`. Figure 4 shows how nodes are connected with *edge* and *weight* based on

	Weather Dataset
Datapoints	10 stations Helsinki-Vantaa airport, Turku airport, Tampere-Pirkkala airport, Vaasa airport, Oulu airport, Kajaani airport, Joensuu airport, Kuusamo airport, Kittilä airport and Inari-Ivalo airport
Resolution	1 hour
Time span	May 2022 - April 2023
Label	Pressure (msl) (hPa): float
Features	Relative humidity (%): int Air temperature (degC): float (–Historical–) Year: int Month: int Date: int Time: str Time zone: str

Table 1: Description of the Weather Dataset used for the problem setup

its coordinates. As a result, we build a connected undirected weighted empirical graph from the origin 10 local data points.

Our hypothesis is that those airports which are geometrically close to each other will share similar air pressure levels and trends. The closer two stations are, the more similarities are found. In contrast, the further two stations are, especially in latitude, the more different in air pressure will show. For example, the air pressure in Turku and Tampere will be similar. The differences between Tampere and Oulu are more visible than between Tampere and Turku.

3. METHODS

3.1. Data preparation

At first, we downloaded the weather dataset from Finnish Meteorological Institute (FMI) weather stations website [5] in 10 latest airport stations in Finland which are: Helsinki-Vantaa airport, Turku airport, Tampere-Pirkkala airport, Vaasa airport, Oulu airport, Kajaani airport, Joensuu airport, Kuusamo airport, Kittilä airport and Inari-Ivalo airport [4]. For each observation, the following parameters are used: instantaneous observations, pressure, relative humidity, air temperature, 1 hour time interval, period 1 May 2022 - 30 April 2023 and the observation name - Figure 1. The observation name and information (latitude, longitude) are acquired from the observation station list, which are available at https://en.ilmatieteenlaitos.fi/observation-stations (FMI, 2023).

The weather dataset D is a collection of n = 10 nodes which represent 10 different airport stations. The original local dataset D(i) at node i is constituted by m = 365(days)

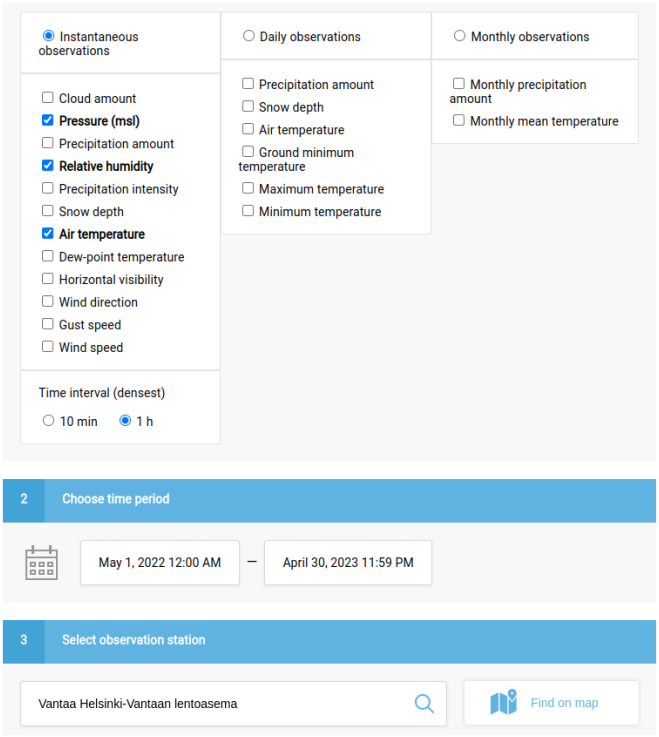

Fig. 1: Download weather dataset for Helsinki-Vantaa airport station from the FMI observation website [5]

* 24(hours) = 8760 data points ((x(i,1), y(i,1)), ... (x(i,m), y(i,m)))

Each data point contains weather features (Air Temperature and Relative Humidity), historical features (Year, Month, Date, Time and Timezone) and the label (Air Pressure) continuously for a one year period from 1 May 2022 to 30 April 2023 - Table 1. To reduce the load of the learning algorithm, we decided to cut down the unnecessary features and focus on the essential ones, which have strong affection on air Pressure. A study on air pressure by M. Raatikainen in 2012 exposed the importances of air Temperature and Humidity on the differential air pressure [8], which shifted our focus on those two crucial features. Because of this reason, we calculated the daily average number, grouped by Air Temperature and Humidity features, for each local dataset. We decided to work on daily data because the amount of hourly data is too large and required a much slower calculation. After preprocessing, each local dataset carries 365 data points as daily data in one year period.

