

Federated Learning and LLM for Privacy-Preserving Wind Speed and Wind Power Forecasting

Miras Shaltayev, Zhansaya Sabitova, Amina Alisheva, Nurdana Razakhbergen, Aldiyar Kezhenov

Department of Computer Science

Nazarbayev University

Astana, Kazakhstan

miras.shaltayev@nu.edu.kz, zhansaya.sabitova@nu.edu.kz, amina.alisheva@nu.edu.kz,

nurdana.razakhbergen@nu.edu.kz, aldiyar.kezhenov@nu.edu.kz

Supervised by:

Anh Tu Nguyen

PhD, Assistant Professor

Department of Computer Science

Nazarbayev University

Astana, Kazakhstan

anh.nguyen@nu.edu.kz

Ton Duc Do

PhD, Full Professor

Department of Robotics and Mechatronics

Nazarbayev University

Astana, Kazakhstan

doduc.ton@nu.edu.kz

I. EXECUTIVE SUMMARY

This project examines a privacy-protective method of wind speed prediction based on federated learning and time-series models based on LLM. The central issue that is considered is that it is understandable that large and heterogenous datasets are often useful to accurate forecasting, whereas in practice in wind energy contexts the raw data of wind farms and meteorological stations cannot always be centrally stored because of privacy, ownership, and trust considerations.

To overcome this, we architected and deployed a centralized and federated forecasting framework, where users have the capability to upload data, set training parameters and run predictions via a web-based dashboard, and federated clients only send updates to models to a central server. This system is tested on the datasets of Kazakhstan, Vietnam, Greece, and a cross-country capitals dataset with various backbones of the LLM, fine-tuning methods, federated aggregation strategies, and forecasting horizons.

The results show that the proposed computing-based solution is feasible and promising. Overall fine-tuning was the most effective, GPT2 the most effective forecasting model of the tested backbones, and FedProx the most effective federated aggregation model in the reported experiments. In a variety of environments, federated learning performed at near-centralized learning, demonstrating that it is possible to maintain high forecasting performance and data locality simultaneously.

Overall, the project shows how to design, implement and test a computing-based solution to solve a real world problem of forecasting renewable energy by balancing predictive accuracy and privacy. The project code can be found at Github and the web-dashboard is deployed

II. INTRODUCTION

Wind speed forecasting is a crucial element in the successful functioning of modern renewable energy systems and facilitates larger uses of wind-energy. Credible predictions enhance grid stability, can aid power scheduling, and contribute to decreasing uncertainty in systems with high shares of energy sources that are dependent on weather conditions [2], [4]. With renewable energy becoming more all-encompassing, forecasting is starting to be perceived not as a solitary wind-power effort, but as a segment of a more generalized weather-driven power structure that influences planning, operations, and choices throughout the energy system, overall [2].

The traditional ways of wind forecasting have been replaced by machine learning models and deep learning models that can predict the complex and nonlinear temporal relationships in meteorological data. Transformer-based and large language model (LLM)-based time-series models have gained interest more recently due to their capability to incorporate long-range dependencies and learn rich sequence representations. Research like GPT2 and other transformer-based forecasting papers indicate that this type of model can be well applied to time-series prediction tasks, like wind speed forecasting [7].

Even with these developments, much of the existing high-performing forecasting models continue to rely on centralized training on large and heterogeneous datasets. This constitutes a serious limitation in the real-life situations of wind energy. Information gathered at wind farms and SCADA systems, and at meteorological stations can be commercially sensitive or owned privately or limited by data-sharing policies, which makes it challenging to directly aggregate raw information across organizations [4], [5]. Despite the benefits of distributing datasets, centralized learning is not always feasible

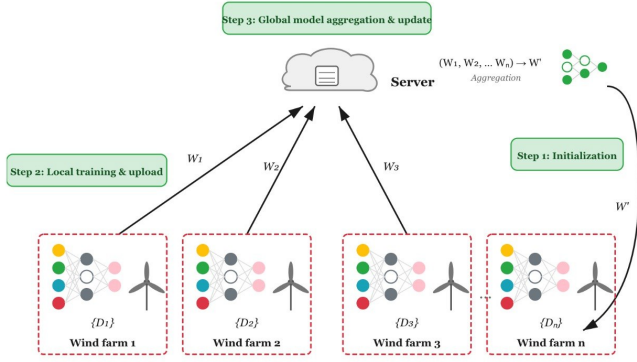


Fig. 1: Illustration of the federated learning process

in practice due to privacy, ownership, and trust issues even in real-world applications although such a combination may enhance model generalization and forecasting accuracy [4].

Federated learning offers a chance to resolve this conundrum as it allows collaborative model training with no movement of raw data across local sites. The process, as shown in Figure 1, starts with the server setting up a global model and sharing it with the involved wind farms. Each client then trains the model locally on its own data, and transfers only the new model weights or parameters back to the server, instead of their raw data. The server then takes these local updates, which in the last step combine to form a more accurate global model, which is then once again sent to all the trainees in the next round of training. This manner maintains data locality and also provides clients the benefit of joint training in federated learning.

Inspired by such progress, this project explores a privacy-aware forecasting system, which integrates federated learning with time-series modeling based on LLM to forecast wind speed. It aims to understand whether it is possible to use more sophisticated transformer-based models in a decentralized learning environment and maintain the privacy-benefits of federated training. Along with the forecasting pipeline itself, the project contains a complete web-based dashboard that facilitates both centralized and federated processes, enabling users to upload data and train settings, as well as to visualize forecasting outputs in an easy-to-use interface. By so doing, the project is concerned not just with predictive performance, but also the design of a realistic end to end system which can be used and deployed in practice.

III. IDENTIFICATION OF CONSTRAINTS

Throughout the project, the design, implementation, and evaluation of proposed federated learning and LLM-based wind forecasting system were affected by several limitations.

A. Economic and Computational Constraints

Economic and computational limitations had occurred in the project. The proposed system had to be trained on several GPUs and needed to have enough memory and a constant

access to compute to be utilized in experiments. In fact, the team only had access to three shared ISSAI GPUs and utilized one local PC in the development and a portion of the training. The amount of experiments and training rounds, as well as the optimization of hyperparameters, were minimized as these resources were not allocated to the project fully. This resulted in some design decisions that were based on feasibility and effective implementation as opposed to the best model performance.

B. Technical Complexity Constraints

Technical complexity was also a constraint in the project. The system in our project incorporates many issues, including federated learning, transformer/LLM-based forecasting, local preprocessing pipelines, global aggregation, and user dashboard. Combining these parts into a working end-to-end prototype complicated the process even more and required taking into account decisions in design to fit within the available time and resources to make the system viable and achievable. As discussed in relevant literature, the use of developed strategies of federated learning coupled with existing systems of deep learning can lead to improved predictive accuracy, however, it also introduces complexity to systems and their use [6].

C. Data Heterogeneity Constraints

Another limitation to the project was due to the statistical heterogeneity of the data. Clients would usually possess non-IID data in the federated learning processes, and it is very true in the case of wind forecasting, where weather patterns differ greatly among various regions. This heterogeneity can lead to stalling of convergence, decreased global model stability, as well as unequal customer performance. This system thus had to be developed keeping in mind that team based training would be more difficult in such decentralized environment [4], [6].

