Titre: Title:	Towards Using Federated Learning to Improve Generalization in Weather Forecasting
Auteur: Author:	Brice Yvan Nanda Assobjio
Date:	2024
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Nanda Assobjio, B. Y. (2024). Towards Using Federated Learning to Improve Generalization in Weather Forecasting [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/61856/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/61856/
Directeurs de recherche: Advisors:	Foutse Khomh
Programme: Program:	Génie informatique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Towards using Federated Learning to improve Generalization in Weather Forecasting

BRICE YVAN NANDA ASSOBJIO
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Décembre 2024

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Towards using Federated Learning to improve Generalization in Weather Forecasting

présenté par **Brice Yvan NANDA ASSOBJIO**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Mohammad HAMDAQA, président
Foutse KHOMH, membre et directeur de recherche
Sarath Chandar ANBIL PARTHIPAN, membre

DEDICATION

To my forever beloved grand-mother,
You left too soon...

ACKNOWLEDGEMENTS

I would like to extend my sincere gratitude to everyone who encouraged me along this research journey.

First and foremost, my deepest appreciation goes to my supervisor, Prof. Foutse Khomh, whose guidance, wisdom, and patience were instrumental in every step of this project. His mentorship allowed me to develop not only as a researcher but also to grow personally and professionally. Prof. Khomh's constructive feedback, thought-provoking insights, and commitment to academic excellence pushed me to strive for the highest standards in my work, and his unwavering support was a source of great motivation and confidence throughout this process.

I am also grateful to the Polytechnique Montreal community for providing a rich, collaborative environment that fostered innovation and growth. I am especially thankful to the incredible team members at SWAT Lab and my peers at Mila, whose camaraderie, discussions, and ideas greatly enriched this research. Each interaction, brainstorming session, and shared experience has contributed to my development and brought fresh perspectives that enhanced the scope and quality of my work.

Acknowledgment is also due to the European Centre for Medium-Range Weather Forecasts (ECMWF) for providing access to ERA5 data, which was essential for the analysis in this thesis. The wealth of meteorological data afforded by ECMWF made rigorous testing and analysis possible, allowing for a thorough investigation into the complex domain of weather forecasting.

Lastly, I am deeply thankful to my friends and family, whose encouragement, patience, and unshakable belief in me provided the resilience needed to persevere. Their understanding and support have been a continuous source of inspiration, and their faith in my abilities has given me the strength to face challenges and work through difficult times with determination and optimism. Their presence has been conducive during this journey, and I immensely thank them for everything they have done to support me.

Thank you all for making this work possible, and for contributing to both my personal and professional growth. This achievement is as much a testament to your support as it is to my efforts.

RÉSUMÉ

Les prévisions météorologiques précises restent un enjeu majeur aux applications variées, allant de la planification des activités quotidiennes à l'agriculture en passant par le transport, et la préparation aux catastrophes. Les techniques de prévision traditionnelles, basées sur des méthodes statistiques et des modèles physiques, rencontrent des difficultés à intégrer la complexité inhérente des systèmes atmosphériques, notamment la nature multidimensionnelle, la non-linéarité, et la variabilité régionale. Avec la croissance des données et les avancées des capacités de calcul, les approches d'apprentissage automatique (ML), en particulier l'apprentissage profond (DL), sont de plus en plus explorées pour la prévision météorologique. Cependant, de nombreux modèles de pointe en DL pour la prévision météorologique reposent sur l'agrégation et l'entraînement centralisés des données, ce qui pénalise les régions présentant une rareté de données ou sous-représentées. De plus, étant donné la diversité des types de climat à travers le monde, le modèle peut rencontrer des difficultés d'adaptation à certaines régions. Ainsi, une approche d'apprentissage fédéré (FL), qui permet l'entraînement du modèle sur des sources de données décentralisées tout en respectant la localité et la confidentialité des données, offre une solution prometteuse pour améliorer la précision des prévisions sans agrégation centralisée des données.

Cette recherche se penche sur l'utilisation de l'apprentissage fédéré pour améliorer les prévisions météorologiques à travers un modèle distribué respectant la nature décentralisée des données météorologiques collectées dans diverses régions géographiques. Plus précisément, nous examinons et comparons plusieurs méthodes d'agrégation FL, incluant FedAvg, FedProx, et SCAFFOLD, pour déterminer laquelle capture le mieux les variations atmosphériques régionales essentielles pour des prévisions précises. L'objectif principal est d'améliorer les prévisions des variables atmosphériques proches de la surface, en utilisant un ensemble de données hétérogène couvrant trois régions distinctes : l'hémisphère Nord, les tropiques et l'hémisphère Sud. Nous visons à établir un cadre capable d'améliorer les prévisions météorologiques à court et moyen termes en mettant l'accent sur la capacité du modèle à capturer les variations régionales et en adaptant les stratégies d'agrégation pour optimiser les performances dans différentes régions geographiques.

Pour évaluer la performance de chaque méthode d'agrégation, nous utilisons le coefficient de corrélation des anomalies (ACC) et l'erreur quadratique moyenne (RMSE). Le RMSE fournit une mesure de la précision absolue des prévisions, tandis que l'ACC évalue la capacité du modèle à capturer les motifs d'anomalies par rapport à la climatologie. En nous concentrant

sur ces métriques, nous visons à quantifier à la fois la précision des prévisions et la capacité du modèle à prédire les écarts par rapport aux conditions climatiques normales, une compétence essentielle pour anticiper les événements extrêmes et les anomalies climatiques. De plus, nous appliquons ces métriques pour évaluer la robustesse du modèle sur des intervalles de prévision étendus, en analysant le comportement du modèle non seulement pour des prévisions immédiates (à $t + \Delta t$) mais aussi pour des prévisions à plusieurs étapes (jusqu'à $t + 2\Delta t$), ce qui est essentiel pour les applications pratiques de la prévision météorologique.

Nos résultats indiquent que certaines techniques d'agrégation surpassent l'approche standard FedAvg, en particulier dans les régions avec une forte hétérogénéité des données, comme les tropiques. SCAFFOLD, avec son mécanisme de correction pour gérer la dérive des clients, montre des améliorations considérables en termes de vitesse de convergence et de précision des prévisions dans les distributions de données non-IID, tandis que les stratégies d'agrégation personnalisées donnent les meilleurs résultats dans les régions présentant des comportements atmosphériques uniques comme dans les tropiques. Ces résultats montrent que l'apprentissage fédéré aide le modèle à mieux s'adapter aux données hors distribution (par exemple, les changements soudains) en raison, par exemple, du changement climatique.

En résumé, cette étude contribue à la littérature croissante à l'intersection de l'apprentissage fédéré et de la prévision météorologique en répondant aux principaux défis de confidentialité des données, d'hétérogénéité régionale, et d'efficacité computationnelle. En améliorant les méthodologies d'apprentissage fédéré pour la prévision météorologique décentralisée, ce travail ouvre la voie à l'utilisation de modèles de prévision précis et respectueux de la confidentialité, capables d'opérer à travers des stations météorologiques distribuées ou des agences météorologiques locales sans compromettre la propriété des données. Les méthodes et résultats présentés ici non seulement soutiennent l'amélioration des prévisions météorologiques, mais offrent également une perspective pour des applications plus larges de l'apprentissage fédéré dans les sciences environnementales, où les données décentralisées et la coopération interrégionale sont primordiales.

ABSTRACT

Accurate weather forecasting remains a crucial endeavor with widespread applications, from daily human activity planning to risk management in agriculture, transportation, and disaster preparedness. Traditional forecasting techniques, which have relied on statistical methods and physics-based models, face challenges in accounting for the inherent complexities of atmospheric systems, such as high-dimensional data, non-linearity, and regional variability. With the surge in big data and advancements in computational capabilities, machine learning (ML) approaches—particularly deep learning (DL)—have gained increasing attention for weather prediction. However, many state-of-the-art DL models for weather forecasting rely on centralized data aggregation and training, penalizing regions with data sparsity and underrepresented regions. Also, given the diversity of climate types across the world, the model could struggle to adapt to some regions. Consequently, a federated learning (FL) approach, which enables model training on distributed data sources while preserving data locality and privacy, offers a promising solution to enhance forecast accuracy without central data aggregation.

This research focuses on leveraging federated learning to enhance weather forecasting through a distributed model that respects the decentralized nature of weather data collected across various geographic regions. Specifically, we investigate and compare multiple FL aggregation methods, including FedAvg, FedProx, and SCAFFOLD, to determine which method best captures regional atmospheric variations critical for accurate forecasting. The primary objective is to improve the forecasting of near-surface atmospheric variables, using a heterogeneous data setup spanning three distinct regions: Northern Hemisphere, Tropics, and Southern Hemisphere. By emphasizing the model's ability to catch regional variations and adjusting aggregation strategies to optimize performance across distinct atmospheric zones, we aim to define a framework that could enhance both short and medium-term weather predictions.

To evaluate the performance of each aggregation method, we employ several metrics, the Anomaly Correlation Coefficient (ACC) and the Root Mean Squared Error (RMSE). RMSE provides a measure of absolute forecast accuracy, while ACC assesses the model's skill in capturing anomaly patterns relative to climatology. By focusing on these metrics, we aim to quantify both the forecast accuracy and the model's ability to predict variations from normal weather patterns, an essential capability for anticipating extreme events and climate anomalies. Additionally, we apply these metrics to evaluate the model's robustness over extended forecast intervals, analyzing the model's behavior not only in immediate predictions

(at $t+\Delta t$) but also in multi-step forecasts (up to $t+2\Delta t$), a crucial requirement for practical weather forecasting applications.

Our findings indicate that certain aggregation techniques outperform the standard FedAvg approach, particularly in regions with high data heterogeneity, such as the tropics. SCAFFOLD, with its correction mechanism to handle client drift, shows significant improvements in forecast accuracy across non-IID data distributions, while personalized aggregation strategies yield the best results in regions with unique atmospheric behaviors such as the tropics. These results demonstrate that federated learning helps the model to better adapt to out-of-distribution (i.e., sudden change) data due to climate change for example.

In summary, this thesis contributes to the growing body of research at the intersection of FL and weather forecasting by addressing key challenges of data privacy, regional heterogeneity, and computational efficiency. By advancing FL methodologies for decentralized weather forecasting, this work opens avenues for deploying accurate, privacy-preserving forecasting models that can operate across distributed weather stations or local meteorological agencies without compromising data ownership. The methods and results presented here not only support enhanced weather forecasting but also provide a pathway for broader applications of FL in environmental sciences, where decentralized data and cross-regional cooperation are paramount.

TABLE OF CONTENTS

DEDICATION iii

ACKNOWLEDGEMENTS iv

RÉSUMÉ v

ABSTRACT vii

TABLE OF CONTENTS ix

LIST OF TABLES xii

LIST OF FIGURES xiii

LIST OF SYMBOLS AND ACRONYMS xiv

CHAPTER 1 INTRODUCTION 1
1.1 Context and Motivation 1
1.2 Problem Statement 2
1.3 Research Objectives 4
1.4 Thesis Outline 4

CHAPTER 2 BACKGROUND 5
2.1 Machine Learning 5
2.2 Deep Learning 6
2.3 Main concepts of Neural Networks 7
2.3.1 Inputs, outputs, and Dataset 7
2.3.2 Activation Function 7
2.3.3 Weight and bias Space, Initialization 8
2.3.4 Forward pass, Loss function, and Backpropagation 9
2.3.5 Optimizer 10
2.3.6 RNN 11
2.3.7 LSTM 13
2.3.8 GRU 14
2.3.9 Stages of Neural Network Learning 16
2.4 Federated Learning 17

2.4.1 Federated Learning Process 17
2.4.2 Benefits of Federated Learning 18
2.4.3 Challenges in Federated Learning 19
2.5 Aggregation Methods in Federated Learning 19
2.5.1 Federated Averaging (FedAvg) 19
2.5.2 Federated Stochastic Variance Reduced Gradient (FedSVRG) 20
2.5.3 Weighted Aggregation (FedProx) 21
2.5.4 SCAFFOLD (Stochastic Controlled Averaging) 21
2.5.5 Trimmed Mean and Krum 22
2.5.6 Personalized Federated Learning (pFedMe) 23
2.6 Weather Forecasting: An Overview 24
2.6.1 Data Sources in Weather Forecasting 24
2.6.2 Numerical Weather Prediction (NWP) 24
2.6.3 Challenges in Weather Forecasting 25
2.6.4 Federated Learning and Weather Forecasting 26

CHAPTER 3 LITERATURE REVIEW 27
3.1 Federated Learning 27
3.2 Applications of FL 28
3.3 Weather Forecasting 29

CHAPTER 4 AGGREGATION TECHNIQUE FOR FEDERATED LEARNING IN WEATHER FORECASTING 33
4.1 Details of the solution 33
4.1.1 Dataset 33
4.1.2 Distributed Structure 35
4.1.3 Training process 37
4.2 Evaluation Metrics: RMSE and Anomaly Correlation Coefficient 39
4.2.1 Root Mean Square Error 39
4.2.2 Anomaly Correlation Coefficient (ACC) 40
4.3 Aggregation technique for Weather Forecasting 40

CHAPTER 5 FEDERATED LEARNING FOR MORE ACCURATE AND ROBUST WEATHER FORECASTING 43
5.1 Details of the evaluation 43
5.2 FL for more accurate weather predictions 43
5.3 Federated Learning for Robustness in Weather Forecasting 44

5.4 Federated Learning for regional improved weather forecast 45
5.5 Discussion and Limitations 47
5.5.1 Internal Validity 47
5.5.2 External Validity 48
5.5.3 Construct Validity 50

CHAPTER 6 CONCLUSION 54
6.1 Summary of Works 54
6.2 Limitations 55
6.3 Future Research 55

REFERENCES 57

LIST OF TABLES

Table 4.1 List of Key Variables in the re-gridded ERA5 Dataset we use in this work. The role indicates either they are just **inputs(I)** or they are **inputs and outputs(I/O)**. 35
Table 5.1 Hyperparameters 43

LIST OF FIGURES

Figure 2.1 Shallow neural network(on the left) vs DNNs(on the right). Source: [1] 6

Figure 2.2 Elman (left) and bidirectional (right) architectures of an RNN, expanded over time. The forward components are in blue, and in red we have the backward components. Source: [2] 13

Figure 2.3 LSTM Block. Source: [3] 15

Figure 2.4 GRU Block. Source: [2] 16

Figure 4.1 Regions delimitation. We employ the same regions and naming way as in the ECMWF scorecards `https://sites.ecmwf.int/ifs/scorecards/scorecards-47r3HRES.html` 36

Figure 4.2 Training process of our models 39

Figure 4.3 Aggregations methods comparison 42

Figure 5.1 Comparison of Federated Pangu-Weather and Pangu-Weather 44

Figure 5.2 Comparison of Federated FourcastNet and FourcastNet 45

Figure 5.3 Comparison of Federated Pangu-Weather and Pangu-Weather against outliers aka change in weather 46

Figure 5.4 Comparison of Federated FourcastNet and FourcastNet against outliers aka change in weather 47

Figure 5.5 Comparison of Federated Pangu-Weather and Pangu-Weather in the ***s.hem*** regions 48

Figure 5.6 Comparison of Federated FourcastNet and FourcastNet in the ***s.hem*** region 49

Figure 5.7 Comparison of Federated Pangu-Weather and Pangu-Weather in the ***tropics*** regions 50

Figure 5.8 Comparison of Federated FourcastNet and FourcastNet in the ***tropics*** region 51

Figure 5.9 Comparison of Federated Pangu-Weather and Pangu-Weather in the ***n.hem*** regions 52

Figure 5.10 Comparison of Federated FourcastNet and FourcastNet in the ***n.hem*** region 53

LIST OF SYMBOLS AND ACRONYMS

IETF	Internet Engineering Task Force
OSI	Open Systems Interconnection
ANN	Artificial Neural Network
DNN	Deep Neural Network
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Units
ML	Machine Learning
DL	Deep Learning
FL	Federated Learning
HIPAA	Health Insurance Portability and Accountability Act
NWP	Numerical Weather Prediction
IID	Identically Distributed
MSE	Mean Square Error
RMSE	Root Mean Square Error
ACC	Accumulated Correlation Coeficient
DP	Differential Privacy
SMC	Secure Multiparty Computation
ARIMA	AutoRegressive Integrated Moving Average (ARIMA)
ECMWF	European Centre for Medium-Range Weather Forecasts

CHAPTER 1 INTRODUCTION

In this thesis, we investigate the use of federated learning in the weather forecasting task. Our objective is to show that the federated approach has an advantage in implementing weather forecasting systems based on machine learning. We evaluate its contribution by comparing it to centralized architectures with ACC and RMSE. Even if in this thesis, we work with two models due to computational constraints, we are confident that it could be expanded to other weather forecasting models.