3.2. Train Local Model

Before the training work starts, each local dataset is splitted into three smaller subsets, including the training set {`Xtrain, ytrain`}, the validation set {`Xval, yval`} and the test set {`Xtest, ytest`} in a proportion 60 : 20 : 20 using

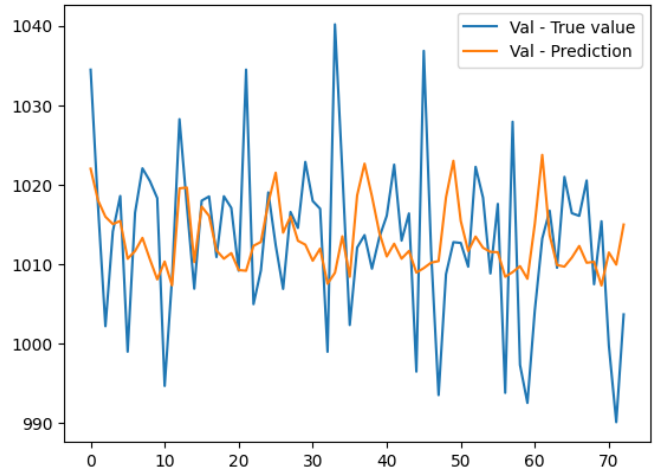

Fig. 2: Testing Linear Regression on Node 0 using the validation set

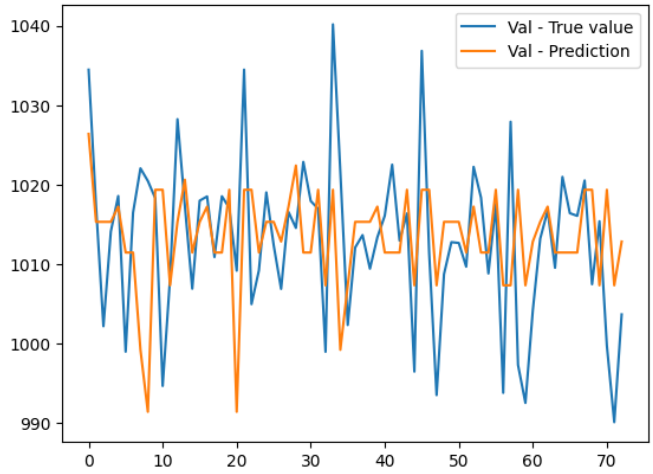

Fig. 3: Testing Decision Tree Regressor in Node 0 using the validation set

`train_test_split()` method from `scikit-learn` library [7]. The training set will be used for learning local hypotheses at the node. The validation set will be used to validate local hypotheses in order to find the best performance, which produces the smallest loss. Finally, the test set is gathered together as a common test set, which is used for final performance assessment to measure the discrepancy between two connected nodes, which has been mentioned in the Section 2 [6].

In each node, we learn a hypothesis individually for its local dataset using the training set. Before Deep Learning and ANN approaches, Linear Regression and Decision Tree methods were popular to deal with the fluctuation of seasonal time-series problems [1] [2]. Therefore, we tried both regressors then measured the average squared error loss using the validation set above all nodes in the empirical graph [6]. We used squared errors loss because those are informative for handling numeric labels in our data points [9] [6] [3]. Each node is attached with a local model - `model`, a training error - `trainerr` and a validation error - `valerr`. The errors are calculated by using the `mean_squared_error` method from `scikit-learn` library [7]. It is seen that Linear Regression produced a slightly smaller validation error. However, we observe from the plots (Figure 2 and Figure 3) that the Decision Tree Regressor is able to capture the variety of the outliner. From this reason, we decided to use Decision Tree Regression as a local model for each node for the following Federated Learning process.

3.3. Model-Agnostic Federated Learning

To learn a local hypothesis map for each node in the empirical graph G, we applied the Model-Agnostic Federated Regression method, proposed by A. Jung et al. (2023) [3] [6]. As we attached a DecisionTreeRegressor to the local model, which is a non-parametric model, this step employs the Algorithm 4 FedRelax Least-Squares Regression from the lecture note [6].

