Federated Learning for Distributed Weather Forecasting: A Practical Approach on Real Multidimensional Georeferenced Data

Attilio Di Vicino[1,†], Giuseppe Fiorillo[1,*,†], Luigi Galluccio[1,†] and Raffaele Montella[1]

[1]*Department of Science and Technology, University of Naples "Parthenope", Naples*

Abstract

The rapid advancement of technology—particularly in machine learning and data availability has led to an increasing demand for reducing communication overhead and operational costs. Edge computing is one of the fastest—growing fields, enabling cost reduction and improving communication. Simultaneously, the advancement of machine learning and the availability of data are both accelerating. Federated learning (FL) addresses not only privacy concerns but also issues related to cost and communication efficiency. This paper presents an application of a federated learning framework to improve weather forecast accuracy through the collaborative analysis of distributed data, while ensuring data confidentiality and computational efficiency. To replicate the federated environment, we developed an architecture that combines real-time data collection using the Signal K server, containerization using Docker, and a Hadoop cluster on Microsoft Azure. We evaluated the performance of a Transformer and a Crossformer, demonstrating the effectiveness of both models in this context, with the Crossformer showing superior performance in managing spatiotemporal dependencies for forecasting. Our experimental results indicate an important improvement in reducing the error with respect to previous methods, achieving a Mean Absolute Error (MAE) of 0.144 for the Crossformer and 0.232 for the Transformer, highlighting the potential of FL and advanced deep learning architectures in managing sensitive data in distributed scenarios, in line with previous research trends. This study proposes a robust and scalable approach, opening new perspectives for future applications in cooperative and secure learning.

Keywords

Federated learning, edge computing, weather forecasting, time series

1. Introduction

The exponential growth in the cost for training neural networks has become an increasingly pressing issue, both in terms of financial cost and computational resources, as highlighted in studies [1]. The trend shows a dramatic increase not only in hardware requirements but also in energy consumption and training time. This underlines the necessity for innovative methods to address such exponential growth.

One of the most promising approaches to address these challenges is edge computing, which aims to offload computation from centralized data centers to distributed local nodes. However, with the rising amount of data being generated—especially in sensitive domains—the need for secure, privacy-preserving data management becomes equally critical. This is where federated learning (FL) is relevant [2].

FL is a recently developed distributed deep learning paradigm in which clients independently train their local neural network models with private data and then jointly aggregate a global model on the

BigHPC2025: Special Track on Big Data and High-Performance Computing, co-located with the 4th Italian Conference on Big Data and Data Science, ITADATA2025, September 9 – 11, 2025, Torino, Italy.

*Corresponding author.

†These authors contributed equally.

attilio.divicino001@studenti.uniparthenope.it (A. D. Vicino); giuseppe.fiorillo001@studenti.uniparthenope.it (G. Fiorillo); luigi.galluccio001@studenti.uniparthenope.it (L. Galluccio); raffaele.montella@uniparthenope.it (R. Montella)

https://github.com/Attilio-Di-Vicino (A. D. Vicino); https://github.com/GiuseppeFiorillo (G. Fiorillo); https://github.com/LuigiGalluccio (L. Galluccio)

0009-0009-8159-9118 (A. D. Vicino); 0009-0005-4769-0920 (G. Fiorillo); 0009-0002-8441-3381 (L. Galluccio); 0000-0002-4767-2045 (R. Montella)

central server [3], offering a promising approach for training machine learning models on local devices while avoiding raw data sharing and maintaining control over sensitive information.

The aim of this project is to investigate the practical use of federated learning in collecting and analyzing multidimensional georeferenced data from dispersed, remote stations. Using cooperative models will help to improve weather forecasting accuracy by lowering the computing cost to train the model and maintaining data privacy while also ensuring that, as observed in [4], peripheral nodes' limited computing power and confidentiality restrictions are respected. However, there are several weather phenomena that are very complicated to predict and that is why it still turns out to be a complicated task [5].