D. Privacy and Ethical Constraints

There were also major constraints in terms of ethics and privacy. Some of the driving forces of the project included that the meteorological and operational information is likely to be classified, proprietary or forbidden to be exchanged directly between organizations. This does not allow for a completely centralized approach to training and greatly influenced the design of the system architecture [4], [5]. In the meantime, the integration of the state-of-the-art deep learning models presented a predictive performance versus interpretability trade-off as more complex models are more difficult to explain empirically and audit.

E. Environmental and Sustainability Constraints

Environmental factors as well as sustainability were also in play. Training Deep learning is computationally intensive, which is another factor that constrained the scale of experimentation practicability, given the hardware capabilities. Simultaneously, the larger aim of the project is consistent with the sustainability objectives, as better forecasting of

wind speed and wind power can facilitate more successful integration of renewable energy in the modern power systems. [1], [2].

F. Health and Safety Constraints

Last but not least were health and safety limitations which were indirect yet applicable. The project is not directly physically dangerous in terms of its development. Nevertheless, the wrong forecasts may adversely affect the operation planning of power systems in a real deployment situation. This is why this system should be considered as a decision-support tool instead of a fully independent functioning controller unless it is highly tested in the real world. [2].

All in all, these listed limitations affected the project, reducing the scale of experiments, adding complexity to implementations, and creating significant trade-offs among privacy, accuracy, interpretability and practicality. They also point out that though the prototype developed is technically feasible, more resources, validation, and infrastructure would be needed to build the prototype in the real world at scale.

IV. BACKGROUND AND RELATED WORK

Climate change is causing the global to move towards the use of renewable energy sources as opposed to the reliance on fossil fuels. Wind energy is widely considered as an essential source of clean energy, with a high degree of technological maturity level of wind energy being a characteristic feature of this type of energy source [1]. The accurate forecasting of wind velocity and energy production is an important factor in sustaining stability of renewable energy systems, streamlining grid operations, and sustainable power production.

Initial methods of wind forecasting were based on classical statistical and physical models, such as autoregressive models and numerical weather prediction methods. These techniques were subsequently enhanced with machine learning algorithms including support vector machine and tree-based algorithms, which performed better by learning linear and nonlinear correlations in wind dynamics. More recently, deep learning systems, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have become popular with their capabilities to capture complex temporal dynamics in time series data.

Models based on transformers were an advancement in the field, with the long-range dependencies capture capability, string performance on a wide variety of forecasting tasks. Following this development, recent research has demonstrated that large language models (LLMs) can be used, which were originally developed to aid natural language processing, to solve time series forecasting. In particular, models, such as GPT4TS, exhibit high-performance in terms of pretraining, transfer learning and improved representation capabilities, enabling both fine-tuned and zero-shot forecasting learners capabilities [7].

Although these innovations have been made in model operation, the real implementation of wind forecasting systems remains a major challenge with regard to data privacy and

availability. The common method of centralized training involves the combination of raw time-series data of several wind farms, creating the risks of data leaks and exposure of commercially sensitive data. These risks can expose location-specific operational features and business-related knowledge, and data sharing is a significant obstacle to collaboration [2], [3]. Consequently, privacy-preserving methods such as secure aggregation, encrypted training, and distributed optimization have been frequently studied [2]–[4].

Besides technical issues, ownership and trust became other limiting factors to centralized approaches. The operators of wind farms are usually reluctant to release any historical data to an outside party or a competitor due to the constraints imposed by the regulations and the market competition conditions [2]. The centralized models need a trusted aggregation, and often, such trust is not guaranteed in practice [2]. These limitations underscore the importance of collaborative learning frameworks that do not violate privacy and ensure better forecasting performance can be implemented [3], [5].

Federated learning (FL) has become an promising paradigm to overcome these issues through providing collaborative model learning without the exchange of raw data. Models in FL are trained locally on each client (e.g., wind farm) and only model updates are aggregated, maintaining data privacy. The previous research has been able to use FL in the context of wind and energy forecasting tasks, illustrating how decentralized training can provide equivalent performance to centralized methods, as well as ensuring data locality. Nevertheless, the majority of the existing research concentrates on rather limited model designs, e.g., LSTM or Transformer-based models, and tends to be applied to a specific problem (e.g., non-IID data, communication efficiency) [6].

Concurrently, recent studies in similar fields have investigated more advanced federated learning methods, such as multitask learning, mixture-of-experts models, and multi-server training systems to enhance robustness and address data heterogeneity [6]. Nonetheless, the incorporation of current high-capacity models, including the time series architecture based on LLM, into the federated learning models of wind speed prediction is underexplored.

The gap demonstrates an important research direction. Although the models based on LSTM show high performance in centralized settings, they are not well studied in the context of federated learning limitations. Specifically, systematized comparison of high-performing modern architectures and established baselines in both centralized and federated environments is in demand. This analysis will be critical towards comparing trade-offs among accuracy, privacy and scalability, and determining the feasibility of deploying advanced models in real-world wind forecasting settings.

V. PROJECT APPROACH

A. Problem definition

Federated learning (FL) learns a shared model with multiple clients without transferring their data to a central server. All the clients train locally and only weights of models are shared.

In the case of wind-speed prediction, this is important since wind-farms are not usually willing to provide raw data of their operations to external parties. The training cycle is simple. The server starts a world model and transmits it to clients involved. Every client learns on its local data and sends the updated weights back. The server aggregates those weights — using one of several strategies described below — and redistributes the updated model. This repeats for a fixed number of rounds, or until validation performance stops improving.

The arrangement is mutually beneficial: every farm receives a model refined on data from multiple wind sites, without any site exposing its own measurements. The cost is that wind data from different locations follows different distributions, which complicates aggregation and is the central challenge this project addresses.

We implement forecasting models across three horizons: short-term (1 step), medium-term (72 steps), and long-term (432 steps).

B. Datasets

Each dataset was divided into training, validation and testing (70%, 10% and 20% accordingly). The dataset was standardized using `StandardScaler`, fit on the training data to avoid data leakage.

1) *Kazakhstan meteorological dataset (KZMET)*: The biggest emitter of CO₂ in Central Asia is Kazakhstan and its carbon intensity of GDP is 86% higher than the world average [18]. The country’s electricity was constructed on the principle of a centralized coal generation, and that history is still evident in grid infrastructure. Wind development has historically had limited room within it, however investors’ plans are shifting. For example, in 2024, capital supply exceeded demand six times [19].

The measurement of the wind speed in Kazakhstan was taken from the POWER project of NASA (Prediction Of Worldwide Energy Resources) [8], specifically Hourly v10.2.1, accessed on 2025-10-10. POWER provides solar and meteorological parameters derived from satellite observations instead of ground. It is suitable for Kazakhstan because there is little meteorological coverage and satellite-derived estimates can be used to fill in those gaps. We extracted wind speed at 50 m above ground over the period 2010-01-01 to 2025-07-31 for five cities — Almaty, Zhezkazgan, Aktau, Taraz, and Aktobe — each treated as a separate federated client. At hourly resolution each city contributes 136,605 samples. The dataset carries no access or redistribution restrictions.