We designed two different Federated Learning (FL) algorithms by constructing two different empirical graphs:

1. FL-1: each node is connected to two nearest neighbors, by adjusting `nrneighbors=2`, seen in Figure 4

2. FL-2: each node is connected to five nearest neighbors, by adjusting `nrneighbors=5`, seen in Figure 5

In both cases, we used the same input parameters as follows: the same empirical Graph G with edge weights Aii', the common test set `gXtest`, regulation parameter `regparam=0.1`, maximum interaction `maxiter=100`.

In each FL algorithm, we run a loop over all connected nodes to define the discrepancy between their predictions. Here, we measured the variation between two models from every two connected nodes on the common test set. The common test set is the collection of 10 *test sets* from 10 local data points. Each pair contains two nodes i, $i' \in G$ in the empirical graph G, which are connected via an *edge* with *weight* Aii'. The edge *weight* Aii' is calculated using the Euclidean distance between nodes as it is convenient to handle numeric data. The discrepancy between nodes is measured through Kullback-Leibler(KL) divergence technical [6] [3] on a common test set, which was defined in Section 3.2. The KL divergence, with a capability to measure the variation of any pair under the same probability space, is suitable for this project, where all local data points are

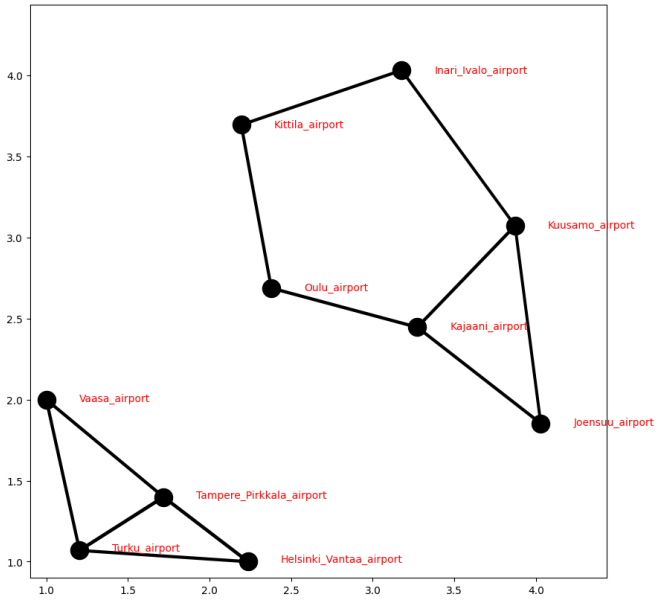

Fig. 4: Plot of airport stations with connected edges by coordinates. Each node connects to two nearest neighbors

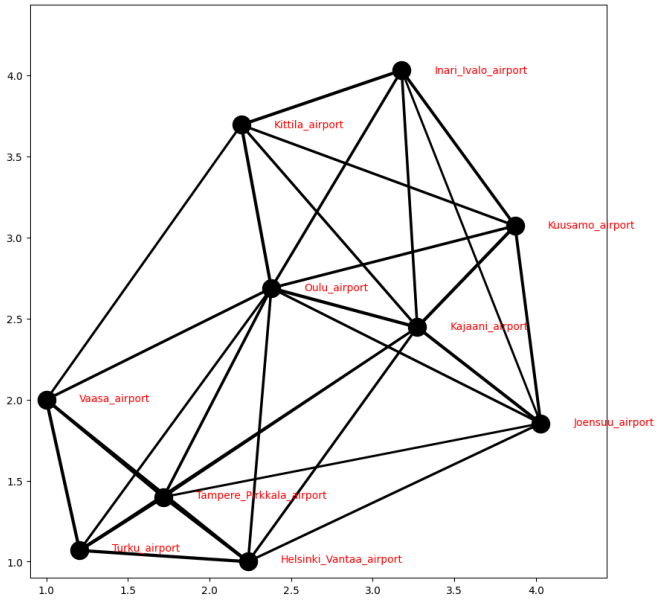

Fig. 5: Plot of airport stations with connected edges by coordinates. Each node connects to five nearest neighbors

under the same probability space with similar statistical information [6] [3].