To address this challenge, an architecture that leverages modern technologies such as Signal K server for real-time data collection, Docker for containerization of services, and a Hadoop cluster on Microsoft Azure for federated paradigm simulation has been designed. The main contribution lies in the definition of a complete pipeline, ranging from data collection and pre-processing to the collaborative training phase with advanced federated learning techniques. In particular, the FedProx algorithm was used to put together the weights of different models.

By integrating Transformer and Crossformer models for space-time analysis [6, 7], it is shown that FL techniques can be effectively applied in real-world scenarios, overcoming the limitations of data centralization and paving the way for new, secure, and cooperative learning frameworks. The best performance was achieved with the Crossformer model, which obtained a Mean Square Error (MSE) of 0.232 and a Mean Absolute Error (MAE) of 0.144, significantly improving over the previous approach of De Vita et al. [8], where the MSE was 1.324 and the MAE was 0.827.

The rest of the paper is organized as follows: in Section 2 we discuss the literature; Section 3 illustrates the main architecture, including data collection and data pre-processing; Section 4 discusses the practical implementation of the models; Section 5 details the experimental results of our solution, comparing them to previous works; Section 6 provides a critical interpretation of the results and highlights their implications; finally, Section 7 wraps up the paper's views and outlines possible directions for future research.

2. Related Work

FL was introduced around 2016 by Google as an innovative approach to training machine learning models. Traditionally, developing high-quality models required access to large amounts of data. This frequently requires working directly with raw and sensitive data, raising privacy and security concerns [9, 10].

FL addresses these challenges by allowing model training across multiple decentralized devices, reducing the need to share or centralize raw data [11, 12]. The core concept is to perform local training on each node, and then combine the weights into a single global model through a coordinating node [13]. Only encrypted model weights are shared, not the raw data itself, which ensures privacy throughout the process [14, 15].

Several recent studies address data privacy in FL applied to time-series forecasting with georeferenced sensors. For instance, privacy-preserving collaborative forecasting techniques have been analyzed in energy and time-series domains, demonstrating how data transformation, secure multi-party computation, and differential privacy can be used to balance forecast accuracy and confidentiality. From a general perspective, data privacy becomes especially important in edge computing scenarios [16].

Reviews focused on IoT sensor aggregation highlight the challenges in resource-constrained settings, such as weather stations. Federated learning approaches combined with secure aggregation have proven effective in residential load forecasting scenarios. These studies support the feasibility of extending federated approaches with differential privacy or secure aggregation approximations in distributed weather forecasting systems [17].

Based on the previous considerations, privacy is in fact a critical issue for both individual users and large enterprises, which is why FL has gained significant attention and experienced a boom in recent years. However, many FL systems still rely on a centralized approach, which can lead to several problems, such as data falsification and lack of transparency [11]. That is why there are several studies on FL, ranging from IoT-based systems to concerns about data privacy and security [15].

While federated learning reduces privacy risks by keeping raw data at the edge, distributed weather forecasting introduces several domain-specific security vulnerabilities. A recent study demonstrates that corrupting a small subset of input observations, even in non-sensitive sensor networks, can significantly skew forecast accuracy. This exemplifies the real risk of poisoning or manipulation at the data collection layer [18].

The study in [11] summarizes how FL and blockchain can be combined to overcome these obstacles. Furthermore, Chen et al. [6] presented prompt federated learning (PFL) to address the challenges of cooperative weather forecasting across different meteorological datasets. This approach exploits a spatiotemporal Transformer-based foundation model, with a novel prompt learning mechanism designed to meet the communication and computational constraints of low-resource sensors. Chen et al. [19] also introduced Federated Prompt Learning for Weather Foundation Models on Devices (FedPoD) to address challenges such as data heterogeneity and communication overhead in on-device weather forecasting, employing adaptive prompt tuning to obtain highly personalized model training while maintaining communication efficiency [19, 20].

In the realm of renewable energy, Li et al. [21] proposed an approach called federated deep reinforcement learning (FedDRL) for ultra-short-term wind power forecasting. FedDRL integrates the deep deterministic policy gradient (DDPG) algorithm within a federated learning framework not only to improve prediction accuracy but also to enable decentralized model training without compromising data privacy. Climate change impacts, and the dependence on non-renewable energy sources for electricity generation, led to the proposal of numerous IoT-based smart weather station systems [22].