2) *Vietnam meteorological dataset*: Vietnam has over 3,000 km of coastline within a monsoon climate zone, which gives it among the strongest and most consistent wind resources in Southeast Asia [20]. Both onshore and nearshore sites are commercially viable, and the country has expanded its installed wind capacity substantially over the past decade.

The Vietnam dataset comes from the campaign “Wind Measurement for Planning and Development of Wind Power Projects,” conducted by Power Engineering Consulting Joint Stock Company 3 (PECC3) and commissioned by the

Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ) [9]. Wind speed was recorded at 80 m above ground at 10-minute intervals across ten sites from 2012 to 2018. We use five of those sites as federated clients: An Ninh Dong, EaDrang, Da Loan, My Thanh, and Thanh Hai. Total samples across all five clients: 1,286,352. The data is published on the World Bank’s ENERGYDATA.INFO platform with no access restrictions.

3) *Greece meteorological dataset*: Greece has been expanding its renewable energy capacity substantially, with on-shore wind farms operating across its islands and mainland, benefiting from strong and consistent Aegean wind resources. The Greece dataset originates from the SCADA system of an onshore wind farm located on a Greek island, operational since 2009, collected over a six month monitoring period from 1 January 2020 to 30 June 2020 [21]. The dataset was produced under the UNDERPIN project, supported by the European Commission’s Digital Europe Programme and the Austrian Research Promotion Agency, and was published by Motor Oil Hellas in collaboration with the AIT Austrian Institute of Technology. Operational data were recorded at 10-minute intervals across ten wind turbines, comprising approximately 26,000 measurements from 131 sensors and 24,000 event log entries per turbine, covering parameters such as wind speed, pitch angle, power output, generator speed, and bearing temperature. Total samples across all ten clients: 262,080. The dataset is publicly available on Zenodo under a Creative Commons Attribution 4.0 license with no access restrictions.

4) *Capitals dataset (CAPITALS)*: To complement the country-specific analysis, we constructed a second dataset from five capital cities located in different parts of the world: Astana, Washington, Paris, London, and Canberra. As with KZMET, wind speed measurements were obtained from NASA’s POWER project (Prediction Of Worldwide Energy Resources) [8], using the Hourly v10.2.1 product. We extracted wind speed at 50 m above ground for a one-year period, treating each capital as a separate federated client.

The motivation for this dataset differs from that of KZMET. Whereas KZMET represents geographically related clients within one country, CAPITALS was designed to examine federated learning under substantially greater spatial and climatic diversity. These five capitals belong to different regions of the world and are shaped by different large-scale weather systems, seasonal cycles, and local wind regimes. As a result, the clients are expected to exhibit more heterogeneous temporal patterns than those in a single-country setting.

This setup allows us to study how federated learning performs when the participating clients come from regions with markedly different wind characteristics. In particular, the CAPITALS dataset is intended to test whether federated models can learn useful shared structure across clients despite strong regional variation, and how such heterogeneity influences forecasting performance. Therefore, this dataset serves as a more challenging benchmark for evaluating generalization in federated wind forecasting.

C. Model Architecture

1) *GPT4TS (GPT-2 decoder)*: GPT4TS [7] adapts a pre-trained GPT-2 for time series analysis by freezing its self-attention and feed-forward layers and training only positional embeddings, layer normalization parameters, and a linear projection head. Zhou et al. argue that the attention patterns GPT-2 acquired during language pretraining are general enough to carry over to sequential numerical data, making task-specific architectural changes unnecessary. The paper — a NeurIPS 2023 Spotlight — evaluated this single frozen backbone on long- and short-term forecasting, classification, anomaly detection, imputation, and few-shot and zero-shot learning, and found it outperformed specialized models on most of those tasks.

2) *BERT*: BERT [10] (Bidirectional Encoder Representations from Transformers, NAACL 2019) pretrains a transformer encoder using Masked Language Modeling and Next Sentence Prediction. Unlike GPT-style models, which process tokens left-to-right, BERT conditions on both past and future context at every layer. For forecasting, we attach a regression head to the encoder output. Two configurations exist: BERT-Base (12 layers, 110M parameters) and BERT-Large (24 layers, 340M parameters).

3) *LLaMA*: LLaMA [11] is a family of decoder-only language models released by Meta AI in February 2023, ranging from 7B to 65B parameters. All models were trained on publicly available text corpora, with no proprietary data. LLaMA-13B outperforms GPT-3 (175B) on most standard benchmarks; LLaMA-65B is competitive with Chinchilla-70B and PaLM-540B. We adapt LLaMA for time series forecasting using the same frozen-backbone approach as GPT4TS: most weights remain fixed, with lightweight task-specific components trained on top.

4) *BART*: BART [12] is a transformer denoising auto-encoder that pairs a bidirectional encoder with an autoregressive decoder. Pretraining corrupts input sequences through sentence shuffling and span masking, then trains the model to reconstruct the originals. For time series forecasting, we use BART as a sequence-to-sequence model: the encoder processes the input window and the decoder generates the forecast horizon.

5) *QWEN*: Qwen [22] is a large language model introduced by Alibaba Cloud, which was used in this work as the foundation of instruction generation. Qwen adopts the ChatML format, which allows describing metadata and content of a dialogue, effectively distinguishing between different types of information and improving the capacity of the model to process complex information. Qwen uses supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) methods to align, and is trained as a dialogue model based on its base model. Among all the open-source LLMs evaluated on instruction generation, Qwen achieved the strongest performance, leading GPT3.5 as the only closed-source alternative.

6) *OPT*: OPT (Open Pre-trained Transformer) [23] is a suite of decoder-only transformer language models that were released by Meta AI, with 125M to 175B parameters, is

a reimplementation of GPT-3 performance and scale that is fully open to the research workforce. The models were pre-trained with about 180B tokens inspired by a concatenation of RoBERTa corpora, a subset of the Pile, tokenized with the GPT-2 byte-level BPE tokenizer. Zero- and few-shot performance on 16 standard NLP benchmarks demonstrate that the performance of OPT-175B in general matches GPT-3 at similar parameter counts, but one- and few-shot results fall slightly behind. We use the facebook/opt-350m as part of an Opt Linear setup, meaning that time series inputs are partitioned into patches, mapped to the embedding space and token sequences are computed. A linear layer is used to map the decoder outputs to the forecast horizon.

7) *GEMMA*: Gemma [24] is a collection of lightweight open models based on the Gemini research program, designed for high-performance language reasoning and safety. It uses a decoder-only architecture with extensions such as GeGLU activation functions, RMSNorm for stable sub-layer normalization, and rotary positional embeddings (RoPE). The model family, encompassing 2B and 7B parameter variants, was trained on up to 6 trillion tokens of primarily English web documents, mathematics, and code. For the purposes of this work, we adapt Gemma-3-270M for time series forecasting using a frozen-backbone approach. The pretrained transformer weights remain fixed, while lightweight task-specific components — a ConvMLP patch embedding layer, input projection, and output prediction head — are trained on top.