Here, we defined the Generalized Total Variation (GTV) regularization parameter by summing the discrepancy over all edges in the empirical graph G [6] [3].

To evaluate the result, we calculated the empirical risk via the average loss over the networked graph [6].

Finally, we obtained Generalized Total Variation Minimization (GTVMin) for our non-parametric local models [6].

At this step, the ERM is updated for learning a local hypothesis in a networked empirical map G. Then, we calculated the average mean squared error (MSE) loss over all nodes to compare the performance of two FL methods during training and validation processes. From the training errors and validation errors, we chose the better FL algorithm, which shows smaller errors to process the final test using the common test set - defined in Section 3.2. Here, we measured the test errors as MSE for each individual node and on average above the networked graph. We chose MSE to measure the performance as these nodes carry data points with numeric labels.

4. RESULTS

Table 2 compares the MSE loss for training and validation test set in each node between two design choices (FL-1 and FL-2). Additionally, we calculated the mean of training error and validation error over all nodes in the networked graph. Overall, the result shows that the first FL-1 algorithm produces smaller training errors than the second FL-2. However, the FL-2 obtains better (smaller) validation errors. In comparison with the average MSE from the initial local training models (trainMSE: 84.71; valMSE: 135.36), both FL-1 and FL-2 show a slight reduction on validation errors but the train error increases. As a trade-off, increasing the number of connected edges from FL-1 to FL-2 (from 2 neighbors to 5 neighbors) has almost doubled the amount of running time (from 2.74s to 4.45s).

From the result, we decided to choose FL-2 as the final FL method to continue testing on the common test set as it produced a smaller validation MSE within an acceptable time amount. FL-2 is constructed by setting up each node connected to other 5 nearest neighbors through `nrneighbors=5`. Each individual node carries a DecisionTreeRegressor as a local model with a parameter set up `max-depth=4` and a `weight` of connected edges. The discrepancy from the connected edges has been updated within the GTV regularization parameter and also updated the edge weight. The final regularization - GTVMin param. λ is set as `regparam=0.1`.

We conducted a test of FL-2 on the common test set to validate its performance on unused data labels. The test set error is calculated using MSE for each node in the empirical

	FL-1/trainMSE	FL-1/valMSE	FL-2/trainMSE	FL-2/valMSE
Average over all nodes	85.41	124.04	86.21	116.65
Node 0	77.31	103.63	70.06	90.12
Node 1	89.14	158.84	87.87	140.54
Node 2	74.47	98.72	75.46	104.43
Node 3	97.93	153.27	101.80	140.28
Node 4	96.91	111.86	103.04	97.51
Node 5	109.84	118.29	110.36	115.18
Node 6	69.03	114.50	71.25	105.86
Node 7	70.78	111.55	71.41	108.83
Node 8	85.96	155.72	85.76	149.68
Node 9	83.75	114.00	85.15	114.10

Table 2: Compare Mean Squared Error for training and validation in each node in two FL algorithms.

graph, shown in Table 3. The average test error over all nodes is reported as 137.80. In overall, the test errors are higher than the validation errors seen in Table 2.

	testMSE
Average over all nodes	137.80
Node 0	104.18
Node 1	157.12
Node 2	114.46
Node 3	146.68
Node 4	137.19
Node 5	157.28
Node 6	118.21
Node 7	125.46
Node 8	173.13
Node 9	144.22

Table 3: Measure testMSE for FL-2 algorithm on the common test set

5. CONCLUSION

In this paper, we proposed an implementation of Federated Learning method over an undirected empirical graph to predict Air Pressure for 10 largest airport stations in Finland. To protect privacy and reduce the heavy centralized traditional learning process, we decentralized the dataset in each separate node in a networked graph. From local training to validation to the final testing, we can see the improvement of the Federated Learning solution through the reduction in mean squared error loss. We also recognized the hint of overfitting in some nodes, where the training errors are much smaller than the validation errors. The two design choices confirm the effect of connected edges on the learning process, the more edges give a better performance, which results in a smaller MSE. In closing, our study has a number of limitations. The first limitation is training all local data points with only one Decision Tree Regressor method. In the future, I would like to include different training models for different nodes. Second, this research was limited to testing with data in the duration of one year, which I believe is not enough to handle an annual time-series problem. To improve for future studying, I suggest collecting more data in a wider timeframe. Lastly, using weather data from other locations around the airports would have provided more accurate predictions since the distance between airports can be fairly large.