De Vita et al. [8] proposed an effective method to predict spatiotemporal weather data using advanced deep learning models. In particular, the method is based on Transformer and Crossformer architectures (the latter being more suitable for spatiotemporal series). Data were collected from weather stations at the University of Naples "Parthenope", specifically two stations, and temperature, humidity, pressure, wind speed, and wind direction were used as input characteristics. The target of the prediction was the temperature. This work highlights not only the effectiveness of FL but also the potential of deep learning models such as Transformers and Crossformers in weather prediction. It was also demonstrated that, in this specific case, the Crossformer outperformed the classical Transformer when dealing with spatiotemporal data [7]. Currently, there is a large body of research on weather prediction, each focusing on different models such as artificial neural networks and deep neural networks.

3. Materials and Methods

To address the challenges of distributed weather data collection and analysis, we have developed an architecture and methodology that leverages state-of-the-art technologies and distributed machine learning paradigms. The objective is twofold: to ensure the accuracy of forecasts while preserving data privacy and optimizing the use of computational resources. This section describes in detail the key components of our infrastructure, the data acquisition and pre-processing phases, and the adopted federated learning model.

The collection of real-time meteorological data from each station is managed via Signal K Server, an open-source protocol originally designed to aggregate data from marine environments [23, 24].

Federated learning is coordinated by Flower, a framework that manages the aggregation of model weights and provides several algorithms, including Federated Averaging (FedAvg) and Federated Proximal (FedProx). Each of these has specific characteristics, but we chose FedProx to reduce communication overhead between the client and the server, as it employs a quantization process to compress model updates [25, 26]. After local training on each station, the encrypted weights of the models are shared with the coordinating node—never the raw data—thus ensuring privacy. The coordinating node aggregates these weights to create a single global model. Subsequently, updated versions of the global model are sent back to the stations for the next training round, iteratively improving the accuracy of the overall

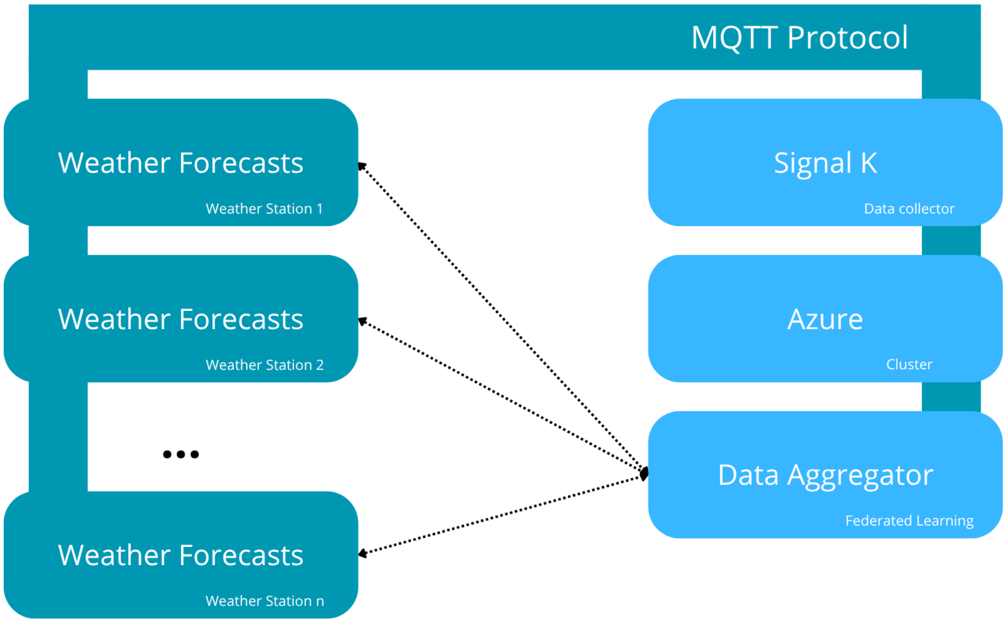


Figure 1: Weather stations $[1, ..., n]$ locally generate forecasts based on sensor data, that are passed to our system (Azure Hadoop distributed system) through MQTT protocol and thanks to Signal K Server. In the system, each client node perform a single training using data from the corresponding weather station.

model.