D. Flower framework

1) *FedAvg*: FedAvg [13], proposed by McMahan et al. at AISTATS 2017, is the foundational federated learning algorithm. Each round, the server distributes the current global model; each client runs stochastic gradient descent for multiple local epochs; the server then averages the returned weight updates. The approach requires 10-100 times fewer communication rounds than naive distributed SGD. Its well-documented weakness is statistical heterogeneity: when client data distributions diverge, the averaged model degrades. The remaining four strategies below all address this problem in different ways, with FedAvg as the common baseline.

2) *FedPer*: In FedPer [14] the network is partitioned into personalization layers and base layers. Base layers are aggregated at each round globally, similar to FedAvg; personalization layers are maintained completely local to individual clients and never shared. Every client thus gets a representation in the global model trained on all the clients but a private head is kept to its own distribution. The technique was tested on non-IID partitions of CIFAR-10 and a personalized image aesthetics dataset.

3) *FedBN*: FedBN [15] (ICLR 2021) handles a particular form of heterogeneity: feature shift, whereby various clients are recording information in different conditions and is not simply data with varying class distributions. The solution is not to aggregate batch normalization layers, while all other parameters are aggregated in the usual way of FedAvg. Li et al. [15] demonstrate that this performs better than FedAvg

and FedProx in speed of convergence and ultimate accuracy on a variety of benchmark and medical imaging data, and has theoretical convergence guarantees. In the case of wind speed prediction, with a farm-specific base statistics (mean speed, variance, seasonal shape) it is beneficial to keep BN local so that each client becomes conditioned to the feature distribution of its farm instead of having to assume the global mean.

4) *StatAvg*: StatAvg [16] treats the problem of heterogeneity as a normalization problem. All clients calculate the local mean and local variance before training and forward this information to the server. The server combines them into global statistics of the actual combined distribution, and then allocates them back to be universally normalized. Since this occurs prior to the FL training loop, StatAvg can be used with any aggregation strategy - it does not replace FedAvg or FedProx but is a preprocessing step that can run concurrently. The approach was first suggested to federated intrusion detection, where the distribution of network traffic changes across client nodes, the mechanism is adapted to wind speed data, where differences in distributions across sites are the key issue.

5) *FedProx*: FedProx [17] (ICLR 2020) generalizes FedAvg by adding a proximal term to each client’s local objective:

$$h_k(w; w^t) = F_k(w) + \frac{\mu}{2} \|w - w^t\|^2 \quad (1)$$

The term $\frac{\mu}{2} \|w - w^t\|^2$ penalizes divergence from the current global model w^t , constraining local updates under both statistical and systems heterogeneity. FedProx also tolerates partial work from slower clients: instead of waiting for all clients to finish or dropping stragglers, it accepts and aggregates partial solutions. Setting $\mu = 0$ recovers FedAvg exactly. In heterogeneous settings, FedProx improves average absolute test accuracy by 22% over FedAvg and includes theoretical convergence guarantees for the non-IID case.

E. Centralized Baseline

In federated learning, each client owns their data, and their data is kept private. The server only aggregates model weights. In the centralized baselines, all clients’ data is pooled, and a single server model is trained on them. However, the evaluation is done on a per-client basis so the results are directly comparable to the federated results. Only after that are the errors averaged.

In federated learning, we define a specific number of rounds and local epochs. To match the same training volume, the centralized model trains for number of rounds $\times$ local epochs total epochs. Other parameters, such as the input sequence and prediction length, are kept the same for both centralized and federated strategies.

Each client’s dataset is also divided into training, validation, and testing sets (70/10/20). The training data of each client is directly concatenated and shuffled. Concatenation is not applied to the testing and validation sets because the evaluation is done on a per-client basis, as mentioned earlier. The data is scaled using the `StandardScaler`, and the scaler

is fit only on the training subset to prevent data leakage. Furthermore, each station gets its own scaler instance, ensuring that scale differences between clients are normalized away independently.

F. Web-Based Dashboard

The project also has a web-based dashboard which combines the forecasting pipeline into an easy to use application to make the proposed solution practical. The dashboard is compatible with centralized and federated learning. Using this interface, people will have an opportunity to load CSV data, select the forecasting model, set prediction length, configure federated learning parameters, including aggregation algorithm and client count, and see the forecasting results. This element will make the project to go beyond model experimentation as illustrated in Figure 2; this is because it will reflect an end to end computing-based solution which integrates machine learning, system integration, and user interaction.

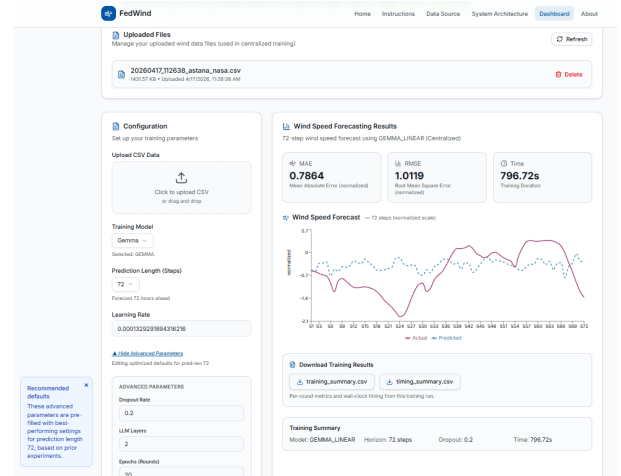


Fig. 2: Dashboard page of the FedWind website.

VI. PROJECT EXECUTION

During the two semesters, the project evolved through a first-stage system integration phase to a more specific experimental and application-focused development phase.

Throughout the first semester, the whole team worked on the integration of the system based on the Flower federated learning framework. The stage consisted of creating an end-to-end pipeline, such as model initialisation, local training processes, aggregation policies, and evaluation processes. Much has been done to make sure that the various aspects, such as datasets, models, and federated learning algorithms, can work together in a single architecture. Since the concept of integrating federated learning with LLM-inspired time-series models is inherently complex, this phase focused on system stability and practicality, rather than a broad-scale experiment.

During the second semester, the tasks were split into two main directions. Three team members worked on experimental testing, which involved testing a variety of datasets, model architectures, and federated learning strategies. The other two

members focused on the creation of a web-based application, frontend and backend programming, and the incorporation of forecasting model into user-friendly application. Other interactive features were also adopted as part of the web-based application such as a discussion board where users could ask questions and feedback, as illustrated in Figure 3. This split allowed the project to work in parallel on both the research and system deployment fronts.

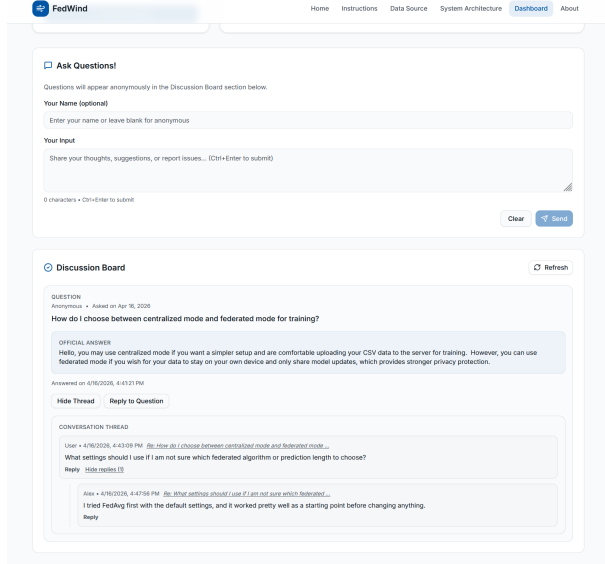


Fig. 3: Discussion board feature of the FedWind website.