1.1 Context and Motivation

In recent years, advancements in computational power and machine learning techniques have driven significant improvements in weather forecasting. Accurate and timely weather forecasts are critical for numerous sectors, including agriculture, disaster preparedness, transportation, and energy management. Traditionally, weather prediction has relied on physical models known as Numerical Weather Prediction (NWP) systems, such as the European Centre for Medium-Range Weather Forecasts (ECMWF) model. These models simulate the atmosphere's behavior by solving complex differential equations that govern atmospheric dynamics. While highly accurate for large-scale predictions, NWP systems can suffer from limitations in capturing fine-scale atmospheric phenomena, particularly in regions with sparse data availability. Weather forecasting is inherently a data-intensive task, it involves vast quantities of data from several sources, including satellites, ground-based stations, aircraft, and weather balloons. These observations feed into NWP models to provide global or regional forecasts. However, the fast evolution of artificial intelligence (AI) and ML offers promising alternatives and complementary approaches to traditional forecasting. ML, particularly DL, has shown the ability to model complex patterns in large datasets, capturing spatiotemporal relationships that are difficult for classical models to resolve. AI-based models offer distinct advantages over NWP systems in computational efficiency and scalability. For instance, DL models like recurrent neural networks (RNNs) and convolutional neural networks (CNNs) are highly effective at learning from large datasets and making predictions faster than NWP models, which can take hours of supercomputing time to process. An example of this shift is the development of models such as **FourCastNet** [4], **pangu-weather** [5] and **GraphCast** [6], which leverage neural networks to produce global weather forecasts with remarkable efficiency, outperforming traditional NWP methods in terms of computational speed while preserving comparable accuracy.

However, these models show mixed results depending on the region: for poorer areas of the world, such as the tropics and the Middle East, due to factors such as data sparsity, the quality of data collected locally, or the difference in climate with Western countries for example during the winter in western countries in tropics we have the dry season. On the other hand, these models have difficulty keeping up with climate change, particularly changes that affect only particular regions. The issue of data heterogeneity is especially pronounced in weather forecasting since Weather data can vary significantly depending on the region, type of sensor, and environmental factors such as altitude, terrain, or proximity to bodies of water. For instance, a weather station in a coastal area may experience data patterns entirely different from those in a mountainous or desert region. This heterogeneity complicates centralized approaches that assume data uniformity.

In parallel with advancements in AI, FL has emerged as a learning approach that preserves data privacy and deals very well with data heterogeneity [7–9]. This is particularly relevant in weather forecasting, where data sources are often geographically distributed, and sharing raw data can be restricted due to privacy concerns or logistical challenges. FL allows models to be trained collaboratively across multiple institutions or data centers, without needing to centralize the data. This offers a powerful approach to improving weather forecasting, especially in regions where data is sparse or isolated.

1.2 Problem Statement

Accurate weather forecasting is a critical need across numerous sectors, from agriculture and disaster management to aviation and energy production. Despite significant advancements in meteorological science, the challenge of predicting weather accurately at various scales and timeframes persists. Current models, particularly Numerical Weather Prediction (NWP) systems, remain computationally expensive and often struggle with fine-scale, short-term forecasts. These models, though robust in predicting large-scale patterns, can have difficulty capturing localized phenomena such as thunderstorms, hurricanes, and other extreme weather events. A central challenge in weather forecasting is the integration and analysis of vast amounts of observational data from diverse sources, including satellite imagery, radar, ocean buoys, weather balloons, and ground stations. This data, essential for feeding predictive models, is often spatially and temporally heterogeneous, meaning it varies in its quality, resolution, and frequency of collection. For example, satellite data may cover large regions but lack detailed information at a local scale, while ground-based weather stations provide high-quality, precise data but only for specific locations. The disparity in resolution and coverage introduces significant complexity to the forecasting process. In addition to the data

heterogeneity, there are logistical and privacy concerns. Data in weather is often collected by a wide array of institutions globally, and centralizing this data is not always feasible or desirable. This presents challenges for traditional machine learning approaches, which typically rely on centralized datasets for training. The inability to effectively transmit raw data due to jurisdictional boundaries hampers or privacy concerns the creation of models that can leverage diverse and widespread datasets for improved accuracy. The emergence of deep learning techniques, such as CNNs and RNNs, has provided new ways of addressing some of these issues. These models can learn complex patterns within large datasets and have demonstrated success in capturing both spatial and temporal relationships. However, the reliance on centralized data remains a significant bottleneck, particularly for meteorological applications that require enormous amounts of heterogeneous data to perform high predictive accuracy. FL offers an advantageous solution to these challenges by allowing models to be trained across distributed datasets without requiring data to be moved or centralized. Instead, models are trained locally at each data source or subregion, and only the learned parameters (such as gradients) are shared and aggregated to form a global model. This approach allows collaboration between different institutions, making it particularly appealing for weather forecasting, where data may be spread across different regions and organizations.

Despite its potential, the application of federated learning in weather forecasting remains under-explored. Several key challenges must be addressed to make FL practical and effective in this domain. First, the inherent heterogeneity of weather data, both in terms of spatial and temporal resolution, must be carefully managed. Effective aggregation methods combining locally trained models from diverse and sometimes conflicting datasets are essential. Additionally, the dynamic nature of weather systems requires models that can adapt over time to new data and changing conditions. Moreover, federated learning faces the challenge of model accuracy. While traditional centralized learning benefits from pooling all available data, federated learning must overcome the limitations of working with fragmented datasets. The performance of the resulting global model depends heavily on the diversity and quality of the local data. Therefore, this thesis seeks to explore various aggregation techniques within the federated learning framework, such as Federated Averaging (FedAvg) and SCAFFOLD, to evaluate their suitability for weather forecasting applications. This research aims to investigate how federated learning can be adapted to weather forecasting by addressing the unique problems posed by heterogeneous data, privacy constraints, and climate change. By evaluating the performance of the federated approach against the traditional centralized approach and numerical weather prediction systems, this thesis will contribute to the growing body of knowledge on machine learning applications in meteorology. Additionally, it will explore ways to optimize FL models for long-term accuracy and scalability, ensuring that they can

handle the complex and evolving nature of weather data. Through this work, the goal is to bridge the gap between the capabilities of advanced ML techniques and the operational needs of weather forecasting systems, ultimately leading to more robust, and accurate predictions that can aid society.

1.3 Research Objectives

Answering the following research questions (RQs) is the aim of this thesis:

1. **RQ1:** What aggregation technique is most suitable for weather forecasting tasks using FL?
2. **RQ2:** Can FL approach methodology help improve weather forecasting models' performances?
3. **RQ3:** Can FL help improve ML-based weather forecasting models' robustness against climate change(which we define here as sudden changes in the weather)?
4. **RQ4:** Can FL help improve weather forecasting accuracy in low-resource regions of the globe?

1.4 Thesis Outline

In the following sections we will try to answer the previous research questions, the plan of the thesis is as follows:

- Chapter 2 presents the essential concepts required for understanding this research.
- Chapter 3 provides a comprehensive examination of the most recent developments in federated learning and the methods employed in machine learning for weather forecasting.
- Chapter 4 describes our methodology and approach and details the performed experiments in order to answer the previous RQs.
- Chapter 5 presents and examines the results obtained.
- Chapter 6 provides ideas for future work as well as further enhancements.

CHAPTER 2 BACKGROUND

This chapter presents and explains the basic and key concepts to understand our work.

2.1 Machine Learning

Machine Learning is a subset of AI that focuses on designing systems or models that can learn from data and improve performance over time without being explicitly programmed. It involves creating algorithms that process and analyze data to detect patterns, make decisions, or predict. ML algorithms are used in a vast variety of domains, such as medicine, aviation, and education.

Depending on the data, the task, and how we proceed, ML approaches are divided into four sorts:

- **Supervised learning**: The dataset provided to the computer is in the form $D = \{(x_i, y_i)\}_{i=1}^{N}$, and the goal of the task is to the computer to map an input x_i to its output also called label y_i.
- **Unsupervised learning**: Here the is just made of inputs; its in the form $D = \{x_i\}_{i=1}^{N}$, and the task here is to identify or find patterns that the data can hide.
- **Semi-supervised learning** : In this approach the dataset is made of a small portion data with output and a large portion of data with no outputs. Semi-supervised learning falls between unsupervised learning and supervised learning. The labeled data guides the learning process, while the unlabeled data helps improve the model's generalization by leveraging patterns and structures in the data.
- **Reinforcement learning**: the computer program called agent operates within a changing environment to achieve a specific goal. During its interactions, the agent receives feedback in the form of rewards, which it seeks to maximize. The process can be represented as (s_t, a_t, r_t, s_{t+1}), where at time t: s_t is the current state of the environment, a_t is the action the agent decides to take, r_t is the reward obtained after executing the action, and s_{t+1} is the resulting state of the environment. The primary objective is to address an optimal control problem by identifying the most beneficial actions in each state to maximize cumulative future rewards.

Other techniques have emerged that do not cleanly fit into this four-fold classification, and the same machine learning system may employ more than one for example meta learning or dimensional reduction.

2.2 Deep Learning

DL is a sub-field of machine learning focused on artificial neural networks(ANNs), which are algorithms that are modelled following the structure and functioning of the brain.

An ANN is a supervised learning system composed of a large number of basic units known as neurones or perceptrons. Each neurone may make basic judgements and transmit those decisions to other neurones, which are organised in linked layers. Given enough training data and processing capacity, the neural network can simulate virtually any function and answer almost any query. A "shallow" neural network is neural network made of three layers of neurons: An input layer takes the model's independent variables or inputs, an hidden layer and an output layer for making predictions. A Deep Neural Network (DNN) has a same structure, with two or more hidden layers of neurons processing inputs (Figure 2.1). Goodfellow et al. [10] demonstrated that, while shallow neural networks can handle difficult issues, DL networks get more accurate and better when more neuron layers are added. Additional layers are beneficial up to a threshold of nine or ten, beyond which their predictive ability begins to deteriorate. Today, most neural network models' implementations employ a deep network with three to ten neuron layers.

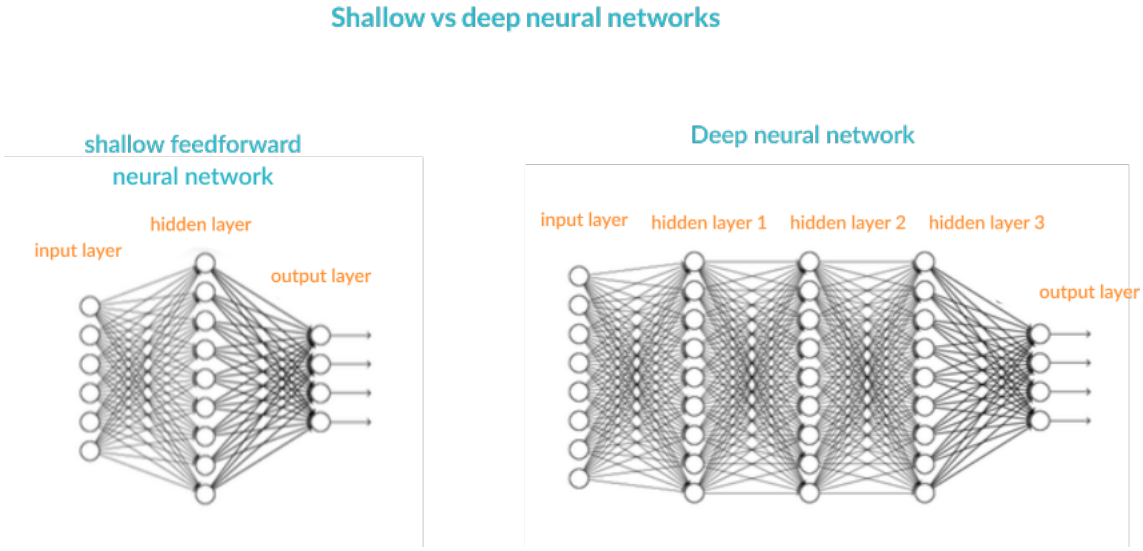

Figure 2.1 Shallow neural network(on the left) vs DNNs(on the right). Source: [1]

2.3 Main concepts of Neural Networks

2.3.1 Inputs, outputs, and Dataset

- **Input** ($x \in \mathbb{X}$): The initial data provided to the neural network, used as the basis for deriving a prediction or decision.
- **Output** ($y \in \mathbb{Y}$) : The result generated by the network, which can vary in type, such as a real value, a value between 0 and 1, a categorical label, or a numerical value.

  For these examples of task we have a specific $\mathbb{Y}$:

  - ◇ binary classification $\mathbb{Y} = \{0, 1\}$
  - ◇ regression $\mathbb{Y} = \mathbb{R}$
  - ◇ multiclass classification $\mathbb{Y} = \{0, ..., C-1\}$
- **Dataset** : the set of inputs x_i (and outputs y_i in the case of supervised learning) forms the dataset $D = \{x_i\}_{i=1}^N$ (resp. $D = \{(x_i, y_i)\}_{i=1}^N$) with N the numbers of samples in the dataset.

  This dataset can be divided into three parts, $D = D_{train} \cup D_{val} \cup D_{test}$:

  - ◇ The set of data that will be used to train the model : training set (D_{train})
  - ◇ The set of data on which the model will be evaluated during its training (used for optimizing the model, hyperparameter tuning, early stopping, etc ...): validation set (D_{val})
  - ◇ The set of data on which the model will be tested after training (used for the final evaluation): test set (D_{test})

2.3.2 Activation Function

An activation function introduces non-linearity in the neural network, allowing it to capture complex patterns and relationships in the data, making it suitable for solving intricate tasks. A neural network without an activation function behaves like a linear regression model, as it processes input data through purely linear transformations.

- **Sigmoid**

  It has a smooth gradient and produces outputs between 0 and 1. However, when the input values are extremely high or low, the function becomes almost flat, causing very

slow updates during the training process. This issue, known as the vanishing gradient problem, makes it difficult for the network to learn effectively in these regions.

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

- **Tanh**

  The Tanh (hyperbolic tangent) function is zero-centered, meaning that its output values range from -1 to 1. Its zero-centered property allows the Tanh function to handle inputs that are strongly negative, strongly positive, or neutral more effectively. It also helps to ensure that gradients during backpropagation are more evenly distributed, which can lead to faster convergence during training.

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

- **ReLu (Rectified Linear Unit)**

  The ReLU [11] function is known for its computational efficiency because it only involves a simple thresholding operation: if the input is positive, it is passed through as is; if negative, the output is zero. However, a key limitation of ReLU is that it does not handle negative or zero values.

$$ReLu(x) = max(x, 0)$$

- **Softmax**

  The Softmax function is commonly used in the output layer of neural networks, particularly for multi-class classification problems. It transforms the output of the network into a probability distribution(i.e. the sum of its output values across all is 1), normalizing the raw output scores into values between 0 and 1. For a vector $x \in \mathbb{R}^n$, we have $softmax(x) \in [0, 1]^n$ with

$$softmax(x)_i = \frac{exp(x_i)}{\sum_{j=1}^{n} exp(x_j)}$$

2.3.3 Weight and bias Space, Initialization

Each neuron is assigned a numerical weight. Weights given to the activation function dictate each neuron's output. When training a DL model, the goal is to determine which weights produce the most accurate results.

Let's take the case of a two layers network (a network without hidden layers), which takes $x \in \mathbb{R}^n$ as input and returns an output $y \in \mathbb{R}^m$ such as $y = \sigma(xW^T + b)$ with σ the activation function $W \in \mathbb{R}^{m \times n}$ the weights and $b \in \mathbb{R}^m$ the bias.