3.1. Data Collection

To collect data from distributed weather stations, we used Signal K, an open-source software protocol originally developed for aggregating sensor data from maritime environments. Signal K was used in this context to gather real-time weather data from sensor-equipped weather stations provided by the University of Naples "Parthenope". A plugin has been set up in the Signal K system to allow distributed weather stations and a central broker to communicate using the MQTT protocol, allowing time series data to be published and transmitted efficiently over the network. Another plugin has been employed in order to save data locally for model training.

This architecture, illustrated in Figure 1, allowed the central server to receive live environmental measurements from multiple remote nodes with no raw hardware access. The Signal K services were containerized using Docker, offering a reproducible and scalable deployment environment for data collection across the cluster.

3.2. Data Pre-processing

The data input is highly sensitive to data quality. To address this problem, which may lead to imprecise forecasts, we need to do some data pre-processing operations [27].

We created a pre-processing pipeline that includes the following processes in order to standardize and modify this data for our purposes:

1. *Wind Direction Encoding*: In order to maintain angular relationships and render it suitable for our machine learning assignment, the wind direction was encoded into its sine and cosine, as it is cyclic.
2. *Data Smoothing*: To reduce noise, each numeric feature was smoothed using a moving average with a window size of 3.

Table 1
Comparison of selected technologies with relevant alternatives and rationale for their adoption.

Component	Technology Used	Alternatives	Rationale for Choice
Data Collection	Signal K + MQTT	WeatherFlow, REST API polling	Signal K was chosen for its modular, real-time, and MQTT-native design suitable for edge deployments.
Containerization	Docker	Podman, Kubernetes	Docker provides lightweight, widely supported containerization, ideal for IoT clusters and low-resource devices.
Orchestration	Flower (with FedProx)	FedML, TensorFlow Federated	Flower allows flexible FL configuration and seamless PyTorch integration, supporting realistic edge use cases.
Data Aggregation	Hadoop (Azure HDInsight)	Spark, Kafka Streams	Hadoop offers mature distributed storage and processing capabilities with proven Azure cloud integration.

3. *Data Normalization*: To increase model stability and make the training process easier, all numerical features-aside from time and direction encoding—were normalized in order to have zero mean and unit variance (i.e., the normal distribution).

In total, there were 20,199 data, collected as follows: 4,196 from Castelvolturno; 6,829 from Città della scienza; 4,919 from Marina di Stabia; 4,255 from Via Acton. The sampling interval was $\Delta T \approx 1$ hour [28].

3.3. Technology Justification and Comparison

To motivate the architectural decisions taken during the implementation of our federated learning testbed, we provide a comparative overview of the adopted technologies and their alternatives. Table 1 summarizes the main components and the rationale behind each selection.

4. Model Architecture

In order to process and train the models on distributed data, we used a Hadoop 3.3.4 cluster using Microsoft Azure, as illustrated in Figure 1. The head nodes were configured using the E2 v3 instance type (2 cores, 16 GB of RAM), with two head nodes deployed for redundancy. For fault tolerance and coordination, we also used three zookeeper nodes based on the A2 v2 instance type (2 cores, 4 GB RAM). This central infrastructure provides the coordination and management capabilities necessary for our distributed environment.

The worker nodes were configured using the E2 v3 instance type (2 cores, 16 GB of RAM), with two worker nodes deployed in the cluster. This infrastructure offers a trade-off between computational efficiency and scalability for our distributed environment. It has been employed to simulate a distributed federated learning environment, where each worker node acted as two different clients participating in model training, mimicking automated weather stations collecting and processing meteorological data in a federated manner.

For weather forecasting, we integrated the Transformer and Crossformer models, advanced deep learning architectures suitable for time series analysis with complex dependencies. The Transformer was used for its ability to capture long-range relationships in data sequences [29]. The Crossformer, specifically designed for multivariate and multimodal time series, has been used to handle spatiotemporal interactions between different stations [30], but both are based on the Transformer architecture [31].