Regardless of this specialization, there was still co-operation within the team. The final report was prepared by all its members, and the following are the tasks that were allocated: Amina Alisheva worked on the Introduction and Background section, Zhansaya Sabitova worked on the Constraints section, Miras Shaltayev worked on the Project Setup section, Aldiyar Kezhenov worked on the Evaluation section, and Nurdana Razakhbergen worked on the Project Execution section.

A number of design choices and changes occurred during the project. First of all, the experiments have been performed on the data of the Kazakhstan meteorological (KZMET) only. The scope was later broadened to incorporate other datasets, i.e. the Vietnam meteorological dataset (VNMET) and the Greece wind dataset, and some supplementary data that was acquired at NASA. This growth allowed a more detailed assessment with non-homogenous data distributions, which is a major challenge in federated learning settings.

Besides increasing the size of the dataset, the project also included an expanded set of models, such as transformer-based and LLM-inspired ones. The group also investigated various fine-tuning algorithms and federated learning methods such as baseline aggregation algorithms and personalization algorithms to enhance the performance and resilience of the models in non-IID setups.

During the implementation stage, there were a number of problems. The size and amount of experiments that could be conducted was limited by computational resource constraints.

The integration of federated learning with multiple deep learning models was quite technical and demanded considerable time to be developed and well coordinated. Moreover, the presence of statistical heterogeneity among datasets led to instability during training and necessitated further experimentation of aggregation strategies.

These issues have been solved by conducting experimentation, prioritizing implementations that are feasible and solving problems together. The team varied by constraining the scope of experiments where required, experimenting with alternative methods and progressively improving the system.

Generally, the project showed good teamwork and each member had well defined roles and well coordinated the project. The joint efforts and the expertise in their respective areas allowed the effective creation of the experimental framework and the application prototype.

VII. EVALUATION

The proposed system was tested to answer whether the system addresses the issue mentioned in the introduction, i.e., whether it is possible to predict wind speed and wind power accurately in a privacy-preserving environment without the need to share raw client data centrally. The assessment was thus targeted at making quality predictions in a centralized and federated learning environment. To be more precise, the assessment was aimed at addressing the following questions:

- 1) Are the time-series prediction models built using LLM capable of high predictive performance on wind data?
- 2) Does FL retain sufficient predictive accuracy to be practical in comparison to centralized training?
- 3) What is the best trade-off between accuracy and practicality of fine-tuning strategy, LLM backbone, and FL algorithm?
- 4) What is the strength of the proposed solution to a change of the number of clients or the participation rate?

This assessment directly supports the computing-based solution due to it quantifies the accuracy of the proposed federated forecasting pipeline given the same privacy constraints that drive the project. That is, it is not just to demonstrate that the system is running, but that it makes valuable forecasts and maintains data that is local to the client.

A. Evaluation Metrics

Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) were the measures to assess the performance forecasting. These are the typical deterministic forecasting metrics and are well suited in this project since all models are tested on the same target variables and under the same experimental circumstances.

$$\text{MAE} = \frac{1}{N} \sum_{t=1}^N |\hat{y}_t - y_t| \quad (2)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{t=1}^N (\hat{y}_t - y_t)^2} \quad (3)$$

MAE is used to estimate the average magnitude of forecasting error and it is straightforward to interpret in the original scale of the target variable. RMSE also punishes large errors more than small ones and this can be particularly helpful in operational forecasting problems where large errors can be more damaging than lots of small ones. A combined assessment with both metrics thus provides a more balanced assessment of model quality.

B. Evaluation Protocol

All the evaluation sections were performed in 12 experimental conditions, which were 4 datasets and 3 prediction lengths (1, 72 and 432). In each setting, the performance of each model was evaluated on the held-out test set using MAE and RMSE.

The assessment involved the following five elements:

- 1) comparison of fine-tuning strategies,
- 2) comparison of backbones of LLM,
- 3) comparison of federated learning algorithms,
- 4) impact of changing the number of clients,
- 5) impact of changing participation rate.

As the main body would otherwise be unnecessarily large to report all the raw results, compact summary tables are provided in the main text, including highlighting the most interesting trends. Detailed per-setting results can be placed in the appendix if needed.

C. Evaluation of Fine-Tuning Strategies

The initial section of the assessment looked at how various fine-tuning tactics influenced performance prediction. This comparison is significant as the project will use pretrained models based on LLM, and their effectiveness will rely on the possibility to adjust them to time-series forecasting tasks. The strategies considered included Full Fine-Tuning (FFT), Partial Fine-Tuning (PFT), LoRA, and AdaLoRA.

To make this comparison experimentation was done with the GPT based model in both centralized and federated configurations on all 12 combinations of datasets and horizon. Table I summarizes the MAE of each fine-tuning strategy, instead of enumerating all 12 raw results in the main text.

TABLE I: Comparison of Fine-Tuning Strategies: Centralized vs. Federated MAE

Dataset	Horizon	FFT Pretrained		FFT Not Pretrained		PFT Pretrained		PFT Not Pretrained		LoRA		AdaLoRA	
		Cent.	FL	Cent.	FL	Cent.	FL	Cent.	FL	Cent.	FL	Cent.	FL
KZMET	1	0.1480	0.1617	0.1210	0.1517	0.1144	0.3486	0.1260	0.3298	0.1236	0.2699	0.1163	0.3267
	72	0.6222	0.6507	0.6760	0.6555	0.6492	0.6928	0.6768	0.6921	0.6988	0.7111	0.6553	0.9953
	432	0.6688	0.6913	0.7087	0.6911	0.6955	0.7181	0.7093	0.7179	0.7306	0.8395	0.7015	0.9084
VNMET	1	0.2010	0.2039	0.2004	0.5199	0.1997	0.3215	0.2006	0.3065	0.2011	0.3637	0.2010	1.3408
	72	0.5349	0.5400	0.5372	0.5411	0.5310	0.5895	0.5374	0.5914	0.5411	0.5843	0.5354	1.0216
	432	0.7072	0.6825	0.7075	0.6840	0.6831	0.7015	0.7055	0.7028	0.7169	0.8031	0.6990	0.8427
GREECE	1	0.1150	0.1245	0.1139	0.1418	0.1113	0.3653	0.1139	0.3909	0.1140	0.4854	0.1134	0.3714
	72	0.4817	0.5208	0.5583	0.5208	0.5520	0.6307	0.5595	0.6323	0.5655	0.6971	0.5581	0.6550
	432	0.6858	0.7085	0.6783	0.7107	0.7637	0.7135	0.6819	0.7124	0.7202	0.7624	0.7393	0.7725
CAPITALS	1	0.1017	0.1132	0.1052	0.1651	0.1007	0.4704	0.1037	0.5772	0.1002	0.5781	0.1008	0.5522
	72	0.6716	0.7011	0.7072	0.7011	0.6978	0.7327	0.7195	0.7344	0.7053	0.7638	0.7035	0.8193
	432	0.7533	0.7773	0.8140	0.7783	0.7758	0.7927	0.8058	0.7937	0.7801	0.8266	0.7754	0.8261

The findings indicate that in most environments FFT and PFT performed better than the parameter efficient adaptation techniques. PFT pretrained won the most number of wins in centralized column, but FFT was still competitive in short horizons federated forecasting. This implies that heavyweight adaptation of the pretrained model is more advantageous than lightweight tuning in wind speed forecasting. Thus, FFT and PFT were chosen as the most effective fine-tuning techniques to be used in subsequent experiments.