6. REFERENCES

[1] Zi-Qi Huang, Ying-Chih Chen, and Chih-Yu Wen, "Real-time weather monitoring and prediction using city buses and machine learning," *Sensors (Basel, Switzerland)*, vol. 20, 09 2020.

[2] Guici Chen, Sijia Liu, and Feng Jiang, "Daily weather forecasting based on deep learning model: A case study of shenzhen city, china," *Atmosphere*, vol. 13, no. 8, 2022.

[3] A. Jung, S. Abdurakhmanova, O. Kuznetsova, and Y. SarcheshmehPour, "Towards model-agnostic federated learning over networks," 2023.

[4] Euroopan lentokentät, "Suomen lentokentät suuruusjärjestyksessä (2022)," 2019.

[5] Finnish Meteorological Institute, "Download observations," 2023.

[6] Alexander Jung, *Lecture notes of the course CS-E4740 Federated Learning*, Aalto University, Aalto University, 2023.

[7] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al., "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[8] Mika Raatikainen, Jukka-Pekka Skön, Markus Johansson, Kauko Leiviskä, and Mikko Kolehmainen, "Effects of weather conditions and differential air pressure on indoor air quality," 01 2012.

[9] Alexander Jung, *Machine Learning: The Basics*, Springer Nature Singapore, Singapore, 2022.

Response to the Comments of the Reviewers on the project "Predict Air Pressure using Model-Agnostic Federated Learning in an Empirical Graph"

[redacted]

June 21, 2023

I express my sincere gratitude for the insightful and constructive comments and suggestions provided by the reviewers for my final project in the course Federated Learning, 2023. Major modification I implemented in the revised manuscript and python notebook include the following:

- I have added motivation for choosing KL divergence and Euclidean distance in Section 3.3 Methodology. The choice of edges and weights has been modified to become more visible.
- I have carefully modified the description of data features in Section 2. to make it clearer.
- I have significantly revised the report paper by fixing typo mistakes, reformatting variables with correct font type and swapping to passive tone voice wherever it is possible.
- I have sharpened the conclusion with the expansion of the limitations and future improvement for the local data source.

In the following section, I respond to each individual comment of the reviewers in a point-by-point manner. In order to improve the quality of the project, this letter mentions only those comments which have pointed out the deficiency of the work. I highly appreciate compliments and expression comments, however, those will not be mentioned in order to keep this paper concise.

Comments of Reviewer #1

1.1 The variation of local models is measured by KL divergence. However, why choosing KL divergence is not introduced.

I have added the reason for choosing KL divergence in Section 3.3 due to its capability of measuring the variation of any pair under the same probability space as all local data points are under the same probability space with similar statistical information.

Comments of Reviewer #2

2.1 No. I did not find the clear representation of the weights of the edges. It was mentioned that the graph network was created based on the k-NN (k=2,5). Somehow it was mentioned about the KLD. However, if I understand right, KLD is used to estimate the similarity between the local data-sets. The discrepancy (in the case of non-parametric models) is calculated via predictions, i.e. d(h^(i), h^(j)).

In the original script, I described the graph's construction, including edges and its weight in Section 2. Problem formulation. The edge's weight is calculated using the Euclidean distance between connected nodes. To make it more visible, I have emphasized the use of the Euclidean distance by changing the font

style to highlight this section. I have also added the reason for choosing the Euclidean distance method in Section 3.3.

2.2 In section 3.3. MSE is somehow called as a regularisation?! " The regularisation, known as the average squarred error loss for the labeled test set, ...". In my understanding the regularisation part of GTVmin considers either the difference between the model parameters (parametric models) or the discrepancy between the predictions (non-parametric models).

Yes, there was a mistake in section 3.3. To clarify things, the regularization update has been moved to a separate part before it.

2.3 I would suggest the Author to test graph learning based on KLD and Wasserstein metric to estimate the similarity of probability distributions of the local data-sets.

I thank the reviewer for suggesting a new approach to calculate the similarity of probability distributions. However, due to the limited time and resources, plus the Wasserstein metric concept is fairly new to me, I am not confident to use this method for this project course.