Table 2
Comparison between two machine learning models. The metrics used to compute the errors are the Mean Square Error (MSE) and Mean Absolute Error (MAE). The best results are in bold.

Models	MSE	MAE
Transformer	0.265	0.194
Crossformer	**0.232**	**0.144**

Table 3
Comparison between our approach and De Vita et al. [8] results on machine learning models. The metrics used to compute the errors are the Mean Square Error (MSE) and Mean Absolute Error (MAE). The best results are in bold.

	De Vita et al. [8] results		**Our results**	
Models	MSE	MAE	MSE	MAE
Transformer	2.851	1.411	0.243	0.313
Crossformer	**1.324**	**0.827**	**0.232**	**0.144**

In addition, both the Transformer and Crossformer architectures had a model dimension of 64, four attention heads, two layers, and a dropout rate of 0.1. The input dimension was 4, including the features `station_id`, `temperature`, `humidity`, and `wind_speed`, as well as the encoded wind components `wind_sin` and `wind_cos`. The output dimension was 1, focusing on `temperature`. The model operated with an `input_window` of 20 hours and an `output_window` of 1 hour. Federated training was conducted over 20 rounds, with 25 local training periods per round.

5. Experimental Results

5.1. Models Evaluation

Model evaluation was conducted to quantify the effectiveness of Transformer and Crossformer architectures in the context of distributed weather forecasting, leveraging the FL approach. The primary objective was to determine the performance of each model in terms of predictive accuracy, taking into account the specificities of the meteorological data collected and the distributed configuration of the training.

The experimental environment has been configured to simulate the real operating conditions of distributed weather stations.

Model performance was evaluated using Mean Square Error (MSE) and Mean Absolute Error (MAE), two of the best metrics for assessing time series predictions [32].

5.2. Federated Learning Results

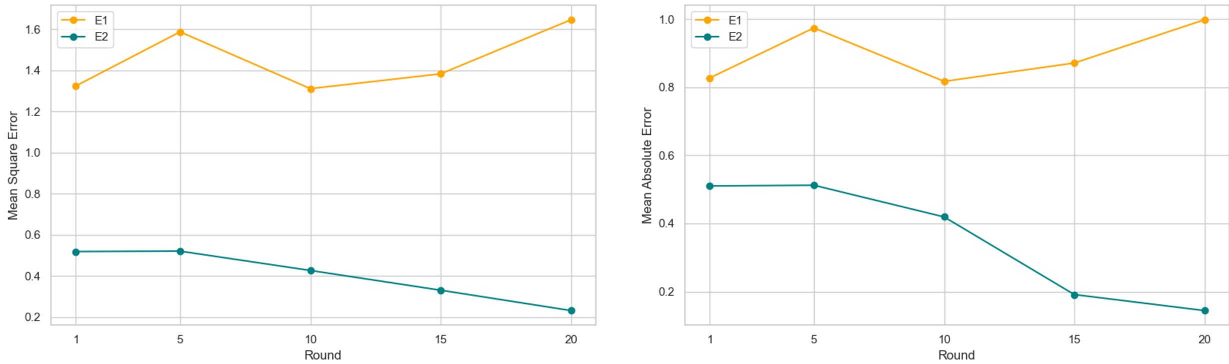
Table 2 compares Transformers and Crossformers models using MSE and MAE metrics. The evaluated model is the aggregated one obtained at round 20 (the last one) using the FedProx algorithm. Table 3 employs the same approach as the first table; however, it compares our results with the work of De Vita et al. [8].

Table 4 shows the performance of the aggregated model (here it refers to the results obtained by the best model between the two, in this case the Crossformer) over the course of the 20 rounds of federated training, showing a progressive improvement in error metrics as the overall model becomes more refined. These preliminary results support the effectiveness of Federated Learning and the architectures chosen for the distributed weather forecasting task.

Figure 2 further compares our approach with that of De Vita et al. [8] for the Crossformer model in the context of federated learning, analyzing MSE and MAE metrics over 20 rounds of training.