D. Evaluation of LLM Backbones

The second section of the assessment compared the various LLM backbones with the aim of establishing the most appropriate architecture to be utilized in the proposed privacy-preserving forecasting pipeline. Despite a well-developed fine-tuning plan, the quality prediction can still be largely reliant on the fine-tuning model. The available backbones assessed were GPT, LLaMA, BERT, BART, Qwen, Gemma and OPT.

To ensure the main body is concise, Table II below shows the performance of the federated learning using each of the backbones of LLMs with prediction length 72 using FFT method on KZMET dataset.

TABLE II: Model Performance: MAE and RMSE

Model	MAE	RMSE
GPT2	0.6507	0.8888
GEMMA	0.7904	1.0366
OPT	0.7957	1.0360
LLAMA	0.8019	1.0305
QWEN	0.8175	1.0548
BART	0.7278	0.9674
BERT	0.7118	0.9641

The best overall performance of the compared backbones was found in GPT2 with the lowest average MAE and RMSE at horizon 72 at KZMET. BART and BERT were also competitive, although neither of them consistently outperformed GPT2 on given datasets and horizons. These findings led to the choice of GPT2 as the backbone of the rest of the federated comparisons.

E. Evaluation of Federated Learning Algorithms

The third section of the assessment tested the effectiveness of the federated aggregation technique that is best in the heterogeneous distributions of clients that exist in wind forecasting. This comparison is particularly crucial due to the fact that the primary goal of the project is to maintain privacy by means of decentralized learning, and the effectiveness of this system is not only related to the backbone of the model but also to the strategy that will be employed to aggregate the outcomes across the clients.

The algorithms compared included FedAvg, FedProx, FedPer, FedBN and StatAvg. A small summary of the findings on KZMET at horizon 72 is summarized in Table IV and Figure 4

shows a graphical representation.

TABLE IV: Summary of federated learning algorithm performance (MAE)

FL Algorithm	Pred Len 1	Pred Len 72	Pred Len 432
FedAvg	0.4420	0.7342	0.7181
FedProx	0.1617	0.6507	0.6913
FedPer	0.2021	0.7340	0.7838
StatAvg	0.1435	0.7346	0.7896
FedBN	0.2024	0.7339	0.6871

Among the strategies federated, the choice of algorithms that excel in all horizons is not unique. At prediction length 1, StatAvg has the lowest MAE of 0.1435, then, FedProx, with 0.1617. FedProx and FedBN are most accurate at the medium-

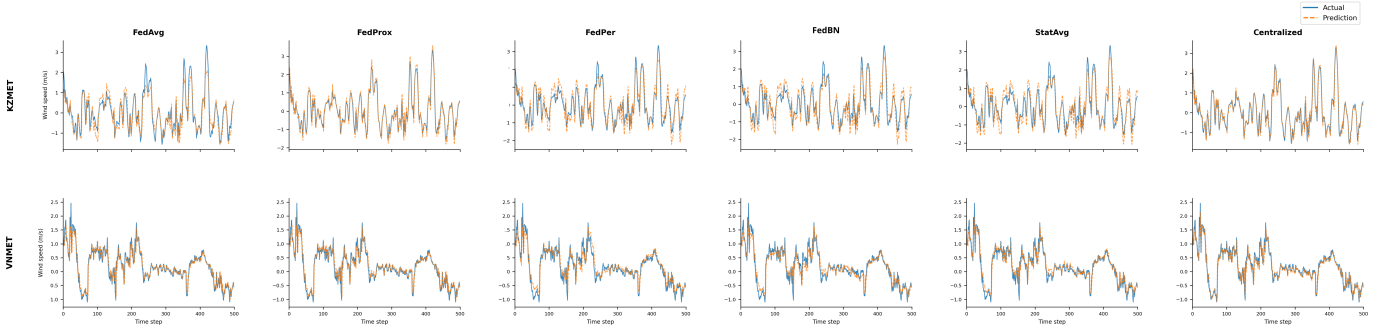


Fig. 4: Predicted vs. actual wind speed at prediction horizon 72 for each federated aggregation strategy and the centralized baseline, evaluated on KZMET (left) and VNMET (right). Results shown for client 0 at the best validation round

TABLE III: Test MAE and RMSE across datasets and number of federated clients for prediction length of 72.

Dataset	Model	2 clients		4 clients		5 clients	
		MAE	RMSE	MAE	RMSE	MAE	RMSE
KZMET	GPT	0.6464	0.8843	0.6493	0.8870	0.6492	0.8888
	LLaMA	0.8029	1.0311	0.8025	1.0308	0.8019	1.0305
	BERT	0.7073	0.9591	0.7105	0.9591	0.7118	0.9641
	BART	0.7161	0.9624	0.7097	0.9630	0.7278	0.9674
	Qwen	0.7943	1.0439	0.8172	1.0544	0.8175	1.0548
	Gemma	0.7748	1.0299	0.7884	1.0361	0.7904	1.0366
	OPT	0.7957	1.0359	0.7827	1.0266	0.7904	1.0360
VNMET	GPT	0.5200	0.7693	0.5838	0.7837	0.5310	0.7929
	LLaMA	0.6314	0.7827	0.6138	0.8141	0.5971	0.8341
	BERT	0.5873	0.7732	0.5853	0.7765	0.5885	0.7793
	BART	0.5788	0.7618	0.5864	0.7780	0.5869	0.7791
	Qwen	0.6195	0.8497	0.6248	0.8159	0.6465	0.8158
	Gemma	0.6136	0.8085	0.6101	0.7955	0.6299	0.8322
	OPT	0.6583	0.8402	0.6689	0.8639	0.6808	0.8736
GREECE	GPT	0.5404	0.6997	0.6056	0.8237	0.5520	0.8309
	LLaMA	0.6285	0.8447	0.6192	0.8447	0.6188	0.8611
	BERT	0.6355	0.8961	0.6490	0.7393	0.6568	0.9032
	BART	0.6294	0.8351	0.6583	0.9001	0.6614	0.9142
	Qwen	0.6167	0.8363	0.6061	0.8266	0.6240	0.8478
	Gemma	0.6068	0.8321	0.6475	0.8804	0.6537	0.8845
	OPT	0.6638	0.8834	0.6727	0.9304	0.6754	0.9340
CAPITALS	GPT	0.6716	0.9549	0.7426	0.9677	0.6978	0.9786
	LLaMA	0.8034	1.0379	0.7830	1.0167	0.7569	1.0137
	BERT	0.7337	0.9833	0.7624	1.0013	0.7733	1.0089
	BART	0.7610	0.9638	0.7386	0.9965	0.7714	1.0092
	Qwen	0.7927	1.0252	0.8117	1.0471	0.8329	1.0683
	Gemma	0.8156	1.0758	0.7995	1.0541	0.8310	1.0359
	OPT	0.8028	1.0407	0.7898	1.0176	0.8247	1.0580

term prediction length of 72 with an MAE of 0.6507 and 0.7339 respectively. When predicting on a long-term horizon at a prediction length of 432, FedBN has the lowest MAE of 0.6871, closely followed by FedProx at 0.6913. FedProx turns out to be the most consistently competitive strategy, as it is first or second in all three horizons, whereas FedAvg is a steady baseline but fails to score the highest in any of the three horizons.