A common method in neural networks training is to randomly initialize the weights at the beginning and then optimize from there. Another strategy is the Xavier optimization [12], which ensures that weights are just right to guarantee sufficient signal flows through all network tiers. For example, Sitzmann et al. [13] draw models weights according to a uniform distribution: $W \sim \mathcal{U}(-fan\_in, fan\_in)$ for first layer and $W \sim \mathcal{U}(-\sqrt{6/fan\_in}, \sqrt{6/fan\_in})$ for others layers, where fan_in is the input dimension of the concerned layer. Other practitioners use the normal distribution instead of the uniform distribution.

2.3.4 Forward pass, Loss function, and Backpropagation

The **forward pass** refers to the process of passing input data through the network to generate predictions or outputs. During the forward pass, the input is passed layer by layer. Its done by applying the two following step at each layer: Linear transformation, applying activation function.

A **loss function** is any function

$$\begin{array}{rccc} \mathcal{L} & : \mathbb{Y} \times \mathbb{Y} & \rightarrow & \mathbb{R}_+ \\ & (y^{'}, y) & \mapsto & \mathcal{L}(y^{'}, y) \end{array}$$

verifying

$$\begin{cases} \mathcal{L}(y^{'}, y) = 0 & \text{if } y = y^{'} \\ \mathcal{L}(y^{'}, y) > 0 & \text{otherwise} \end{cases}$$

In machine learning we have several commonly used **loss functions**, so here is a non-exhaustive list.

- **Mean Squared Error (Regression Loss)** : $\mathbb{Y} = \mathbb{R}^n$

$$MSE(y^{'}, y) = \frac{1}{n} \sum_i (y_i^{'} - y_i)^2$$

- **Mean Absolute Error (Regression Loss)** : $\mathbb{Y} = \mathbb{R}^n$

$$MAE(y^{'}, y) = \frac{1}{n} \sum_i |y_i^{'} - y_i|$$

- **Negative Log Likelihood (Classification Loss) or Cross Entropy Loss** : $\mathbb{Y} = [0,1]^n$

$$CE(y^{'}, y) = -\frac{1}{n}\sum_{i} y_i log(y_i^{'})$$

Backpropagation is a key method used in neural networks to compute the gradient of the loss function ($\mathcal{L}$) with respect to the model's parameters (θ). This gradient is then utilized by an optimizer to adjust the model's parameters during training. By propagating the error backward from the output layer to the input layer, backpropagation allows the network to iteratively reduce the loss and improve its predictions over time. It is also called backward propagation errors.

$$\nabla_\theta \mathcal{L} = \frac{\partial \mathcal{L}}{\partial \theta} = \left\{ \frac{\partial \mathcal{L}}{\partial w^k} = \left(\frac{\partial \mathcal{L}}{\partial w_{ij}^k} \right)_{i \sim columns, j \sim lines}, \frac{\partial \mathcal{L}}{\partial b^k} = \left(\frac{\partial \mathcal{L}}{\partial b_i^k} \right)_{i \sim columns} \right\}_{k \sim layers}$$

The **backwards** part of the term comes from the fact that the gradient is calculated in reverse order across the network, starting from the last layer and moving toward the first. The gradient for the final layer is computed first, followed by the calculation of the gradient for preceding layers. This process allows for the reuse of partial gradient computations from each layer in the calculation for the layer before it. This reverse flow of error information enables a more efficient gradient computation compared to the less efficient method of calculating each layer's gradient independently.

This algorithm relies on the chain rule, which states: If a variable z depends on y, and y its turn depends on x, then z indirectly depends on x through y. The chain rule expresses this relationship by stating that the derivative of z with respect to x can be obtained by multiplying the derivative of z with respect to y by the derivative of y with respect to x, i.e.,

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$$

Thus, the derivative with respect to the model output of the loss is first calculated, and used to calculate the derivative with respect to the model's last layer parameters. This derivative is then used to compute the derivative of the loss with respect to the second last layer's parameters, and so on until the first layer of the model.

2.3.5 Optimizer

An optimizer is an algorithm used in machine learning to minimize the loss function by adjusting the weights of the model during training. The primary goal of an optimizer is to

update the model parameters iteratively to reduce the difference between the predicted and actual outputs.

Some common optimizers include:

- **Stochastic Gradient Descent (SGD):** A basic optimizer that updates weights after each training example, making it computationally efficient but sometimes slow to converge.
- **Momentum:** An improvement on SGD that incorporates the gradient from previous iterations to accelerate convergence and reduce oscillations.
- **Adam (Adaptive Moment Estimation):** A widely used optimizer that adjusts learning rates for each parameter, making it effective for complex problems with large datasets.
- **RMSProp:** Designed for noisy gradients, it adapts the learning rate by averaging the squared gradients over time, helping models to converge more quickly in such cases.

The optimizer plays a crucial role in the training process by gradually improving the model's performance and minimizing the loss function.

2.3.6 RNN

Recurrent Neural Networks (RNNs) are a versatile and expressive neural architecture capable of processing sequential data, making them well-suited for tasks like language modeling. An RNN takes an input sequence of vectors, $x = [x_1, \ldots, x_n]$, and processes them sequentially to produce an output sequence, $y = [y_1, \ldots, y_m]$. At each step t, an RNN updates its internal memory or hidden state h_t, which encodes information about the sequence observed so far, $x_{1:t}$. This hidden state can be seen as a summary representation of the partial sequence.

The defining characteristic of an RNN is its recurrence relation, which governs how the hidden state is updated. In its simplest form, the vanilla RNN computes the hidden state and output as:

$$h_t = f(x_t, h_{t-1})$$

$$y_t = g(h_t)$$

Here:

- $h_t \in \mathbb{R}^H$ represents the hidden state at time t,

- f is a non-linear function, often implemented using activation functions like the logistic sigmoid or gated mechanisms,
- g is an output function, such as a softmax layer for classification tasks.

The initial hidden state, h_0, is typically initialized to $0 \in \mathbb{R}^H$, though other initializations are possible. This iterative process allows RNNs to efficiently model temporal dependencies in data.

Different choices on f and g lead to different architectures of the RNNs proposed in the literature, for instance, Jordan or Elman networks [14]. The left side of Figure 2.2 shows an Elman RNN unfolded in time. Regular RNNs have two main drawbacks: first, they struggle with the vanishing gradient problem [15], which makes them unable to model long-term relationships. This problem is tackled by using special types of recurrent units (LSTM, GRU, etc). The second problem is that they process input sequences in a single direction, typically from left to right. This unidirectional approach prevents the model from utilizing contextual information from future inputs (i.e., information from right to left). To address this issue and incorporate context from both directions, Schuster et al. [16] introduced bidirectional RNNs.

Bidirectional RNNs consist of two distinct recurrent layers: the *forward layer*, which processes the input sequence in its natural order (from 1 to n), and the *backward layer*, which analyzes the sequence in reverse order (from n to 1). These layers function independently, with no interaction between their respective hidden states, allowing training to follow algorithms similar to those used for standard unidirectional RNNs. Using prior notation, a bidirectional RNN can be described as:

$$h_t^f = f(x_t, h_{t-1}^f)$$

$$h_t^b = f(x_t, h_{t-1}^b)$$

$$h_t = [h_t^f, h_t^b] = f(x_t, h_{t-1})$$

$$y_t = g(h_t)$$

Here, $[.,.]$ represents vector concatenation, h_t^f is the hidden state for the forward layer, and h_t^b is the hidden state for the backward layer. The model's output combines information from both layers, enabling it to leverage context from both directions in the sequence. The right part of Figure 2.2 illustrates a bidirectional Elman network expanded over time.

A commonly used definition for f is shown below, where σ denotes a non-linear activation

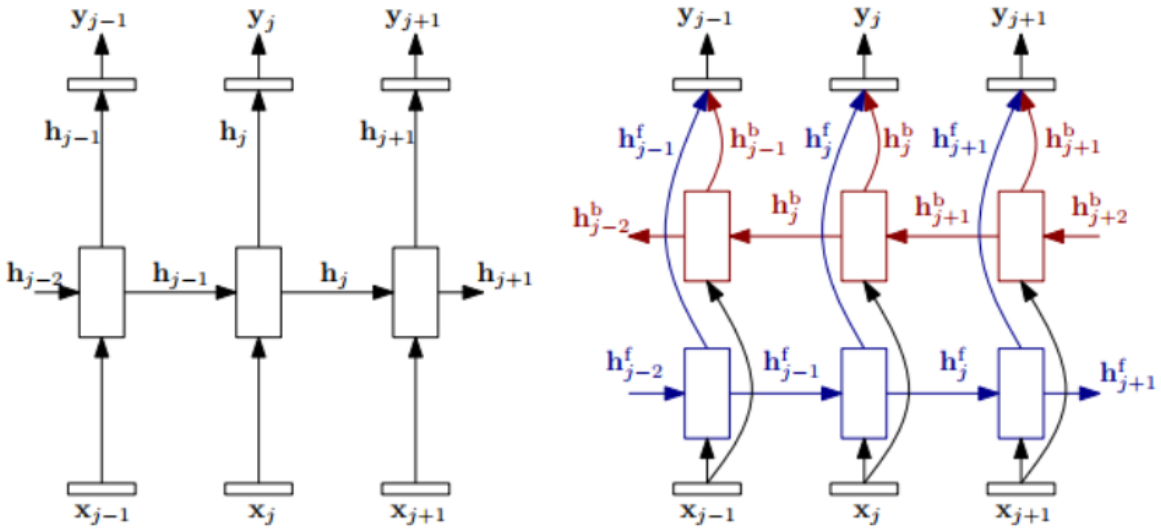

Figure 2.2 Elman (left) and bidirectional (right) architectures of an RNN, expanded over time. The forward components are in blue, and in red we have the backward components. Source: [2]

function such as the sigmoid or tanh function.

$$f(x_t, h_{t-1}) = \sigma(W_{xh}x_t + W_{hh}h_t + b_h)$$

At each timestep t, an RNN can optionally emit an output symbol y_t, which can either be discrete or real-valued. For the case of discrete outputs, often encountered in linguistic tasks, a probability distribution p over a set of output classes Y is computed as:

- ◇ score vector/logits : $s_t = W_{hy}h_t + b_s$
- ◇ probability vector : $p_t = softmax(s_t)$

The softmax function converts the score vector s_t into a probability vector p_t.

2.3.7 LSTM

While the RNNs are theoretically straightforward, there are certain problems like the vanishing gradient issue(due to its nature, recurrent networks propagate the error gradient back in time in multiple steps. Depending on the recurrent weight matrix and the activation function, gradients may rapidly tend to be zero. Hence, the influence of the current time-step on

a far, previous one becomes almost zero, and long relationships in the sequence are lost) and the exploding gradient problem.

To address the issue of vanishing gradients, one approach is to use specialized neural network architectures that are designed to prevent this problem by maintaining a gradient value of one during backpropagation. LSTMs [17] are specifically engineered to capture long-range dependencies in sequential data by using memory cells that store information over extended periods, mitigating the vanishing gradient problem. The core concept behind LSTMs is the introduction of a memory cell, c, in addition to the standard hidden state, h, used in traditional neural networks. This memory cell's gradient, dc_t/dc_{t-1}, is maintained at exactly one, which ensures that the information stored in the cell does not degrade due to vanishing gradients. This characteristic allows LSTMs to capture long-range dependencies in data much more effectively than standard RNNs, which struggle with preserving information over long time sequences.

Figure 2.3 shows the LSTM architecture and its equations are follows:

$$u_t = tanh(W_{xu}x_t + W_{hu}h_{t-1} + b_u)$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

$$c_t = i_t * u_t + c_{t-1}$$

$$h_t = o_t * tanh(c_t)$$

$*$ denotes the element-wise multiplication.

2.3.8 GRU

For tackling the vanishing gradient problem, gated units were developed and used as a hidden state function (f in previous equations). Analogously to the vanishing gradient, if gradients are large, the exploding version of the problem appears, producing numerical instability and learning issues. Fortunately, by clipping the to a predefined value the maximum of the gradients during training, this problem is solved [18]. Gated units are able to cope with long-term relationships by deciding, at each time step, how much information flows from the previous to the current steps and how much information from the input is taken into account in the current time step. An illustration of such a function can be found in Figure 2.4. We can see this type of cells as a simpler version of LSTM units. In a GRU cell, the hidden state

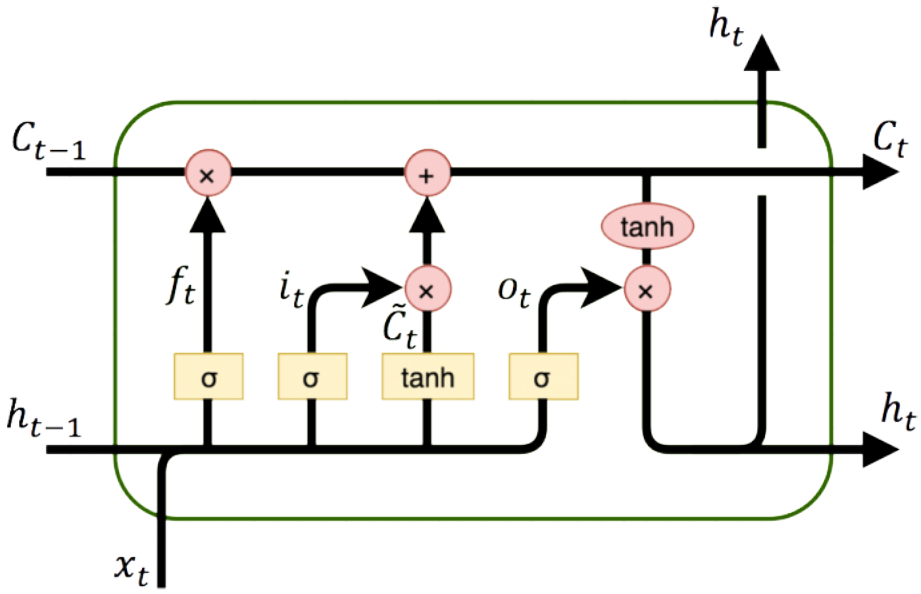

Figure 2.3 LSTM Block. Source: [3]

h_t depends on an updated state $\tilde{h}_t \in \mathbb{R}^H$ and the previous hidden state h_{t-1} both modulated by an update gate $z_t \in \mathbb{R}^H$:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

with $*$ the element-wise multiplication.

The updated state $\tilde{h}_t$ depends on the previous state h_{t-1} and the current input x_t, which is regulated by a reset gate $r_t \in \mathbb{R}^H$:

$$\tilde{h}_t = tanh(Wx_t + U[r_t * h_{t-1}] + b_h)$$

where $W, U \in \mathbb{R}^{H \times H}$ are respectively the previous hidden state and the weight matrices of the input.

The reset and update gates are calculated as follows:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

where $W_r, U_r \in \mathbb{R}^{H \times H}$ represent the weight matrices of the reset gate; $W_z, U_z \in \mathbb{R}^{H \times H}$ represent the weight matrices of the update gate and σ represent the logistic sigmoid function(element-

wise).

The architecture of a GRU block is shown in Figure 2.4.

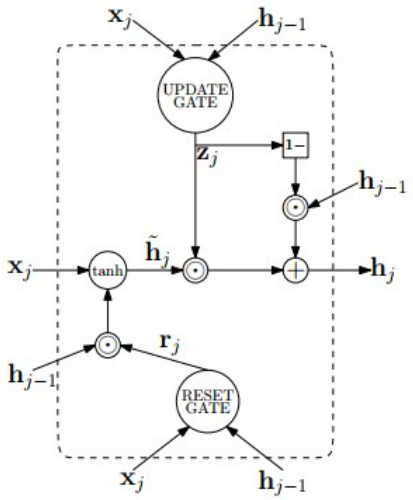

Figure 2.4 GRU Block. Source: [2]

2.3.9 Stages of Neural Network Learning

The process of training a neural network or deep learning model typically unfolds in six main steps:

1. **Initialization**: Initial weights are assigned to all neurons in the network.
2. **Forward Propagation**: The inputs from the training set are passed through the neural network to compute an output.
3. **Error Function**: The error function is defined to capture the difference between the model's output and the correct output, measuring how far off the model is.
4. **Backpropagation**: The purpose of backpropagation is to adjust the weights of the neurons to minimize the error function.
5. **Weight Update**: The weights are updated based on the feedback from the backpropagation process, moving them toward optimal values.
6. **Iteration until Convergence**: Since the weights are adjusted in small steps, multiple iterations are necessary for the network to learn. After each iteration, the weights are updated in a direction that reduces the overall error. The number of iterations required

for convergence depends on factors such as the learning rate, network parameters, and the chosen optimization method.