Table 4
Results obtained for federated learning training of 20 rounds, 25 epochs per round. The metrics used to compute the errors are the Mean Square Error (MSE) and Mean Absolute Error (MAE). The best results are in bold.

Round	MSE	MAE
1	0.519	0.510
5	0.521	0.512
10	0.427	0.419
15	0.331	0.191
20	**0.232**	**0.144**



(a) MSE results

(b) MAE results

Figure 2: Crossformer federated learning comparison. E1 refers to De Vita et al. [8] results; E2 refers to the results of our work. The metrics used to compute the errors are the MSE (a) and MAE (b).

The data clearly show the superiority of our method. Although the results of De Vita et al. [8] show some variability with a minimum MSE of 1,311 and an MAE of 0,817 in round 10, our approach has significantly lower error values from the first round (MSE 0,510, MAE 0,519). At the end of the 20 rounds, our model achieves optimum performance with an MSE of 0.232 and an MAE of 0.144, demonstrating not only a more effective convergence but also overall greater stability and accuracy compared to the above work.

6. Discussion

As can be seen in Table 2, the Crossformer has demonstrated superior performance compared to the Transformer in terms of MSE and MAE (using the fictitious values provided), suggesting a greater ability to manage complex space-dependent weather patterns in meteorological data. This outcome aligns with the Crossformer design, which is tailored for the cross-analysis of multivariate time series [30].

The results obtained, summarized in Table 3, demonstrate an improvement in the performance of the Transformer and Crossformer models compared to the values reported by De Vita et al. [8] In particular, our approach has achieved an MSE of 0.243 and an MAE of 0.313 for the Transformer, and an MSE of 0.232 and an MAE of 0.144 for the Crossformer. These values are lower than those of De Vita et al. [8] (MSE of 2,851 and MAE of 1,411 for the Transformer; MSE of 1,324 and MAE of 0,827 for the Crossformer), indicating a greater predictive accuracy of our model, with the Crossformer, which proves to be the best in both contexts. However, we have to consider that their work used a different dataset as well as different techniques for data processing.

As shown in Table 4 and Figure 2, the federated learning setup demonstrates a consistent improvement in prediction accuracy over the course of the 20 training rounds. Both MAE and MSE decrease significantly from round 1 to round 20, confirming the effectiveness of the federated learning strategy. In particular, the MAE drops from 0.510 to 0.144, while the MSE decreases from 0.519 to 0.232, indicating a more precise model with fewer large errors as training progresses. This improvement is especially evident after round 10, where the error metrics begin to show a sharper decline. The stability and convergence observed in the metrics suggest that the federated approach not only preserves data privacy reduces communication and computational costs but also maintains high predictive performance. Compared to the baseline values reported in [8], our approach shows better generalization and optimization over time, reinforcing the value of structured federated training even with relatively few rounds and clients.

While the current testbed consists of a limited number of physical weather stations and a reduced amount of data, this design choice was driven by practical constraints related to data availability and the need to ensure controlled, verifiable experimentation. Given the fragmented nature of data access, we integrated multiple technologies for data collection, selecting solutions that are inherently scalable. In particular, the use of containerized services (Docker), publish-subscribe communication protocols (MQTT), and federated orchestration frameworks (such as Flower) provides a solid foundation for future extensions. Although our experiments are currently limited in scale, the presence of a historical database means that additional data can be incorporated into the architecture to validate performance and scalability at a larger scale. Ultimately, this represents a strategic starting point to explore the real-world applicability of our federated framework in meteorological scenarios.

7. Conclusions and Future Work

In this work, we have introduced and developed an innovative application of FL in the context of the collection and analysis of distributed meteorological data. The goal is to improve the accuracy of weather forecasts through collaborative models and to ensure data confidentiality and computing efficiency at peripheral nodes.

We designed and implemented an architecture that leverages modern technologies such as Signal K for real-time data collection, Docker for service containerization, and a Hadoop cluster on Microsoft Azure for federated paradigm simulation. A key contribution of this work is the definition of a complete pipeline that covers data collection and preprocessing up to the collaborative training phase, using advanced FL techniques.