F. Centralized Versus Federated Performance

One of the major components of the assessment was the direct comparison of centralized and federated learning. Such a comparison is essential as centralized training can have a rough upper-bound reference, but federated learning does not violate privacy since raw information is not shared across clients. Thus, the difference between the performance of the two settings reflects the price of privacy protection.

The difference between the centralized and federated structures of the three prediction horizons and various datasets is summarized in Table V. The default model is the GPT2, FFT method.

TABLE V: Comparison of Centralized and Federated Best GPT-based model’s MAE across datasets and horizons.

Dataset	Horizon	Central MAE	Fed MAE	% Diff
KZMET	1	0.1480	0.1617	9.24%
	72	0.6222	0.6507	4.58%
	432	0.6688	0.6913	3.37%
VNMET	1	0.2010	0.2039	1.44%
	72	0.5349	0.5400	0.95%
	432	0.7072	0.6825	3.49%
GREECE	1	0.1150	0.1245	8.26%
	72	0.4817	0.5208	8.11%
	432	0.6858	0.7085	3.31%
CAPITALS	1	0.1017	0.1132	11.31%
	72	0.6716	0.7011	4.39%
	432	0.7533	0.7773	3.18%

Generally, federated GPT-based model is competitive with centralized baseline in datasets and horizons. The findings indicate that the relative performance is not uniform across all datasets and forecasting horizons, but rather depends on the dataset and the forecasting horizon. Federated learning is not majorly behind centralized performance considering the pros of preserving privacy, showing that the training can be decentralized without significantly losing accuracy.

G. Impact of Number of Clients

The fourth section of the assessment explored the concept of scalability by evaluating the variation in the forecasting performance as the number of clients to participate in the process increases. The significance of this analysis is that feasible federated learning systems can consist of numerous distributed actors, and a viable approach ought to be stable as the size of collaboration increases. In this experiment, the overall data used in training and testing were kept constant and only the separation between clients was varied. Consequently, it implies

that with fewer clients, a client is provided with a bigger local dataset and with many clients, they are provided with fewer data and more data heterogeneity amongst participants.

As can be seen in Tables III and VI the effect of the increasing number of clients is not monotonic, yet there is an evident overall trend. With 2 or 4 clients, the best results are achieved and with 5 clients, MAE and RMSE are more likely to increase. This trend is particularly clear in the case of GPT as the performance tends to decrease with the number of clients in all four datasets. We can probably explain this by the fact that the larger the number of clients, the smaller and possibly more specialized a subset of the data is trained locally by each participant, leading to noisier local updates and more challenging aggregation.

Meanwhile, the outcomes also indicate that the moderate growth of the number of clients may be helpful in some cases. As illustrated in Table VI in KZMET, there are federated aggregation schemes like StatAvg and FedBN that perform better with 4 clients compared to 2 clients, which means that a moderate level of distributional diversity can lead to better generalization in some scenarios. But this advantage is not sustained as the client base grows further to 5 where performance drops uniformly with all federated methods being tested. These results indicate, in general, that given a constant data total budget, trade-offs exist between more cross-client diversity and enough local data per client, with small or middle client numbers offering the best balance.

H. Impact of Participation Rate

The fifth section of the assessment measured the impact of participation rate, i.e., the percentage of clients who provide updates in every round of communication. The analysis is significant since in a real-world federated system not all clients would be available at any round.

The impact of various participation rates on forecasting performance is summarized in Table VII. An increased number of participants will enhance the forecasting performance and demonstrate that the global model will be less affected by the updates of a smaller portion of clients. However the system was still usable at lower participation rates indicating that the proposed solution is strong enough to be used in realistic federated environments, in which complete participation is not always possible.

I. Overall Evaluation Summary

Altogether, the analysis shows that the offered system can help to resolve the issue that was stated at the very start of the project to a considerable extent. These findings indicate that the models of LLM-based forecasting can be successfully adapted to the federated learning environment and that their performance can be similar to that of centralized learning, without harming the privacy. The strongest adaptation strategies (full fine-tuning and partial fine-tuning), the most successful backbone (GPT2), and the best overall federated performance (across datasets and prediction lengths) were all achieved with FedProx.

TABLE VI: Test MAE and RMSE on KZMET dataset across federated learning algorithms and number of clients for prediction length of 72.

Model	FL Method	2 clients		4 clients		5 clients	
		MAE	RMSE	MAE	RMSE	MAE	RMSE
GPT2	FedAvg	0.6938	0.9299	0.6938	0.9313	0.7342	0.9317
	FedProx	0.6464	0.8843	0.6493	0.8870	0.6507	0.8888
	FedPer	0.6921	0.9295	0.7168	0.9333	0.7340	0.9484
	FedBN	0.6925	0.9297	0.6331	0.9449	0.7339	0.9497
	StatAvg	0.6933	0.9306	0.6241	0.9406	0.7346	0.9494

TABLE VII: Test MAE and RMSE on KZMET across training fractions

Dataset	Model	0.1		0.25		0.5		1.0	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
KZMET	BART	0.7136	0.9721	0.7118	0.9649	0.7104	0.9642	0.7093	0.9625
	BERT	0.7118	0.9703	0.7112	0.9655	0.7095	0.9623	0.7070	0.9591
	Gemma	0.7998	1.0596	0.7981	1.0489	0.7952	1.0396	0.7904	1.0366
	GPT2	0.6962	0.9353	0.6961	0.9328	0.6509	0.8885	0.6507	0.8888
	LLaMA	0.8656	1.0494	0.8453	1.0479	0.8329	1.0430	0.8019	1.0305
	OPT	0.8005	1.0403	0.7992	1.0379	0.7998	1.0377	0.7957	1.0360
	Qwen	0.8226	1.0563	0.8223	1.0565	0.8203	1.0560	0.8175	1.0548

Although forecasting error increased for longer prediction horizons and under more difficult federated conditions, the performance degradation remained moderate. This suggests that the developed system is not only technically feasible, but also practically promising as a privacy-preserving solution for collaborative wind forecasting.