Once the model has been trained through these stages, it is ready to generate predictions when fed with new input data, using a forward pass to compute the output.

2.4 Federated Learning

As data becomes increasingly valuable across industries, the need to develop machine learning models taking into consideration data security and privacy has become paramount. FL is an emerging paradigm in decentralized machine learning that addresses these concerns by allowing to train the model across multiple decentralized servers or devices, where data are stored locally. It differs fundamentally from traditional centralized learning, where data from all sources is collected centrally in a repository for model training. In federated learning, only the model updates, such as the learned parameters, are sent to a central server for aggregation, thereby preserving the privacy of the original data.

First introduced in 2017 by Google and FL has since been applied in various sectors where privacy in particular in data is critical, such as finance, healthcare, and mobile device ecosystems. By keeping the data decentralized, FL allows organizations to leverage the power of ML and DL without the risks associated with moving sensitive information across networks. In healthcare for example, federated learning has been used to train predictive models on sensitive patient records without compromising patient confidentiality by transmitting raw medical data. Similarly, in mobile ecosystems, FL is used to improve predictive text or voice recognition models directly on users' devices without collecting personal information centrally.

2.4.1 Federated Learning Process

The core of FL is a process that allows multiple clients (e.g., local servers, devices, or sensors) to collaboratively train a machine learning model without needing to share their local data. The process generally follows these steps:

1. **Local Training**: Each client or local model trains a model indepentdently on its local dataset. This step ensures that the data never leaves the client's premises, maintaining privacy.

2. **Model Update**: Once local training is complete, each clients sends only their model

updates (such as gradients and weighs) to the central server. The actual data remains decentralized and is not shared.

3. **Model Aggregation**: After receiving the updates the central server aggregates the received updates from the participating clients, using an aggregation technique to update the global model. This aggregated global model incorporates insights from each client's data without exposing the individual datasets.

4. **Model Distribution**: The new updated parameters are then sent back to each client, who use it the next round of local training. This process is repeated until the global model converges or for a fixed number of rounds.

2.4.2 Benefits of Federated Learning

FL offers several some advantages over conventional centralized machine learning approaches, particularly in situations where data privacy, communication costs, and heterogeneity of data sources are key concerns:

- **Data Privacy**: FL minimizes the risk of exposing sensitive data by keeping the data local. This is very important especially in industries with stringent data privacy regulations, such as healthcare (HIPAA) or finance (GDPR).

- **Reduced Communication Costs**: By sharing only model updates instead of entire datasets, FL significantly reduces the bandwidth required for communication. This is crucial in applications where moving large volumes of data is expensive or impractical, such as in mobile or IoT networks.

- **Data Decentralization**: FL is well-suited for environments where data is naturally decentralized, such as edge computing networks or geographically dispersed sensors. In these cases, FL can harness the diverse, localized datasets to improve model performance without the need for centralized data storage.

- **Adaptation to Data Heterogeneity**: Weather data, for example, often varies in quality and format based on the source, location, and type of sensor used. FL allows clients to train on their unique dataset local models, enabling the system to account for local variations while contributing to a globally improved model.

2.4.3 Challenges in Federated Learning

While FL offers promising advantages, several challenges remain, particularly in environments like weather forecasting, where data comes from diverse sources and must be processed quickly:

- **Communication Overhead**: Though FL reduces the need for transferring large datasets, the repeated communication of model updates can become expensive, especially in scenarios where bandwidth is limited or model updates are frequent.
- **Data Heterogeneity**: The data held by each client can be vastly different in terms of quantity, quality, and distribution, often leading to imbalanced updates. This can result in convergence issues or reduced model accuracy if not properly addressed through robust aggregation techniques.
- **Privacy Concerns**: Although FL improves privacy by keeping data decentralized, it is still possible to infer sensitive information from model updates (a phenomenon known as model inversion attacks). Addressing these concerns often requires additional techniques such as secure multiparty computation or differential privacy.
- **Model Convergence**: Due to the asynchronous nature of client participation (some clients may train on their local data more frequently than others), ensuring the convergence to an optimal solution of the global model can be difficult. Techniques like weighted aggregation and client sampling can help mitigate this issue.

2.5 Aggregation Methods in Federated Learning

An important aspect of federated learning is aggregating local models into the global model. This aggregation step is crucial to ensure that insights from decentralized datasets are integrated without sharing raw data. Below are several common and relevant aggregation methods in FL.

2.5.1 Federated Averaging (FedAvg)

Federated Averaging (FedAvg) is the most commonly used aggregation technique in FL, introduced by McMahan et al. [7]. It averages the model updates from all participating clients, with the updates weighted by the size of the local dataset on which each client is trained.

The global model update w^t is computed as:

$$w^t = \sum_{i=1}^{N} \frac{n_i}{\sum_{j=1}^{N} n_j} w_i^t$$

where:

- w_i^t represents the local model update at round t from client i ,
- n_i is the size of client i's local dataset,
- N is the total number of clients.

Advantages:

- **Simplicity**: Easy to implement and computationally efficient.
- **Scalability**: Works well with many clients and large model parameters.

Challenges:

- **Data Heterogeneity**: FedAvg struggles when data across clients is not identically distributed(non-iid), which is often the case with weather data from different regions.

2.5.2 Federated Stochastic Variance Reduced Gradient (FedSVRG)

FedSVRG aims to reduce the variance of local updates, which can help accelerate convergence in heterogeneous data settings. It introduces a variance reduction term that corrects the difference between local and global gradients, stabilizing the training process.

Advantages:

- **Faster Convergence**: By controlling the variance of updates, FedSVRG allows for more stable and faster convergence.
- **Handling Heterogeneity**: More suitable for non-iid data than FedAvg.

Challenges:

- **Increased Complexity**: It introduces additional variance reduction steps, making it more computationally complex than FedAvg.

2.5.3 Weighted Aggregation (FedProx)

FedProx extends FedAvg by introducing a proximal term to the objective function, penalizing updates that diverge too far from the global model. This makes FedProx more robust to system and data heterogeneity.

The local objective function in FedProx is modified as follows:

$$minh_i(w, w^t) = F_i(w) + \frac{\mu}{2}\|w - w^t\|^2$$

where:

- w^t is the global model,
- w is the local model,
- μ is the regularization parameter,
- $F_i(w)$ the loss function computed over the local data of client i

Advantages:

- **Robustness to Heterogeneity**: FedProx is designed better deal with data heterogeneity. It better handles non-iid data than FedAvg.
- **Flexibility**: The level of regularization can be tuned based on the degree of heterogeneity.

Challenges:

- **Tuning**: Choosing an appropriate value for μ requires experimentation.
- **Computational Overhead**: The proximal term adds to the computational burden during local training.

2.5.4 SCAFFOLD (Stochastic Controlled Averaging)

SCAFFOLD addresses *client drift*, which occurs when local models trained on non-iid data diverge from the global model. This method introduces *control variates* to align local updates with the global objective, correcting for the divergence between global and local models.

The rule to update the local update rule in SCAFFOLD is:

$$w_i^{t+1} = w_i^t - \eta \left(\nabla l_i(w_i^t) - c_i^t + c^t \right)$$

where:

- w_i^t is the local model after round t at client i,
- η is the learning rate,
- c_i^t and c^t are the control variates for the client and the global model, respectively.
- $\nabla l_i(w_i^t)$ is the local loss function gradient

The control variates c_i^t and c^t help reduce the variance caused by client drift, ensuring that local models do not vary significantly from the global model.

Advantages:

- **Mitigates Client Drift**: Ensures more consistent local and global model updates in non-iid settings.
- **Faster Convergence**: Reduces the effect of data heterogeneity on convergence speed.
- **Improved Global Model Quality**: Results in a more accurate global model, even when client data is highly heterogeneous.

Challenges:

- **Increased Communication Costs**: Additional control information must be shared between clients and the server.
- **Complexity**: Implementing SCAFFOLD requires managing control variates, adding to the computational overhead.

2.5.5 Trimmed Mean and Krum

In settings where outliers or malicious updates may occur, robust aggregation methods like *Trimmed Mean* and *Krum* are useful. These methods aim to exclude outlier updates that could degrade the global model.

Trimmed Mean removes a certain fraction of the lowest and highest updates before averaging the rest.

Krum selects the model update that is most representative of the majority by minimizing the distance to other updates.

Advantages:

- **Robustness**: These methods are resistant to outliers and malicious updates, ensuring that faulty data from some clients does not disrupt the global model.
- **Improved Stability**: More stable updates, particularly in environments with noisy or faulty sensors.

Challenges:

- **Computational Overhead**: Filtering or calculating distances between updates increases the computational cost.
- **Information Loss**: Important outliers may be excluded from aggregation, leading to potential loss of critical data.

2.5.6 Personalized Federated Learning (pFedMe)

Personalized Federated Learning (pFedMe) concentrates on creating specified models for clients, while still contributing to a global model. This is particularly useful in scenarios with high data heterogeneity, such as weather forecasting, where data from one region may differ significantly from another.

Advantages:

- **Personalization**: Adapts to diverse local data distributions, improving the performance of local models.
- **Flexibility**: Allows clients to balance global collaboration with local optimization.

Challenges:

- **Complexity**: Managing both global and local models increases computational demands.
- **Increased Resource Use**: Requires more communication and computational resources.

2.6 Weather Forecasting: An Overview

Weather forecasting is to the use of scientific principles and technology to predict atmospheric conditions at a specific location and time. Accurate weather forecasting is crucial for numerous applications, ranging from agricultural planning and disaster management to aviation and daily life. Traditionally, weather forecasting involves the collection and analysis of observational data, numerical models, and physical laws governing atmospheric dynamics.

2.6.1 Data Sources in Weather Forecasting

Weather forecasting relies on vast quantities of data, which are gathered from various sources, including:

- **Ground Stations**: Weather stations across the globe record data like temperature, humidity, wind speed, and atmospheric pressure. These measurements are often spatially limited.
- **Satellites**: Satellite data provides extensive coverage, offering critical information on cloud patterns, sea surface temperatures, and large-scale weather systems.
- **Radars**: Weather radars are essential for detecting precipitation, thunderstorms, and severe weather conditions, providing real-time data on rainfall intensity and wind velocities.
- **Weather Balloons (Radiosondes)**: These provide vertical profiles of the atmosphere, giving crucial information about temperature, pressure, and humidity at different altitudes.
- **Reanalyses Data**: These datasets combine observational data and model output to create a comprehensive picture of past atmospheric conditions.

Each of these data sources provides crucial information for creating accurate weather forecasts, but they also introduce variability due to differing spatial and temporal resolutions.

2.6.2 Numerical Weather Prediction (NWP)

NWP models form the backbone of modern weather forecasting. These models use mathematical representations of the atmosphere to simulate future weather conditions. The equations governing NWP are founded on the fundamental physical laws of thermodynamics and fluid dynamics, including:

- **The Navier-Stokes equations** for fluid motion.
- **The continuity equation** for mass conservation.
- **The thermodynamic equation** governing heat transfer.
- **The equation of state** relating pressure, temperature, and density of air.

These equations are solved iteratively on a grid that represents different parts of the atmosphere.

However, NWP models face several challenges:

- **Initial Condition Errors**: Small errors in the initial state can grow over time due to the messy nature of the atmosphere.
- **Computational Limits**: High-resolution models require vast computational resources, limiting the frequency and accuracy of predictions.
- **Data Sparsity**: Observations from remote or oceanic regions are often sparse, limiting the precision of forecasts.

2.6.3 Challenges in Weather Forecasting

- **Data Heterogeneity**: One of the significant challenges in weather forecasting is the heterogeneity of the data. Data collected from different origins, such as satellites, radars, and ground stations, vary in terms of resolution, frequency, and quality. For instance, satellite data may cover a large area but lack temporal resolution, while ground-based stations provide frequent, localized observations but lack spatial coverage. Integrating these diverse datasets is a challenge, especially when attempting to make accurate, high-resolution forecasts.
- **Chaotic Nature of the Atmosphere**: The atmosphere is a messy system, where smallish uncertainties in initial conditions can grow rapidly over time. This is often referred to as the "butterfly effect" and limits the accuracy of long-term predictions.
- **Computational Complexity**: Weather forecasting models require solving complex differential equations over large spatial grids, which demands immense computational power. High-resolution models, which are needed for local predictions, can be computationally expensive.

- **Regional Specificity**: Weather patterns differ significantly across regions, meaning that models must be tailored to local conditions, further complicating the forecasting process.

2.6.4 Federated Learning and Weather Forecasting

As data becomes increasingly diverse and distributed, weather forecasting can benefit from advanced machine learning techniques, particularly FL. Traditional forecasting relies on centralized models that require pooling data from all sources into a single location. However, this is often impractical, especially when considering privacy concerns, data ownership issues, or regional-specific data requirements.

Federated learning offers a decentralized approach, allowing different weather stations or regions to train models jointly without exchanging raw data. Each region can train a local model on its data, and then share model updates (rather than the data itself) with a central server. The server aggregates these updates to improve the global model while respecting the data heterogeneity between regions.

In the context of weather forecasting we can have:

- Data heterogeneity between regions (e.g., tropical vs. polar climates) can be managed by FL's capacity to train local models suited to the specific data distributions while still contributing to a global model.
- Computational efficiency can be improved, as local weather stations need only to transmit model updates, reducing the need for large-scale centralized computation.

CHAPTER 3 LITERATURE REVIEW

In this part, we are going to go through the state of the art in federated learning and its applications, and the state of the art in weather forecasting.

3.1 Federated Learning

Introduced by McMahan et al. [7], they introduce the key concepts of it, conduct a thorough empirical evaluation, and present a workable method for the federated learning of deep networks based on iterative model averaging, introducing the aggregation method known as FedAvg. These experiments have shown that the approach is robust to the non-IID and unbalanced data distributions. When compared to synchronized stochastic gradient descent, they demonstrate a 10–100× decrease in the number of communication rounds needed. Unlike centralized training, FL often deals with highly non-IID data; Which can lead to poor generalization or slower convergence in global models. Since clients may have limited bandwidth, it is crucial to optimize the number of model updates sent from clients to the server: Li et al. [8] have introduced methods such as FedProx to address these challenges by adding a proximal term to the optimization process, which helps improve the stability of the method. Techniques such as quantization, compression, and sparse updates have been explored to reduce communication overhead. Sattler et al. [19] propose compression methods that reduce the size of the model updates sent by clients, using sparse representations and quantizing model updates to binary values. Konečný et al. [20] demonstrate that a naïve stochastic binary rounding strategy produces a mean squared error (MSE) of $\Theta(d/n)$ for d dimensional data with n clients. While FL enhances privacy by keeping data local, it is still weak to inference attacks (e.g., model inversion). Techniques like differential privacy (DP) and secure multiparty computation (SMC) have been incorporated into FL systems to mitigate these risks. Truex et al. [21] discuss how DP and SMC can ensure that model updates do not leak sensitive information from the underlying datasets. That combination helps them reduce the growth of noise injection as the number of parties increases.

Different aggregation techniques have been proposed beyond the basic Federated Averaging (FedAvg) [7] to improve convergence rates, robustness, and accuracy in federated settings. Some algorithms generalize FedAvg by incorporating server-side optimizers like Adam, Yogi, or Adagrad into the aggregation process. Reddi et al. [22] showed how different optimizers at the server can stabilize learning, especially when client data is highly non-IID. Some methods

like MOON (Model Contrastive Federated Learning) [9] and FedAtt [23] have introduced attention mechanisms into FL aggregation to weigh clients' contributions based on relevance and data similarity. The principle of MOON is to conduct contrastive learning in model-level training by using the similarity between model representations to rectify the local training of individual parties. By dynamically assigning attention weights to client contributions, FedAtt ensures that updates from more relevant or representative clients have a larger impact on the global model. This approach enhances model performance while reducing the effect of noisy or less useful updates, making the process more communication-efficient and robust in heterogeneous data environments. SCAFFOLD [24]: This method tackles the issue of client drift in non-IID data environments by introducing control variates (or variance reduction techniques). It ensures that model updates are more aligned between clients, improving convergence when data distributions are highly heterogeneous, and moreover when clients' data are similar it leads to a faster convergence.