The integration of Transformer and Crossformer models for space-time analysis has demonstrated the effectiveness of FL techniques in real scenarios, overcoming the limitations of data centralization, and paving the way to new safe and cooperative learning scenarios. In particular, it was highlighted that the Crossformer has outperformed the Transformer in space-time data management for weather forecasting. The experimental results show that the Transformer model had an MSE of 0.265 and an MAE of 0.194, while the Crossformer model had an MSE of 0.232 and an MAE of 0.144.

However, an important consideration is about data security, integrity, and reliability. Attacks could potentially originate not only from individual nodes but also from the central coordinating node [11].

Data were collected from distributed meteorological stations of the University of Naples "Parthenope", with a total of 20,199 data from different locations. To ensure data quality and improve the stability of the model, a pre-processing phase has been implemented which includes wind direction coding, smoothing of data, and normalization of numerical features.

In summary, this study not only confirms the effectiveness of Federated Learning and deep learning models such as Transformer and Crossformer in weather forecasting but also proposes a robust and scalable approach for managing sensitive data in distributed environments, opening new perspectives for future applications in cooperative and safe learning environments. Edge computing has proven to be an instrumental tool in optimizing resource utilization, leading to significant economic and communication efficiencies.

It could be interesting to explore possible directions for future work. An interesting path could be starting from the work of De Vita et al. [8] and applying federated transfer learning. This would involve leveraging a pre-trained model while integrating our additional features in the original feature space, as well as across different nodes. This could lead to a more accurate and robust model.

Acknowledge

This research has been partially funded by the “High-performance computing Weather nowcasting with Federated Artificial Intelligence (Hi-WeFAI)” project[1] founded by the National Center ICSC – “National Center for HPC, Big Data and Quantum Computing” – Cascade Call Spoke 9 – “Digital Society & Smart City” – CUP E63C22000980007 ID CODE CN_00000013. The computational infrastructure used to train the predictive model has been provided by the Research Computing Facilities of the University of Naples “Parthenope”.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] E. Strubell, A. Ganesh, A. McCallum, Energy and policy considerations for modern deep learning research, in: Proceedings of the AAAI conference on artificial intelligence, volume 34, 2020, pp. 13693–13696.

[2] H. Kaur, V. Rani, M. Kumar, M. Sachdeva, A. Mittal, K. Kumar, Federated learning: a comprehensive review of recent advances and applications, Multimedia Tools and Applications 83 (2024) 54165–54188.

[3] R. Yu, P. Li, Toward resource-efficient federated learning in mobile edge computing, IEEE Network 35 (2021) 148–155.

[4] F. R. Mughal, J. He, B. Das, F. A. Dharejo, N. Zhu, S. B. Khan, S. Alzahrani, Adaptive federated learning for resource-constrained iot devices through edge intelligence and multi-edge clustering, Scientific Reports 14 (2024) 28746.

[5] G. Giunta, R. Montella, P. Mariani, A. Riccio, Modeling and computational issues for air/water quality problems: A grid computing approach, NUOVO CIMENTO-SOCIETA ITALIANA DI FISICA SEZIONE C 28 (2005) 215.

[6] S. Chen, G. Long, T. Shen, J. Jiang, Prompt federated learning for weather forecasting: Toward foundation models on meteorological data, arXiv preprint arXiv:2301.09152 (2023).

[7] A. Zeng, M. Chen, L. Zhang, Q. Xu, Are transformers effective for time series forecasting?, in: Proceedings of the AAAI conference on artificial intelligence, volume 37, 2023, pp. 11121–11128.

[8] C. G. De Vita, G. Mellone, A. Casolaro, M. G. Orsini, J. L. Gonzalez-Compean, A. Ciaramella, Federated learning and crowdsourced weather data: Practice and experience, in: 2024 IEEE 20th International Conference on e-Science (e-Science), IEEE, 2024, pp. 1–9.

[9] E. Dritsas, M. Trigka, Federated learning for iot: A survey of techniques, challenges, and applications, Journal of Sensor and Actuator Networks 14 (2025) 9.