2.4 While reading the report, I did not get the idea why the title proposed included: "model-agnostic". In which way/how this "model-agnostic "approach was implemented/employed. Why the Author should have been highlighted "model-agnostic".

I thank the reviewer for pointing it out. The concept of "model-agnostic FL" originates from an article "Towards model-agnostic federated learning over networks" by Alexanderia Jung et al. (2023). Because the FedRelax algorithm, which I used in this project, is presented thoroughly and completely in this article under an official name "model-agnostic FL", so I borrowed this concept into my report to have a clear reference to the original paper. To make it clearer, I have added a short description about the origin concept and the paper to the beginning of section 3.3.

Comments of Reviewer #3

3.1 There is a brief explanation of the content of the most important parts of the next section.

Yes. In the origin script, the last paragraph of Section 1. Introduction summarized the rest of the paper.

3.2 Perhaps, the features paragraph could be presented more clearer as it is quite cluttered.

I have carefully revised the feature paragraph and modified it to improve the clarity of the description.

3.3 Weights are mentioned. However, no explanation of choice or estimation is clarified in the paper.

The choice of edges and its weights are significantly described in the second paragraph in Section 2. Here, the constructed graph is described with its edges and weights. The edge weight for each edge is

calculated based on the geographical distance between neighboring nodes by “using the Euclidean distance between nodes by the parameter coord.”

3.4 Albeit quite a lot of features and justifications were mentioned sometimes making it a bit confusing.

To clarify the feature selection, I have modified this part as seen in my response to the Reviewer comment #3.2.

3.5 However, the paper could describe in better detail the motivation behind the choice of these models.

I have added one sentence about the advantage of using DecisonTreeRegression for local model as “Decision Tree method was popular to deal with the fluctuation of seasonal time-series problem” and also “Decision Tree Regressor is able to capture the variety of the outliner” in Section 3.2.

3.6 variation of local models are mentioned but no motivation is provided.

See my response to the Reviewer comment #1.1.

3.7 The training and validation sets are explained, albeit the example of splitting could be explained better.

I have added more details about how the splitting method was applied on each local data set in Section 3.2.

3.8 It mentions edges but doesn't go into detail on why they are chosen.

See my response to the Reviewer comment #2.1

3.9 Conclusion is simple but things that were missing can be added and can be formatted better.

I tried to keep a concise and clear conclusion so the reader can understand better. In conclusion, I have self-criticized for the limitations and provided a basis for further research. The results were discussed and the meaning of those were explained. It would be helpful for improvement work if the reviewer could point out the missing point.

3.10 Paragraphing can be improved.

I thank the reviewer for commenting on the paragraphing. However, it is difficult for improvement work to not know which part and which paragraph should be improved. It would be more helpful if the reviewer could point out the specific section.

3.11 However, it is lacking in motivation for things like weight.

See my response to the Reviewer comment #2.1.

3.12 I suggest working on improving the conclusion in this regard to have a paper that is better than the one you have.

See my response to the Reviewer comment #3.9.

Comments of Reviewer #4

4.1 What I would fix would be the use of we as for me academic writing should be in passive form.

I have carefully reviewed the report and switched to the passive form in the possible places, mostly in Section 3. Methodology. However, many other parts required its specific subjects, which are difficult to change to passive form.

Comments of Reviewer #5

5.1 Only some of the variable titles were not formatted correctly.

I have carefully reviewed the report and reformatted some variable titles using monospace font.

5.2 Maybe using weather data from other locations around the airports would have provided more accurate predictions since the distance between airports can be fairly large. This could be done in further research.

I thank the reviewer for suggesting a new approach to improve the learning algorithm. However, due to the limited time and resources, I am not capable of recollecting local data sources from other locations. I have added this idea as a future improvement plan in the conclusion.

Comments of Reviewer #6

6.1 The quality of the discussion of the obtained results is not mentioned in the provided text.

See my response to the Reviewer comment #3.9.

6.2 However, some aspects, such as the discussion of obtained results and reproducibility of numerical results, could be further improved

See my response to the Reviewer comment #3.9.

References

1. Response to the Comments of the Associate Editor and the Reviewers on Manuscript SPL-28200-2020, “Local Graph Clustering with Network Lasso”. Jung, Alexander. Aug 13, 2020. Available at

https://github.com/alexjungaalto/FederatedLearning/blob/main/material/ResponseLetterSPL-28200-2020.pdf