3.2 Applications of FL

The application of FL spans diverse domains, leveraging privacy-preserving mechanisms and addressing challenges like communication efficiency, model convergence, and data heterogeneity. In this part, we are presenting a list of applications (works).
FL has become crucial in healthcare, where patient privacy is a top concern. It allows healthcare providers to jointly train machine learning models on sensitive from patients without violating regulations. Rieke et al. [25] provide a comprehensive review of federated learning in medical image analysis and predictive diagnostics, emphasizing how FL helps in model training on decentralized data from several hospitals and research institutions. Sheller et al. [26] also demonstrate how FL was used in brain tumor classification using MRI data across several hospitals, significantly improving model generalization without exposing patients; they used FL on a U-Net [27] topology model.

FL has been widely adopted in edge computing and mobile applications, where data privacy and communication efficiency are critical. FedAvg [7], which is widely used in edge devices like smartphones for applications such as personalized predictive text models, as seen in Google's Gboard. Hard et al. [28] extended this work, showcasing how FL improves the performance of machine learning models in mobile applications, including voice recognition and smart assistants, by reducing communication overhead and enhancing personalization. In the financial sector, FL enables collaborative and secure training of machine learning models on highly sensitive data such as credit scores, transaction data, and fraud detection. Yang et al. [29] explore FL's application in anti-money laundering efforts, where banks can collab-

oratively train models for multi-party borrower detection without sharing clients' data like good clients to other parties. Wang et al. [30] propose a method that improves credit scoring while preserving client data privacy, by combining FL to knowledge transfer. Similarly, Lee et al. [31] use FL to enhance the credit risk assessment, they proved that empirically, FL helps increase the credit risk assessment of smaller financial institutions.

Zhang et al. [32] introduce a collaborative learning framework where autonomous vehicles share model updates based on local driving data, allowing global models to improve without data centralization. Pokhrel and Choi [33] discuss the ability of blockchain-based FL to enhance vehicle-to-vehicle communication, enabling real-time updates on traffic conditions, obstacle detection, and route optimization. FL has also found applications in Internet of Things networks and smart cities, where privacy and real-time data processing are paramount. Valente et al. [34] propose an FL framework for real-time traffic prediction, enabling the collaboration of decentralized edge devices such as traffic lights and sensors to optimize urban mobility. FL ensures that sensitive data, such as location or personal energy consumption, remains decentralized while contributing to the global model.

3.3 Weather Forecasting

Weather forecasting has evolved significantly, transitioning from traditional statistical methods to advanced AI techniques, especially deep learning. AI-based approaches have introduced efficiency and accuracy improvements in predicting weather patterns, capturing spatio-temporal dependencies, and handling complex datasets. Below is a comprehensive review of the progress and key approaches in AI-driven weather forecasting.

Initially, weather forecasting was dominated by traditional statistical models, which relied on methods like auto-regressive integrated moving average (ARIMA), time series analysis, and regression,. These models worked by extrapolating from historical data to make short-term predictions but often lacked the capacity to model nonlinear, complex relationships in the atmosphere. Box and Jenkins [35] became widely used for early weather prediction tasks. These statistical models worked well for short-term projections but struggled with long-term forecasts due to the inherent complexity of weather systems. With the rise of ML, more advanced models such as support vector machines (SVMs), decision trees, and random forests began to be applied to weather data. Studies like those by Dritsas et al. [36] show that random forests and SVMs improved prediction accuracy for parameters such as temperature and precipitation, leveraging large datasets and improved feature engineering. However, these machine learning models, while effective for specific tasks, lacked the ability to capture both the temporal and spatial dependencies necessary for holistic weather fore-

casting. The advent of deep learning brought a paradigm shift in weather forecasting. Models like the Convolutional LSTM (ConvLSTM) [37] and TrajGRU [38], introduced by Shi et al., became a breakthrough in precipitation nowcasting by combining the strengths of CNNs, LSTM, and GRU. ConvLSTM was able to model both spatial and temporal dependencies, making it particularly effective in radar-based precipitation forecasting. It outperformed traditional methods in short-term prediction, marking one of the first successful applications of DL in weather forecasting. More recent innovations in AI have pushed the boundaries of global weather forecasting, particularly through models like FourCastNet [4]. FourCastNet leverages Fourier Neural Operators (FNOs), which efficiently handle high-resolution global weather data by applying Fourier transforms to capture large-scale atmospheric patterns. It particularly combines Adaptative Fourier Neural Operator (AFNO) with Vision Transformer (ViT). This approach is computationally more efficient than traditional numerical weather prediction (NWP) methods, such as those used by ECMWF's Integrated Forecast System (IFS), while maintaining competitive accuracy. Similarly, EarthFormer [39] leverages attention-based transformer architectures to process multimodal data from satellites and weather stations, further improving the ability to model both temporal and spatial complexities in Earth system forecasting. RainFormer [40] combines CNN and Swin-Transformer for precipitation nowcasting, showcasing the power of attention mechanisms in handling large-scale radar data and improving forecasting accuracy for extreme weather events. Another promising direction involves the use of Graph Neural Networks (GNNs), as seen in GraphCastNet [6]. GNNs are particularly suited to weather forecasting because they can represent weather stations or geographic locations as nodes in a graph, with edges representing spatial relationships. This approach allows for better generalization across different regions and enhanced long-term forecasts by explicitly modeling geographic dependencies between data points. Hybrid models that combine AI techniques with traditional physical models have also gained traction. Pangu-Weather [5] integrates deep learning with NWP to create a fast and accurate global weather forecast model in a 3D dimension. This fusion allows the model to benefit from the physical constraints and domain knowledge embedded in NWP, while significantly reducing computational costs compared to running traditional NWP simulations alone. In addition to these cutting-edge models, foundational AI techniques continue to be explored. ConvLSTM remains a cornerstone for spatiotemporal weather modeling, with its applications extending from precipitation nowcasting to wind and temperature prediction. Further, researches like [5] explored spatiotemporal attention mechanisms to enhance the performance of LSTMs, leading to more accurate medium- and long-term forecasts in various weather scenarios. Moreover, the trend of using multimodal data is gaining momentum, as models like ClimaX [41] focus on handling diverse climate and weather datasets. ClimaX

is a foundation model that processes different data types—such as satellite images, radar data, and surface measurements—to create more robust and scalable forecasts.

High-quality datasets are crucial for training and evaluating models in AI-driven weather forecasting. One important dataset is the NCEP/NCAR Reanalysis [42] introduced in 1996, which offers global meteorological data at a resolution of 2.5°, dating back to 1948. This dataset is especially useful for long-term climate studies. The Global Precipitation Measurement (GPM) dataset [43], with half-hourly updates and a spatial resolution of 0.1°, is valuable for precipitation nowcasting and hydrological applications, providing satellite-derived global precipitation estimates. The Climate Forecast System Reanalysis (CFSR) [44] by NCEP, with a finer resolution of 0.5°, focuses on atmospheric and oceanic parameters and is utilized in medium-term weather forecasting. For model comparison, the Observations for Model Intercomparison Projects (Obs4MIPs) [45] offers datasets specifically formatted for direct comparisons with climate models, making it an essential resource for validating forecasting models. Among the most widely used is ERA5 [46], provided by ECMWF, which offers global reanalysis data at a horizontal resolution of 0.25° (25 km) and hourly time steps from 1979 to the present. It is frequently employed for deep learning applications in global weather modeling due to its high temporal and spatial resolution. Lastly, WeatherBench [47] is an essential dataset to benchmark machine learning models for weather forecasting. It is derived from ERA5 data. It offers various resolutions, from coarse 5.625° to fine 0.28125°, providing flexibility for training models with different granularities.

Even if deep learning-based state-of-the-art models offer remarkable computational efficiency, they remain heavily reliant on centralized, high-quality datasets. This dependency inherently disadvantages underrepresented or data-scarce regions, resulting in suboptimal generalization—as in Figure 17 in GraphCastNet [6], showing the regional evaluation—. These gaps highlight a critical weakness in effectively capturing diverse regional atmospheric behaviors. EarthFormer and GraphCastNet, despite excelling in modeling spatial-temporal relationships, encounter scalability challenges when deploying GNNs across expansive geographic regions. The computational overhead of processing massive graphs becomes a significant barrier, reducing their feasibility for global-scale applications. Similarly, RainFormer and other attention-based architectures require extensive computational resources to process large-scale datasets, restricting their accessibility for smaller meteorological agencies or developing regions with limited infrastructure. Hybrid approaches that integrate physics-based constraints with data-driven methodologies, such as Pangu-Weather, demonstrate improved accuracy by embedding domain knowledge. However, these models still depend on centralized, high-quality datasets, making them prone to overfitting in well-monitored climate zones while

underperforming in areas with sparse data coverage. This highlights the broader challenge of ensuring equitable model performance across heterogeneous regions.

FL provides a robust solution to these challenges by enabling decentralized model training while preserving data locality. This approach promotes more equitable model generalization by leveraging region-specific data directly at the source, which is particularly beneficial for underrepresented or data-sparse areas.

CHAPTER 4 AGGREGATION TECHNIQUE FOR FEDERATED LEARNING IN WEATHER FORECASTING

In this chapter, we thoroughly outline our approach to addressing the challenges associated with applying federated learning to weather forecasting. This section details our methodological framework, the specific assumptions that guide our design, the datasets and configurations utilized, and the tools implemented in our approach. We begin by presenting the primary assumptions and the rationale for each, establishing the foundation for the choices we make in model development, data handling, and evaluation metrics.

To address the need for a robust initial model, we assume that a centralized pre-training phase will enable better convergence and performance across decentralized clients since the starting parameters of the client models will be contextualized. We outline our decision to pre-train the model centrally on aggregated data for a set number of epochs. This allows the model to gain a comprehensive understanding of the temporal and spatial relationships within weather data before it is distributed to the clients, where local training occurs. The strategy is particularly essential for federated learning applications in meteorology, as it helps mitigate the challenges of data heterogeneity across clients and speeds up convergence, ultimately improving forecasting accuracy. We finally show the results of our investigation to see what aggregation technique is better for weather forecasting tasks, corresponding to our first research question stated in 1.3; **RQ1: What aggregation technique is most suitable for weather forecasting tasks using FL?**

4.1 Details of the solution

In this section, we outline the experimental settings used throughout this thesis. The primary goal is to explore the potential benefits of FL in weather forecasting.

4.1.1 Dataset

The dataset utilized in this research is the ERA5 reanalysis dataset, produced by the ECMWF. ERA5 is one of the most comprehensive global reanalysis datasets available, designed to provide high-quality estimates of atmospheric, oceanic, and land-surface variables by assimilating a diverse range of meteorological data sources. These sources include satellite observations, weather stations, ship and buoy measurements, and radiosondes. Through this extensive assimilation process, ERA5 reconstructs weather conditions at an hourly temporal resolution,

offering global coverage from 1979 to the present, with plans to continue updates into the future.

For the purposes of our research, ERA5 data has been re-gridded to a spatial resolution of 1.5° ($\sim$ 150 kilometers at the equator), downsampled from its native 0.25° resolution. This adjustment was necessary to accommodate computational constraints and optimize Mila's high-performance computing cluster efficiency. By reducing the spatial resolution, we were able to lower the volume of input data, thereby decreasing the computational and storage demands, as well as reducing the number of parameters of the model, all of which are critical for managing training and communication costs in a FL framework. Despite being coarser than the original ERA5 resolution, the 1.5° grid retains the essential spatial information needed to capture significant regional and global weather patterns, making it well-suited for our meteorological forecasting objectives.

The ERA5 re-gridded dataset provides a practical compromise between data granularity and computational feasibility, striking a balance that enables efficient training and model consistency across our federated learning framework. The selected resolution allows for a scalable model training process, essential in federated learning setups where data is processed across multiple distributed nodes with limited individual computational capacity. The re-gridded data supports accurate regional climate modeling and captures key meteorological phenomena, maintaining spatial integrity while allowing the model to handle expansive geographical coverage efficiently.

Key atmospheric variables included in the dataset are temperature, pressure, wind speed components, and humidity levels. These variables are crucial for accurately representing weather dynamics and are foundational inputs for models focused on tasks like precipitation prediction, temperature forecasting, and wind pattern analysis. The diverse range of ERA5 variables provides comprehensive atmospheric profiles, supporting the development of generalizable and robust models that can handle the complexity of regional and seasonal variations. Furthermore, the high temporal resolution of ERA5 enables our model to capture fine-scale temporal patterns, which is especially important for short-term forecasting in federated learning contexts.

In a nutshell, the re-gridded ERA5 dataset offers an optimized and versatile foundation for our forecasting experiments. It balances the fidelity needed for capturing meteorological patterns with computational efficiency, thereby facilitating effective implementation within a federated learning framework. This dataset supports the scalability and distributed nature of our approach, laying the groundwork for improved weather prediction models that can address the specific challenges of resource-limited and heterogeneous computational environments.

Table 4.1 shows all the variables including the surface-level and the pressure-level variables in the version of the dataset we used in the scope of this work. Note that the 13 pressure levels present in dataset are: 50 hPa, 100hPa, 150hPa, 200hPa, 250hPa, 300hPa, 400hPa, 500hPa, 600hPa, 600hPa, 700hPa, 850hPa, 925hPa, 1000hPa.

Table 4.1 List of Key Variables in the re-gridded ERA5 Dataset we use in this work. The role indicates either they are just **inputs(I)** or they are **inputs and outputs(I/O)**.

Variable	Symbol	Description	Role
Surface-Level Variables			
2m Temperature	T_{2m}	Air temperature at 2 meters above the surface	I/O
Surface Pressure	P_s	Atmospheric pressure at the surface	I
Total Precipitation	P_{total}	Accumulated precipitation over the time interval	I
10m U-wind Component	U_{10}	Zonal wind component (west-east) at 10 meters	I/O
10m V-wind Component	V_{10}	Meridional wind component (south-north) at 10 meters	I/O
Solar Radiation	SW_{down}	Downward surface solar radiation	I
Pressure-Level Variables (e.g., 500 hPa, 850 hPa, etc.)			
Geopotential Height	Φ	Geopotential height at given pressure levels	I
Temperature	T	Air temperature at given pressure levels	I
Relative Humidity	RH	Humidity relative to saturation at given levels	I
U-wind Component	U	Zonal wind (west-east direction) at given levels	I
V-wind Component	V	Meridional wind (south-north direction) at given levels	I
Specific humidity	q	Meridional wind (south-north direction) at given levels	I
Vertical velocity	w	Meridional wind (south-north direction) at given levels	I

The bold variables are the outputs on which we focus our study, these are the main target variables we use to evaluate our model, they are widely [4–6] used in weather forecasting for evaluation. These are the three variables we train our model to predict. Several other static or external variables such as:-land-sea mask, radiation at the top of the atmosphere, surface geopotential- are also provided as input context. Which means:

At 1.5° the grid points (or cells) are given by

$$G_{1.5^\circ} = \{-90.0, -88.5, -87.0, ..., 88.5, 90.0\} \times \{-178.5, -177.0, ..., 180.0\}$$

for total of $121 \times 240 = 29,040$ grid points for the surface each of the 13 vertical pressure levels. So for the output variables we have $29,040 \times 3 = 87,120$ values.

4.1.2 Distributed Structure

Since we are using the federated architecture, the idea is to split the entire globe into subregions where each region represents approximately one type of climate. So for the experiments,

we consider the separations of the globe proposed by in the ECMWF scorecards and adopted in the GraphCast paper [6]. Figure 4.1 presents the delimitation of each subregion. It is made of three main regions corresponding to the number of clients in our distributed architecture. Note that we could have gone with a more distributed one including more segmentation but due to computational limitations, we did the experiments with three clients, each for each segment of the globe. The segments are presented as follows:

- ***s.hem***: for **Southern Hemisphere** This region represents the space going from the latitude $-90°$ to $-20°$.
- ***tropics***: for **Tropics** represents here the regions where the climate is approximately the tropical one it corresponds to the latitude going from $-20°$ to $20°$.
- ***n.hem***: for **Northern Hemisphere** This region represents the space going from the latitude $20°$ to $90°$.

So for each of these regions, we will have one client (model) running, training, and sending its parameters to the server (central model).