VIII. CONCLUSION AND POSSIBLE FUTURE WORK

This project researched the combination of pretrained large language model-based time-series forecasting with federated learning to preserve clients' privacy. Experiments were conducted across four geographically diverse datasets (Kazakhstan, Vietnam, and Greece and cross-country Capitals), seven LLM-based architectures, four fine-tuning strategies, and five federated aggregation algorithms, evaluated at three forecasting horizons.

The experiments yielded several key findings. First of all, it was found that full and partial fine-tuning outperform LoRA or AdaLoRA.

Secondly, GPT2 achieved strongest performance regardless of the task type compared to other LLM backbones. This suggests that GPT4TS's time-series specific simple architecture is more suitable for univariate wind speed prediction in contrast to bigger LLaMA and Qwen models. These other models' architectures may be too complex for simple univariate wind speed prediction.

Thirdly, federated learning achieved similar performance to centralized models for longer horizon forecasting, even though the difference is significant for short-term prediction (1-step). This may be the result of model's performance deteriorating past a certain prediction length, after which model's results plateau and do not improve regardless of the strategy. Other hypothesis is that short-horizon prediction relies on client-specific, fine-grained dynamics that are averaged by a global model, and longer-horizon prediction is driven by coarse

seasonal structure which is preserved by federated averaging. Its verification is left to future work.

Fourthly, there was no clearly the best aggregation strategy. However, FedProx achieved the best result in the medium prediction length and was the second best for short-term and long-term prediction.

There were several limitations, including the univariate nature of wind speed data, which makes application findings to multivariate settings or direct wind power prediction unverified. Additionally, computational constraints limited the scale of hyperparameter tuning. Simulation of the federated environment using Flower framework on a single device simplifies the data transferring, and thus real-world challenges such as communication latency, device heterogeneity, or asynchronous updates were not tested thoroughly.

Future work should address these limitations by incorporating multivariate inputs (direction, temperature, pressure, humidity), testing on actual wind power output data, and deploying the system in a genuinely distributed environment. Finally, investigating why federated training achieves performance comparable to centralized training could yield better insights and prove or disprove our initial hypothesis.

REFERENCES

- [1] S. Moradian, S. Gharbia, G. Iglesias, and A. I. Olbert, "A copula post-processing method for wind power projections under climate change," *Energy Conversion and Management: X*, vol. 23, p. 100660, 2024. <https://doi.org/10.1016/j.ecmx.2024.100660>
- [2] G. Giebel et al., "Forecasting for the weather driven energy system - a new task under IEA wind," *IEA Wind*, 2022. <https://doi.org/10.1049/icp.2022.2830>
- [3] H. Fan, X. Fan, W. Wei, T. Hao, K. Chen, G. Wang, and W. Xu, "Privacy preserving ultra-short-term prediction in clustered wind farms with encrypted data sharing: A secure multi-party computation approach," *Expert Systems with Applications*, vol. 278, p. 127218, 2025. <https://doi.org/10.1016/j.eswa.2025.127218>

- [4] Y. Wang, W. Zhang, Q. Guo, and Y. Wu, "A Privacy-preserving Wind Speed Prediction Method Based on Federated Deep Learning," *ICPET*, pp. 638–643, 2022. <https://doi.org/10.1109/ICPET55165.2022.9918344>
- [5] M. Jia and C. Shen, "Privacy-Preserving Distributed Joint Probability Modeling for Spatial-Correlated Wind Farms," *arXiv: Learning*, 2018. <https://dblp.uni-trier.de/db/journals/corr/corr1812.html#abs-1812-09247>
- [6] R. Wang, Y. Fan, H. Yun, R. Rayhana, L. Bai, and Z. Liu, "Multiserver Federated Learning for Building Energy Prediction With Wind Speed," *IEEE Transactions on Instrumentation and Measurement*, vol. 74, pp. 1–14, 2025. <https://doi.org/10.1109/TIM.2024.3522628>
- [7] T. Zhou, P. Niu, L. Sun, and R. Jin, "One Fits All: Power General Time Series Analysis by Pretrained LM," in *Advances in Neural Information Processing Systems*, vol. 36, 2023, pp. 43322–43355. [Online]. <https://doi.org/10.48550/arXiv.2302.11939>
- [8] P. W. Stackhouse et al., "NASA Prediction of Worldwide Energy Resources (POWER) project," NASA Langley Research Center, Hampton, VA, Hourly v10.2.1, 2018. <https://power.larc.nasa.gov/>
- [9] Power Engineering Consulting Joint Stock Company 3 (PECC3), "Wind measurement for planning and development of wind power projects," Commissioned by Deutsche Gesellschaft für Internationale Zusammenarbeit (GIZ), 2018. <https://energydata.info/dataset/vietnam-wind-measurements-giz>
- [10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, Minneapolis, MN, 2019, pp. 4171–4186. <https://arxiv.org/abs/1810.04805>
- [11] H. Touvron et al., "LLaMA: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023. <https://arxiv.org/abs/2302.13971>
- [12] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. ACL*, 2020, pp. 7871–7880. <https://arxiv.org/abs/1910.13461>
- [13] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, Fort Lauderdale, FL, 2017, pp. 1273–1282. <https://arxiv.org/abs/1602.05629>
- [14] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," *arXiv preprint arXiv:1912.00818*, 2019. <https://arxiv.org/abs/1912.00818>
- [15] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "FedBN: Federated learning on non-IID features via local batch normalization," in *Proc. ICLR*, 2021. <https://arxiv.org/abs/2102.07623>
- [16] P. S. Bouzinis et al., "StatAvg: Mitigating data heterogeneity in federated learning for intrusion detection systems," *arXiv preprint arXiv:2405.13062*, 2024. <https://arxiv.org/abs/2405.13062>
- [17] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. ICLR*, 2020. <https://arxiv.org/abs/1812.06127>
- [18] United Nations Economic Commission for Europe (UNECE), "Energy connectivity: Kazakhstan policy brief," UNECE, Geneva, Jan. 2025. <https://unece.org/sites/default/files/2025-01/Energy%20Connectivity-Kazakhstan%20Policy%20Brief.pdf>
- [19] Kazakhstan Electricity and Power Market Operator (KOREM), "Annual report 2024," KOREM, Astana, 2024. https://www.korem.kz/eng/o_kompanii/godovye_otchety/
- [20] Energypedia, "Wind energy country analysis: Vietnam," Energypedia, 2023. https://energypedia.info/wiki/Wind_Energy_Country_Analysis_Vietnam
- [21] A. Ntalfalias, A. Weissenfeld, S. Visvardi, S. Tsakanikas, and P. Papadopoulos, "Descriptor: Six-Month Monitoring Dataset From a Ten-Turbine Onshore Wind Farm in Greece (SMD10TOWFGR)," *IEEE Data Descriptions*, vol. 2, pp. 187–192, 2025, <https://doi.org/10.5281/zenodo.14546479>
- [22] J. Bai et al., "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023. <https://arxiv.org/abs/2309.16609>
- [23] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, "OPT: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068*, 2022. <https://arxiv.org/abs/2205.01068>
- [24] Gemma Team, Google DeepMind, "Gemma: Open Models Based on Gemini Research and Technology," *arXiv preprint arXiv:2403.08295*, 2024. <https://arxiv.org/abs/2403.08295>