[10] M. Polato, B. Hammer, F.-M. Schleif, et al., Machine learning in distributed, federated and non-stationary environments-recent trends, ESANN 2024 proceedings (2024) 47–56.

[11] P. M. Mammen, Federated learning: Opportunities and challenges, arXiv preprint arXiv:2101.05428 (2021).

[12] S. Ferretti, L. Cassano, G. Cialone, J. D’Abramo, F. Imboccioli, Decentralized coordination for resilient federated learning: A blockchain-based approach with smart contracts and decentralized storage, Computer Communications 236 (2025) 108112.

[13] H. Zhu, R. Togo, T. Ogawa, M. Haseyama, Prompt-based personalized federated learning for medical visual question answering, in: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2024, pp. 1821–1825.

[14] B. McMahan, D. Ramage, Federated learning: Collaborative machine learning without centralized training data, Google Research Blog 3 (2017).

[1] https://www.hiwefai-project.org

[15] S. Shen, T. Zhu, D. Wu, W. Wang, W. Zhou, From distributed machine learning to federated learning: In the view of data privacy and security, Concurrency and Computation: Practice and Experience 34 (2022) e6002.

[16] C. Gonçalves, R. J. Bessa, P. Pinson, A critical overview of privacy-preserving approaches for collaborative forecasting, International journal of Forecasting 37 (2021) 322–342.

[17] J. D. Fernández, S. P. Menci, C. M. Lee, A. Rieger, G. Fridgen, Privacy-preserving federated learning for residential short-term load forecasting, Applied energy 326 (2022) 119915.

[18] E. Imgrund, T. Eisenhofer, K. Rieck, Adversarial observations in weather forecasting, arXiv preprint arXiv:2504.15942 (2025).

[19] S. Chen, G. Long, T. Shen, J. Jiang, C. Zhang, Federated prompt learning for weather foundation models on devices, arXiv preprint arXiv:2305.14244 (2023).

[20] P. Hamedi, R. Razavi-Far, E. Hallaji, Federated continual learning: Concepts, challenges, and solutions, arXiv preprint arXiv:2502.07059 (2025).

[21] Y. Li, R. Wang, Y. Li, M. Zhang, C. Long, Wind power forecasting considering data privacy protection: A federated deep reinforcement learning approach, Applied Energy 329 (2023) 120291.

[22] S. Ganesan, C. P. Lean, L. Chen, K. F. Yuan, N. P. Kiat, M. R. B. Khan, Iot-enabled smart weather stations: innovations, challenges, and future directions, Malaysian Journal of Science and Advanced Technology (2024) 180–190.

[23] D. Di Luccio, A. Riccio, A. Galletti, G. Laccetti, M. Lapegna, L. Marcellino, S. Kosta, R. Montella, Coastal marine data crowdsourcing using the internet of floating things: Improving the results of a water quality model, IEEE Access 8 (2020) 101209–101223.

[24] R. Montella, D. Di Luccio, L. Marcellino, A. Galletti, S. Kosta, G. Giunta, I. Foster, Workflow-based automatic processing for internet of floating things crowdsourced data, Future generation computer systems 94 (2019) 103–119.

[25] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, Proceedings of Machine learning and systems 2 (2020) 429–450.

[26] S. Dasari, R. Kaluri, 2p3fl: A novel approach for privacy preserving in financial sectors using flower federated learning, CMES-Computer Modeling in Engineering and Sciences 140 (2024) 2035–2051.

[27] L. Zheng, W. Hu, Y. Min, Raw wind data preprocessing: a data-mining approach, IEEE Transactions on Sustainable Energy 6 (2014) 11–19.

[28] S. Wang, J. Cao, S. Y. Philip, Deep learning for spatio-temporal data mining: A survey, IEEE transactions on knowledge and data engineering 34 (2020) 3681–3700.

[29] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI conference on artificial intelligence, volume 35, 2021, pp. 11106–11115.

[30] Y. Zhang, J. Yan, Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting, in: The eleventh international conference on learning representations, 2023.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[32] A. Botchkarev, Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology, arXiv preprint arXiv:1809.03006 (2018).