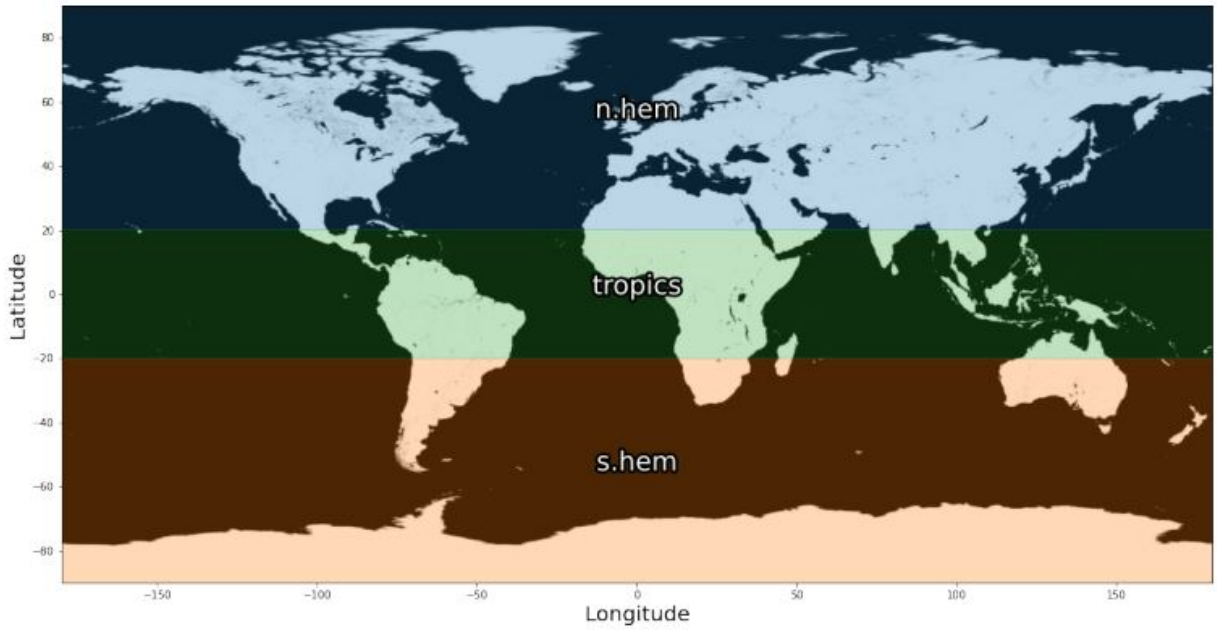

Figure 4.1 Regions delimitation. We employ the same regions and naming way as in the ECMWF scorecards `https://sites.ecmwf.int/ifs/scorecards/scorecards-47r3HRES.html`

4.1.3 Training process

The training process is separated into two steps, the first one in centralized settings and the second one in federated settings. Each with its precise objective.

1. **Pre-training**
   To effectively initialize the local models (clients) and improve convergence in our federated learning framework, we begin by training the model centrally on an aggregated dataset available publicly like ERA5(so in real-world applications we won't use the distributed data at this stage). This approach, often termed pre-training in centralized learning, allows us to establish a robust set of initial parameters that can guide the distributed learning process in subsequent federated rounds. Specifically, we train our model for 80 epochs in a centralized setting, leveraging the entire dataset to optimize the initial model parameters before deploying it across individual clients.

   During this pre-training phase, we train the model learns to predict atmospheric output variables at the next time step, $t + \Delta t$, based on the input data at a given time t. By framing the task this way, we can improve the ability of the model to capture essential temporal dependencies, which are critical for accurate weather predictions. Training centrally with global data provides a comprehensive understanding of the spatial and temporal correlations across different regions, which is particularly beneficial for complex tasks like weather forecasting where regional variations are significant.

   This centralized pre-training serves two main purposes: first, it provides a consistent starting point for all local models, reducing the impact of non-IID data across clients; and second, it accelerates the convergence of the federated learning process. Without this step, clients initialized with random weights would require far more rounds to reach optimal performance, given the heterogeneity in weather data and the distribution of clients. In this way, the centralized training phase not only improves model accuracy but also mitigates the computational burden during federated rounds by starting each local model from a strong baseline.

   After the centralized phase, the initialized model parameters are distributed to each client, allowing local updates to further refine predictions based on region-specific data. This two-step approach balances the benefits of global data exposure and local specialization, resulting in a more efficient and scalable federated learning process for weather forecasting.

2. **Fine-tuning**:
   In this stage of model training, we use the federated setting, we design our approach

with dual objectives to enhance forecasting accuracy and model robustness over extended time intervals. The first objective focuses on predicting the weather condition at time $t + \Delta t$ based on observed ground truth data at time t. This objective prioritizes the model's capacity to accurately capture and represent immediate atmospheric patterns and physical processes, aligning the predictions closely with real-world data. By optimizing predictions over this shorter interval, we aim to ensure that the model learns foundational weather patterns and dependencies that are essential for forecasting changes in conditions over brief periods.

The second objective introduces a longer-term forecasting goal by predicting the weather condition at time $t + 2\Delta t$, using the forecasted output from the model at the previous forecast time $t + \Delta t$ as input. This iterative prediction step mirrors real-world weather forecasting scenarios where models must rely on their prior forecasts to generate projections further into the future. This second step helps the model learn dependencies that accumulate over time, effectively training it to reduce error propagation over successive forecast intervals. By relying on its own forecasted outputs, the model learns to handle accumulated biases and deviations from ground truth, thus improving robustness and resilience against the compounding errors that can occur in multi-step forecasting.

Together, these dual objectives create a training process that balances the model's ability to predict immediate conditions with its capacity to project further into the future based on previous forecasts. This approach is particularly useful for applications in weather forecasting where accurate short-term predictions are necessary. Still, the model must also be resilient to compounding forecast errors when projecting further. The structure of this training strategy ultimately strengthens the model's performance across a range of forecasting intervals, providing a reliable foundation for both near-term and extended forecasting.

The federated process is run for 80 rounds of aggregation, each round representing one epoch of local training.

Let's note Z^t the global weather at the time t, we have:

$$\hat{Z}^{t+\Delta t} = f(Z^t)$$
$$\hat{Z}^{t+2\Delta t} = f(\hat{Z}^{t+\Delta t}) = f\left(f(Z^t)\right)$$

Thus, the loss function is given by:

$$\mathcal{L}\left((\hat{Z}^{t+2\Delta t}, \hat{Z}^{t+\Delta t}), (Z^{t+2\Delta t}, Z^{t+\Delta t})\right)$$

The Figure 4.2 shows a simpler explanation of the process.

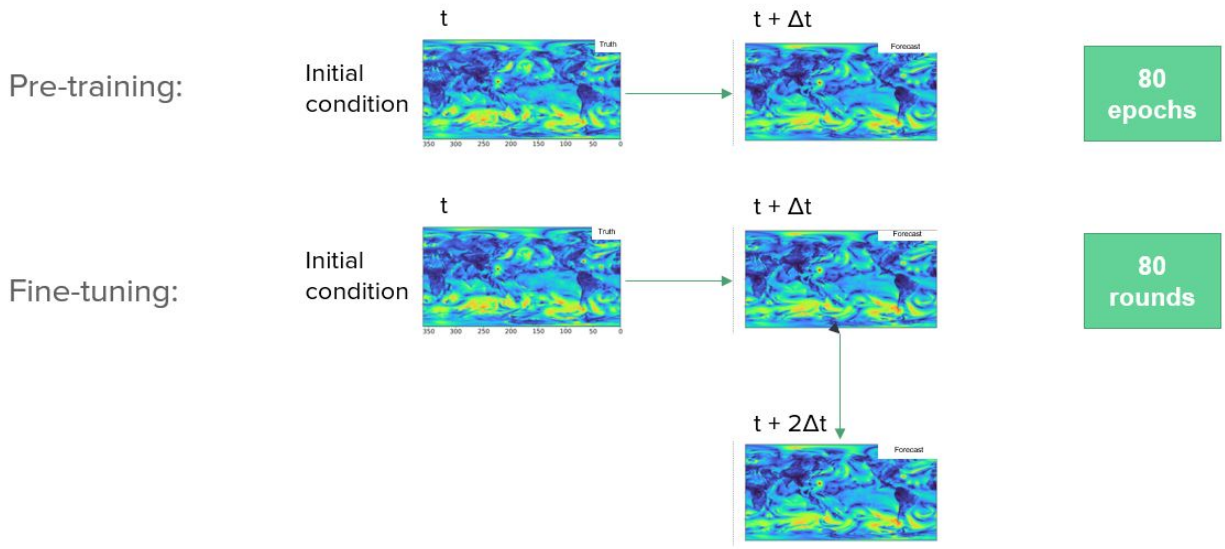

Figure 4.2 Training process of our models

4.2 Evaluation Metrics: RMSE and Anomaly Correlation Coefficient

When assessing the performance of weather forecasting models, accurate and interpretable evaluation metrics are essential for understanding both short-term and medium-term forecast quality. Here, we use **Root Mean Square Error (RMSE)** and **Anomaly Correlation Coefficient (ACC)** as key evaluation metrics, as they address distinct aspects of forecast accuracy and relevance.

4.2.1 Root Mean Square Error

It is a widely used metric in weather forecasting that provides a measure of the average magnitude of forecast error. Calculated as the square root of the average of squared differences between observed values and forecasted ones, RMSE is particularly effective for penalizing larger errors, thus prioritizing models that minimize extreme deviations. For a forecasted variable $\hat{x}_j$ and forecast time, t, latitude-weighted RMSE is computed as:

$$\mathrm{RMSE}(\hat{x}_j, t) = \sqrt{\frac{1}{NM} \sum_{g \in \mathcal{S}} L(g) \left(\hat{x}_j^t(g) - x_j^t(g) \right)^2}$$

Where g represents grid points (or cells), $g \in \mathcal{S} \subseteq G_{1.5^\circ}$ are the grid co-ordinates, M, N are respectively the number of latitudes and longitudes in $\mathcal{S}$.

$$L(g) = \frac{\cos\left(lat(g)\right)}{\frac{1}{M}\sum_{s\in\mathcal{S}}\cos\left(lat(s)\right)}$$

RMSE values are in the same units as the forecasted variable, making it an intuitive and interpretable indicator of prediction accuracy. In weather forecasting, a lower RMSE indicates closer alignment between the model and actual atmospheric conditions, signaling higher forecast precision. Given its sensitivity to large errors, RMSE is particularly useful for evaluating daily or short-term forecasts where the precise prediction of atmospheric variables, such as temperature and precipitation, is critical.

4.2.2 Anomaly Correlation Coefficient (ACC)

The ACC is another crucial metric, especially valuable for assessing long-term forecast skills. Unlike RMSE, which measures the absolute error, ACC quantifies the correlation between forecasted and observed anomalies, focusing on the ability of the model to capture the spatial and temporal patterns of deviations from the climatological mean. For a given forecasted variable $\hat{x}_j$ and forecast time, t, the ACC is computed as follows:

$$\text{ACC}(\hat{x}_j, t) = \frac{\sum_{g\in\mathcal{S}}(\hat{x}_j^t(g))(x_j^t(g))}{\sqrt{\sum_{g\in\mathcal{S}} L(g)(\hat{x}_j^t(g))^2}\sqrt{\sum_{g\in\mathcal{S}} L(g)(x_j^t(g))^2}}$$

The ACC ranges from -1 to 1, with values closer to 1 indicating a strong positive correlation, demonstrating that the model effectively captures the observed anomaly patterns. ACC is especially valuable in medium-to-long-range forecasting, where understanding the overall pattern of anomalies (such as temperature or pressure deviations) is more relevant than exact values.

Together, RMSE and ACC provide complementary perspectives on model performance: RMSE quantifies the model's absolute accuracy, while ACC assesses its skill in capturing anomaly patterns. Using both metrics allows for a comprehensive evaluation of the capability of the model to forecast across both short-term precision and medium-term anomaly tracking.

4.3 Aggregation technique for Weather Forecasting

In this part of our work, we undertake a comprehensive exploration of various aggregation techniques to assess their suitability for federated learning in the context of weather forecasting. Corresponding to **RQ1: What aggregation technique is most suitable for**

weather forecasting tasks using FL?

Recognizing the challenges unique to weather prediction—such as data heterogeneity, temporal dependencies, and regional variation—we specifically compare different methods to identify the aggregation strategy that best captures spatial and temporal nuances. We include both widely used and novel approaches, leveraging their specific features to understand how each one manages to handle the variability across different climates and meteorological patterns.

In this study, we incorporate three primary aggregation techniques: two selected based on their ability to address data heterogeneity and communication efficiency, and the standard FedAvg approach, which serves as a baseline for comparative evaluation. We apply each aggregator in federated settings (*n.hem, tropics, s.hem*) presented above, we use for this experiment the FourCastNet [4] model.

For this experiment, we focus on predicting the 2-meter temperature (T_{2m}), a critical variable for short-term and medium-term weather forecasts. The aggregation techniques are evaluated using the ACC. Figure 4.3 displays the evolution of the ACC over the rounds.

In examining the performance plot, it is clear that SCAFFOLD consistently outperforms the other aggregators. The plot shows that SCAFFOLD reaches higher ACC and converges more rapidly, making it an ideal choice for federated learning in weather forecasting. The superior performance of SCAFFOLD likely results from its use of control variates, which effectively reduce gradient variability across clients and address client drift. FedProx, while not as effective as SCAFFOLD, demonstrates reasonable stability on the plot, reflecting its improved handling of non-IID data compared to FedAvg. In contrast, FedAvg shows the least convergent performance, with its curve exhibiting slower convergence rates. The plot highlights how FedAvg struggles with data heterogeneity: as client updates are averaged without any form of correction. For the next experiments of this work, we will be using SCAFFOLD.

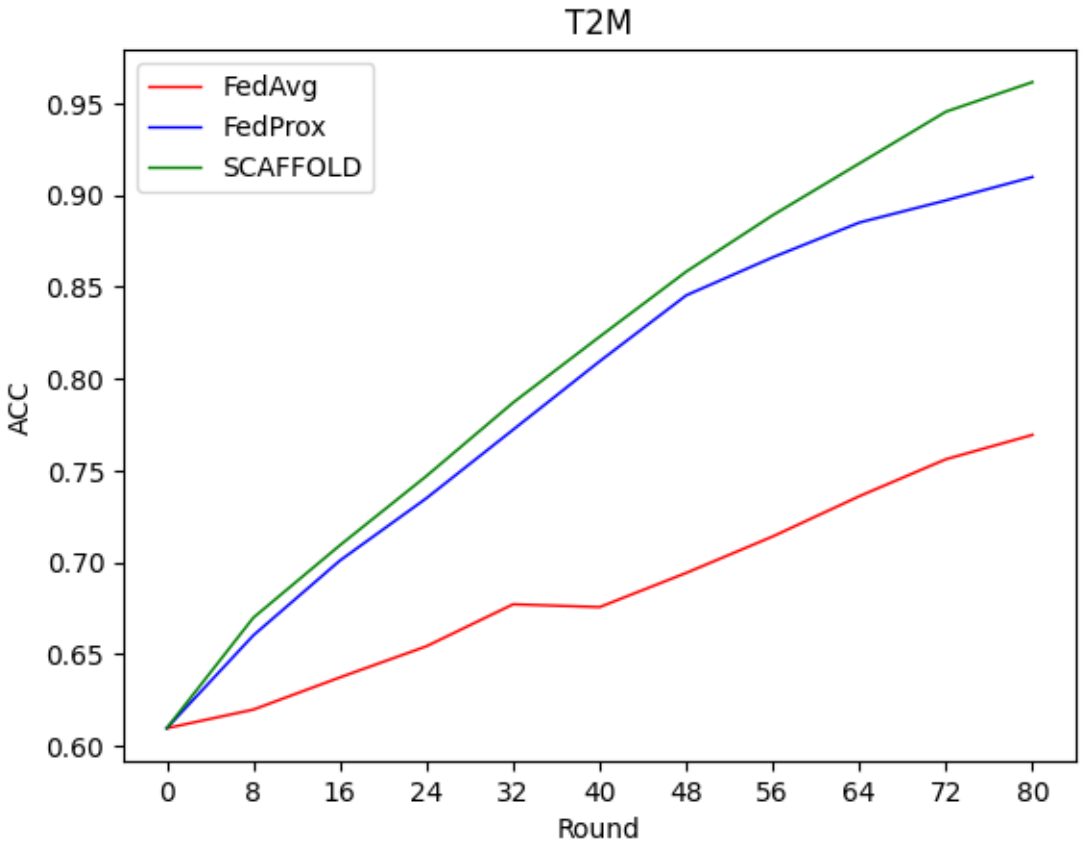

Figure 4.3 Aggregations methods comparison

CHAPTER 5 FEDERATED LEARNING FOR MORE ACCURATE AND ROBUST WEATHER FORECASTING

In this part of our work, we present the second set of results, focusing specifically on the impact of Federated Learning on model performance. Our analysis examines how the model's ACC and RMSE are influenced by the transition from a traditional centralized training approach to a federated learning framework. We will mainly analyze two widely known models: FourcastNet [4] and Pangu-Weather [5], and answer our remaining RQs(see Section 1.3).

5.1 Details of the evaluation

To evaluate the models, we train them on data from 1979 to 2017 and evaluate them on 2018 data. To calculate the RMSE and ACC we select $\mathcal{D}$ initial states, starting from these initial states we predict the following ones then the next one using the predictions, and so on over 5 days. Then we compute the ACC and RMSE of the models with the ground truth. The hyperparameters we use are enumerated in the Table 5.1.

Table 5.1 Hyperparameters

Variable	**value**
Δt	6hours
learning rate(lr)	0.005
$\|\mathcal{D}\|$	200

5.2 FL for more accurate weather predictions

RQ2: Can FL approach methodology help improve weather forecasting models' performances?

Figures 5.1 and 5.2 show the comparison between the federated and the traditional (centralized) Pangu-Weather and FourcastNet respectively. We can see that the federated approach helps reduce the RMSE and increase the ACC over time. The ACC values remain higher over time, suggesting that the federated models maintain a better correlation with actual atmospheric conditions, which is essential for accurate medium- and long-term predictions. This shows that federated learning helps enhance the model's in-distribution generalization.

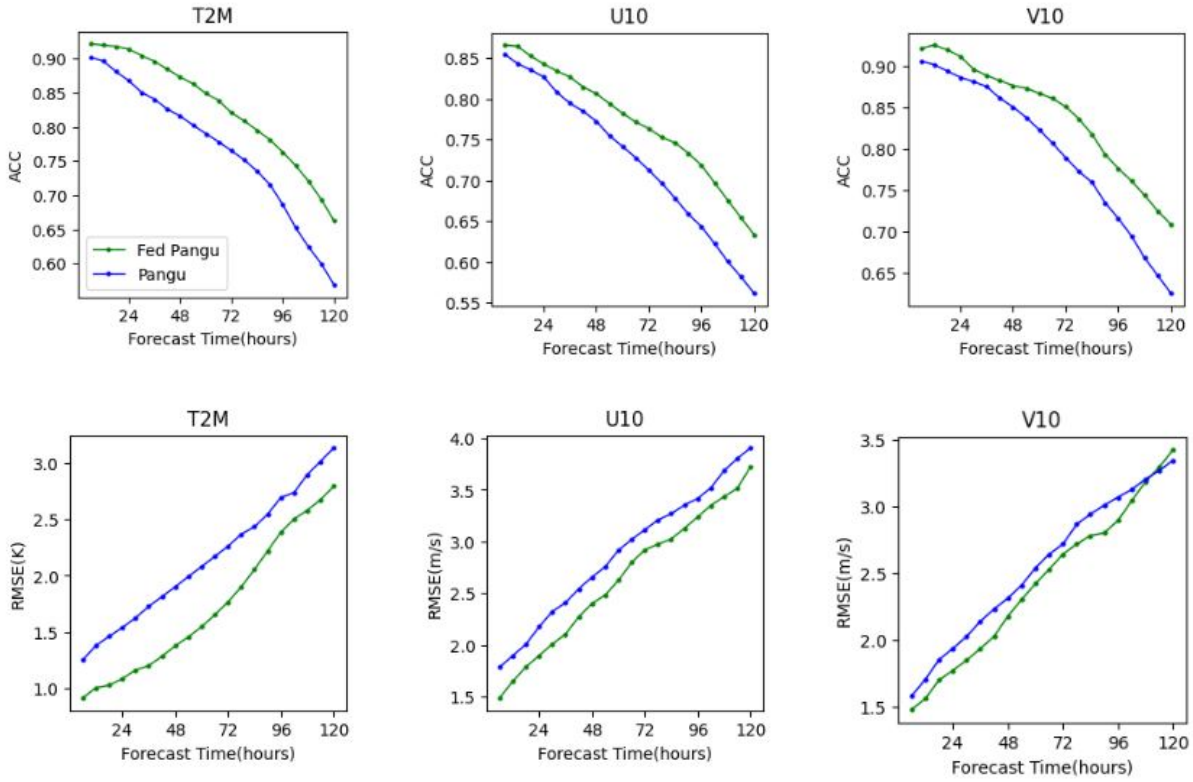

Figure 5.1 Comparison of Federated Pangu-Weather and Pangu-Weather

5.3 Federated Learning for Robustness in Weather Forecasting

RQ3: Can FL help improve ML-based weather forecasting models' robustness against climate change?

In this part, we evaluate the behavior of the models against climate change. We would like to see how our model behaves when it faces changes out of the distribution of the data or the weather. To do so we will switch positions of 30% of our testing set with respect to reality, the rules we apply are as follows:

- we pick an equal number of datapoints' pairs to switch from each of our three regions presented above.
- We make sure that in the pair each data point represents two consecutive seasons in the region's climate. For example, in the northern hemisphere, one data point is in the spring and the other one is in the summer, or one in the fall and the other one in the winter.

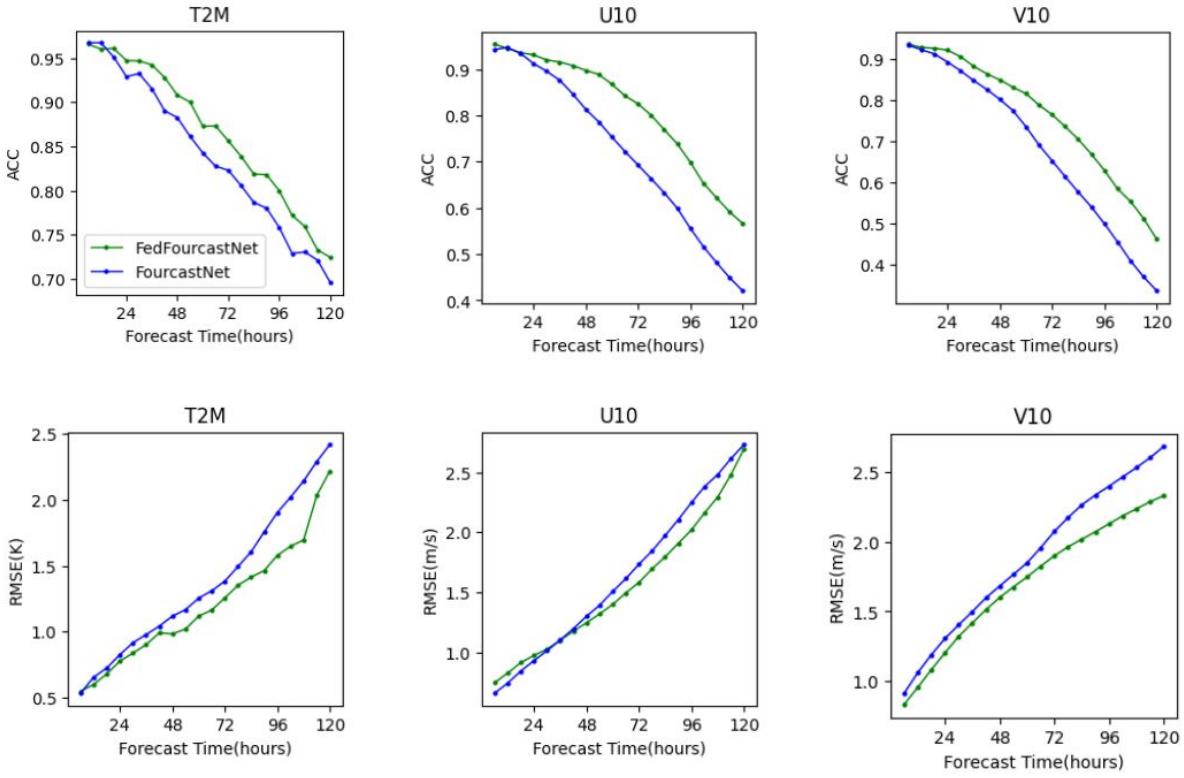

Figure 5.2 Comparison of Federated FourcastNet and FourcastNet

- The two data points represent the same grid i.e., similar latitude and longitude.

Figures 5.3 and 5.4 show the comparison of the ACC and RMSE overtime of forecasting respectively for Pangu-weather and FourcastNet in poisoned data settings. We can see that, it affects the models' performances by reducing the ACC, despite that the federated model still achieved higher ACC and lower RMSE showing that it better adapts to sudden weather changes.

5.4 Federated Learning for regional improved weather forecast

RQ4: Can FL help improve weather forecasting accuracy in low-resource regions of the globe?

In this part of our study, we focus on evaluating models on each of the three specific regions to see the impact and the change FL can bring to the model compared to the traditional centralized model. For this part, we sample the same number $\|\mathcal{D}\|$ of initial weather in each region (*s.hem, tropics, n.hem*). The results are presented as follows:

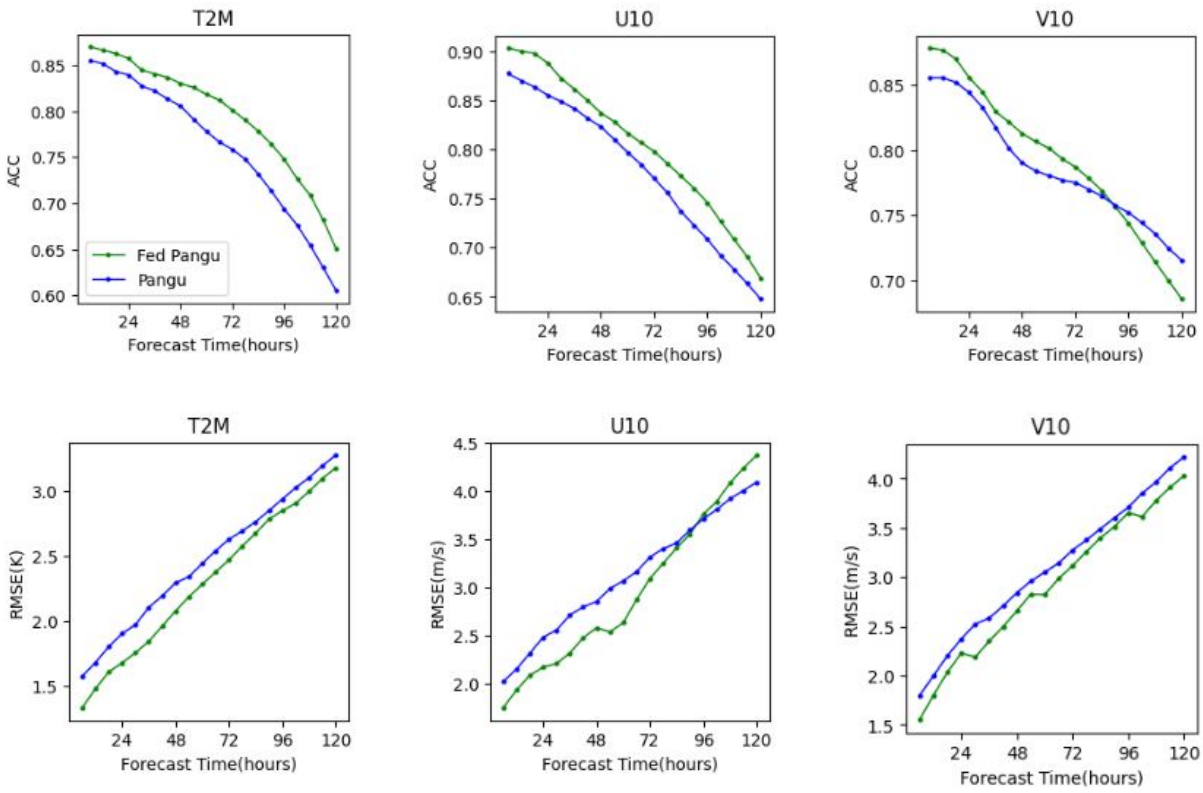

Figure 5.3 Comparison of Federated Pangu-Weather and Pangu-Weather against outliers aka change in weather

- ***s.hem*** Figures 5.5 and 5.6 show the comparison plot of both approaches in the southern hemisphere.
- ***tropics*** Figures 5.9 and 5.10 show the comparison plot of both approaches in the Tropics.
- ***n.hem*** Figures 5.9 and 5.10 show the comparison plot of both approaches in the northern hemisphere.

We can see that in the *s.hem* and *n.hem* the federated approach does not help much as the difference is just slightly in favor of the federated models. On the other hand, we notice that the federated approach helps a lot in *tropics* which is the region known as the "underrepresented" regions of the world in terms of data, the ACC and the RMSE are significantly improved by the federated approach and we can even see that with time it helps the model keep being consistent.

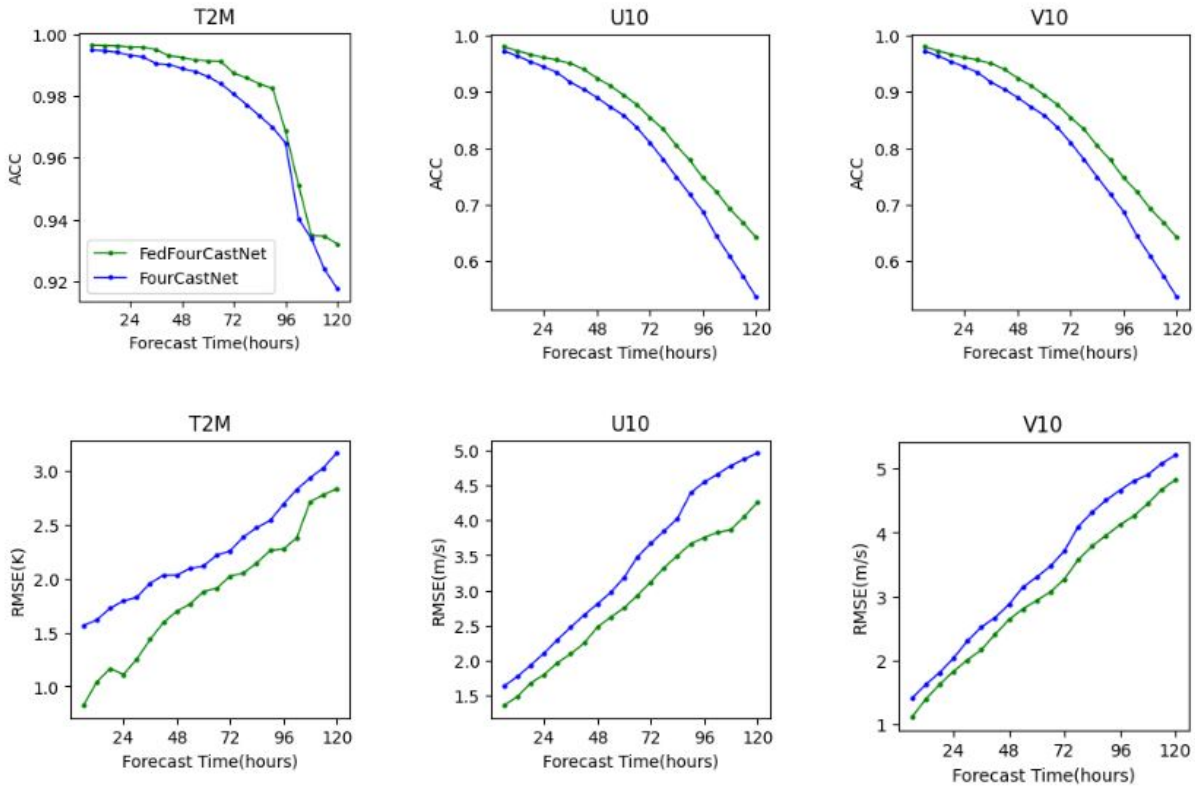

Figure 5.4 Comparison of Federated FourcastNet and FourcastNet against outliers aka change in weather

5.5 Discussion and Limitations

While this research presents useful insights into the application of federated learning for weather forecasting, several potential threats to validity should be acknowledged to ensure the credibility and reliability of the findings.

5.5.1 Internal Validity

Variability in client participation rates, data quality, the number of splitter regions, and computational capabilities among decentralized nodes may significantly impact the performance of federated models, regardless of the aggregation method used. Additionally, factors such as randomness in model initialization or data sampling can introduce unintended bias, potentially obscuring the true effectiveness of the aggregation techniques under evaluation.

To mitigate these confounding variables, we employed uniform pre-processing protocols across all clients to ensure consistent data handling. Moreover, all clients were included in every

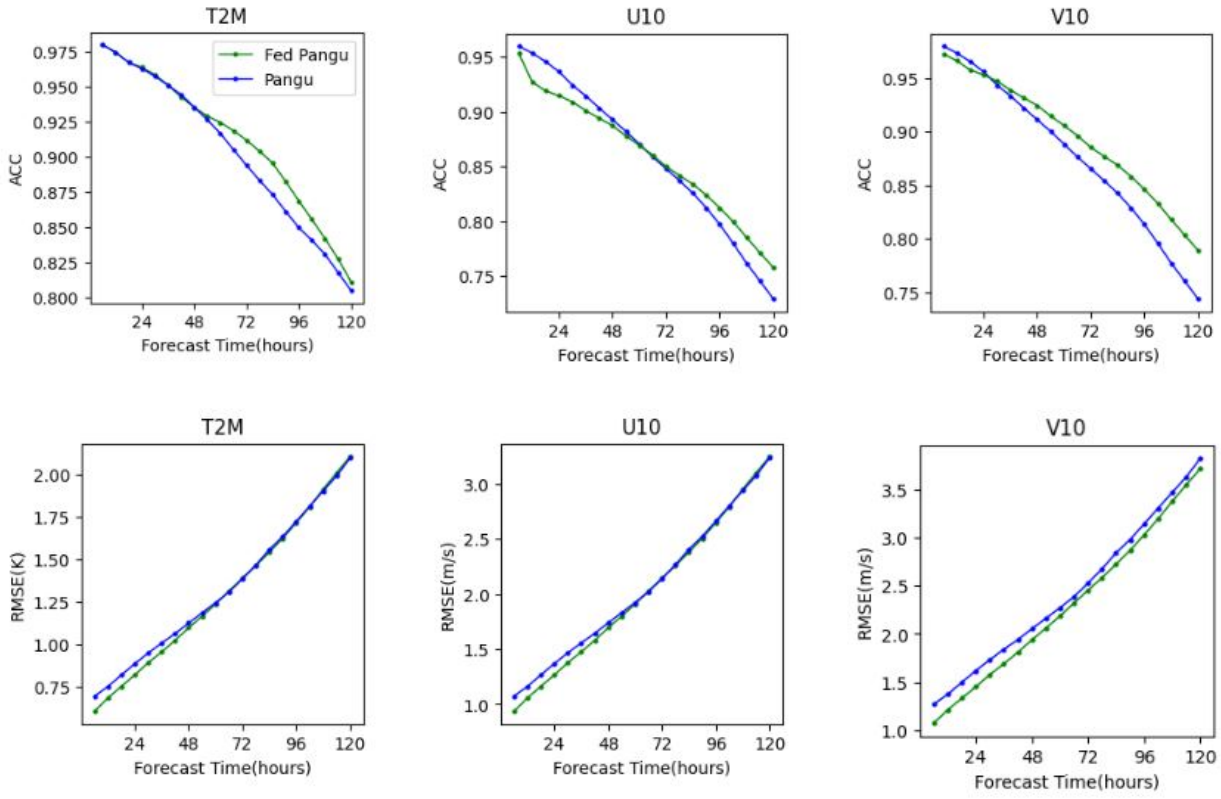

Figure 5.5 Comparison of Federated Pangu-Weather and Pangu-Weather in the ***s.hem*** regions

aggregation round to eliminate variability due to client participation rates. To further ensure robustness and reliability, the experiments comparing aggregation techniques—specifically analyzing RMSE and ACC—were repeated in a second independent experimental setup. This validation step confirmed the consistency of the results and reduced the influence of stochastic factors, providing stronger evidence for the observed performance trends across the aggregation methods.

5.5.2 External Validity

Models trained on ERA5 data might face challenges in generalizing to other datasets or regions with distinct atmospheric conditions, especially those characterized by sparse or noisy data. While the experimental setup was designed to be representative, it may not fully account for the diverse operational challenges encountered by meteorological agencies worldwide, such as real-time data variability and infrastructural constraints. To improve generalizability, the models were rigorously evaluated across three distinct climatic zones—Northern

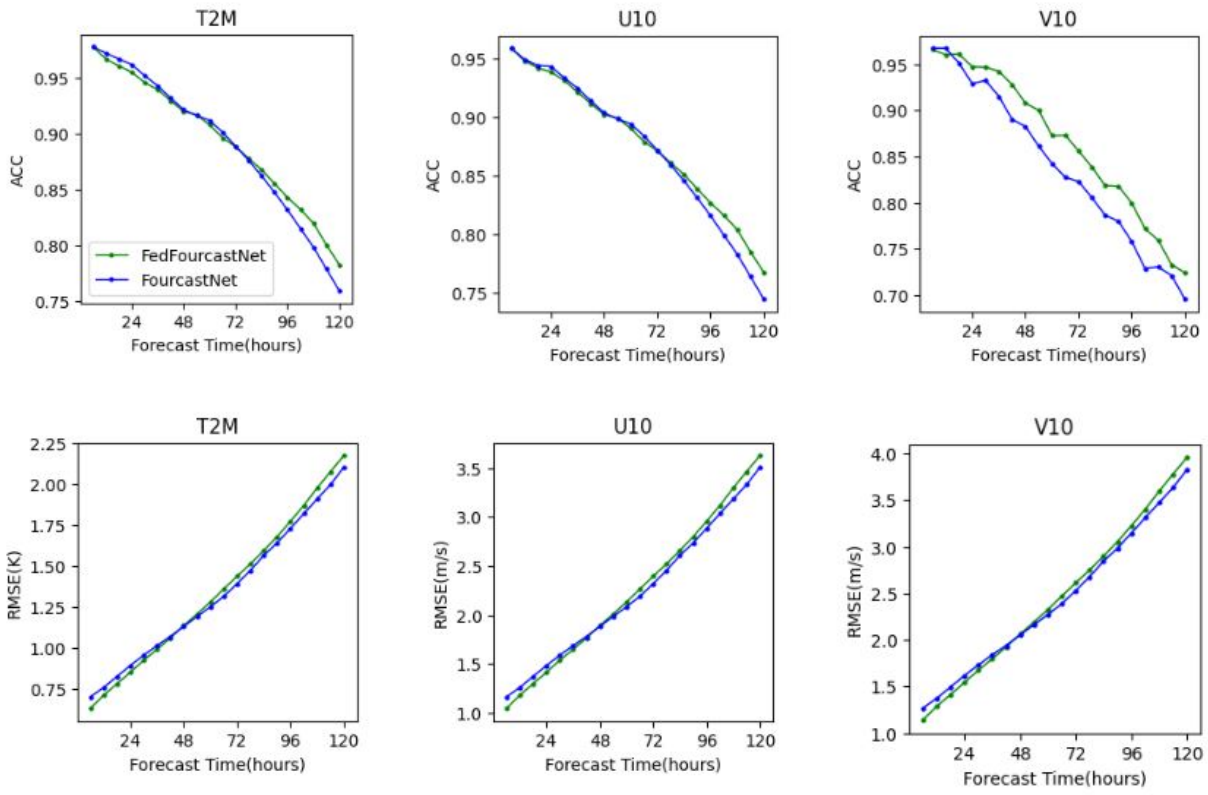

Figure 5.6 Comparison of Federated FourcastNet and FourcastNet in the ***s.hem*** region

Hemisphere, Tropics, and Southern Hemisphere—each with unique atmospheric dynamics. This approach ensures a broader assessment of the model's performance. Furthermore, future work will focus on extending validation efforts to include additional datasets, such as WeatherBench and GPM, which encompass diverse data sources and resolutions. This step seeks to enhance the model's adaptability to various climatic conditions and operational scenarios, ensuring wider applicability and robustness in global weather forecasting systems.

Deep learning models inherently differ in architecture, training dynamics, and data requirements, which can limit the reproducibility of findings across diverse models. To mitigate this, our study incorporates two distinct models to evaluate the robustness and consistency of results under different architectures and approaches. This dual-model strategy reduces the risk of overfitting conclusions to a single architecture, enhancing the validity of our findings. Future research can build upon this foundation by exploring additional models, particularly those leveraging emerging paradigms such as transformer-based architectures or hybrid physics-informed neural networks. By broadening the scope of tested models, subsequent work can further validate the generalizability of the techniques presented and adapt them to

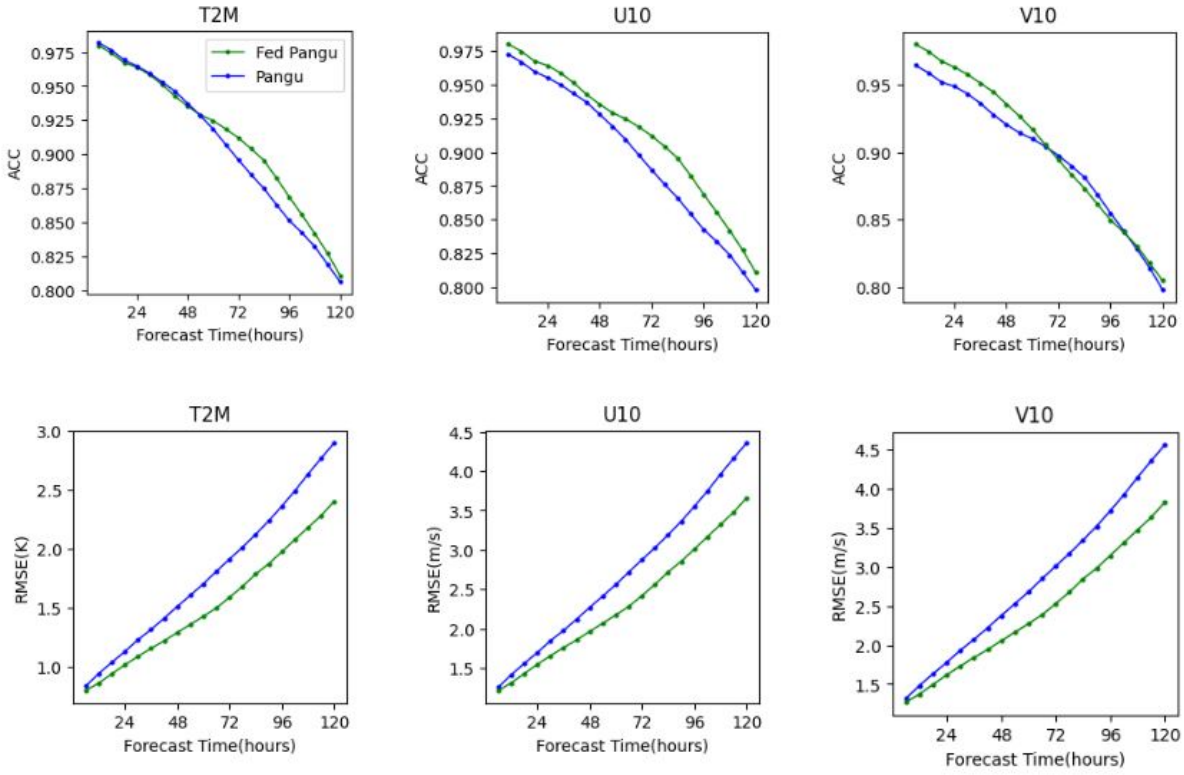

Figure 5.7 Comparison of Federated Pangu-Weather and Pangu-Weather in the ***tropics*** regions

evolving advancements in the field of weather forecasting.

5.5.3 Construct Validity

The reliance on metrics such as RMSE and ACC, while widely used, may not fully capture the model's capability to predict rare or less recurrent events, such as those associated with climate change, as defined in this thesis. These metrics are limited in their ability to evaluate the model's robustness against anomalies or sudden shifts in atmospheric patterns. Moreover, the method employed for data poisoning, though deliberate, may not fully replicate the intricacies of real-world data inconsistencies or adversarial perturbations.

To address these concerns, we adopted a data poisoning strategy emphasizing consistency, such as swapping seasonal data points (e.g., interchanging spring and fall data) or switching data from mid-summer with early fall observations. This approach ensures that perturbations reflect plausible, albeit erroneous, atmospheric scenarios, testing the model's resilience in a realistic manner.

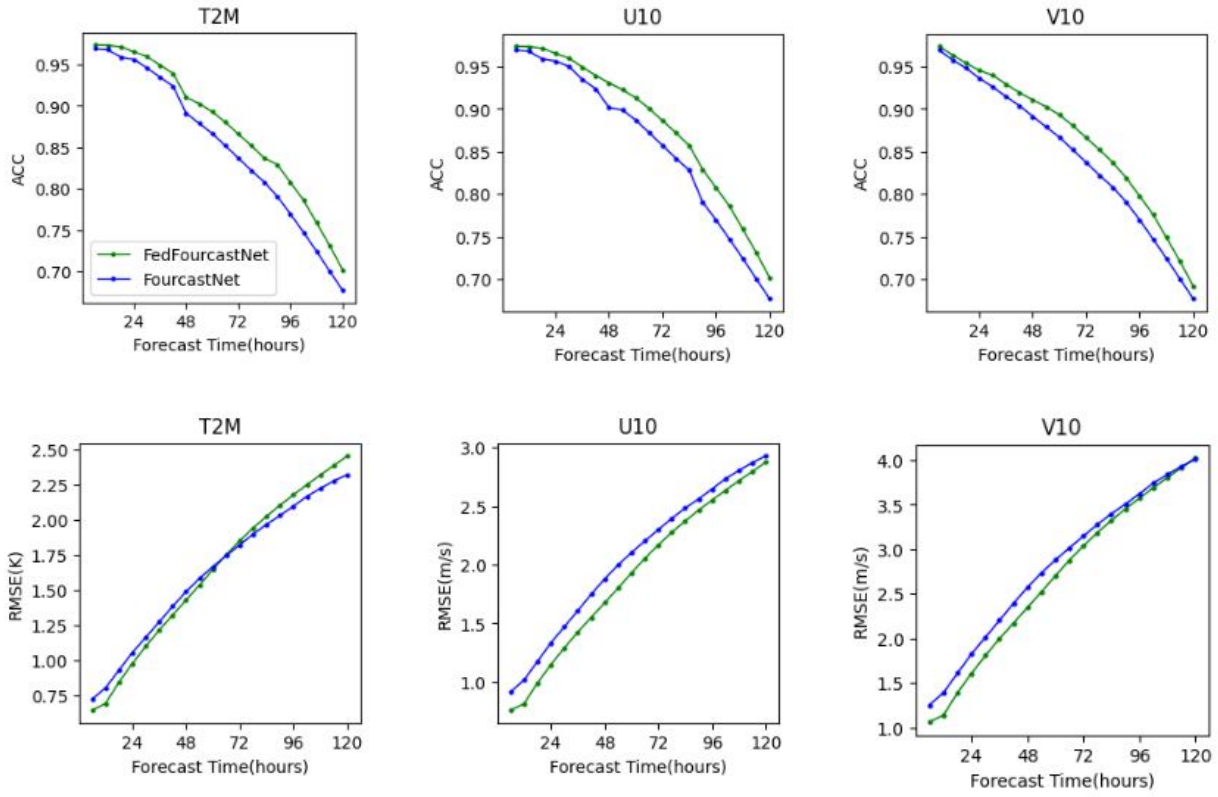

Figure 5.8 Comparison of Federated FourcastNet and FourcastNet in the ***tropics*** region

However, these efforts still operate under simplified experimental conditions that may not adequately represent the complexities of operational weather systems. Future work will explore more advanced evaluation methods, such as spatial anomaly detection, to better assess the ability of the model to capture nuanced irregularities and extremes in atmospheric behavior. By extending these measures, we aim to provide a more comprehensive assessment of model performance and its potential for real-world deployment.

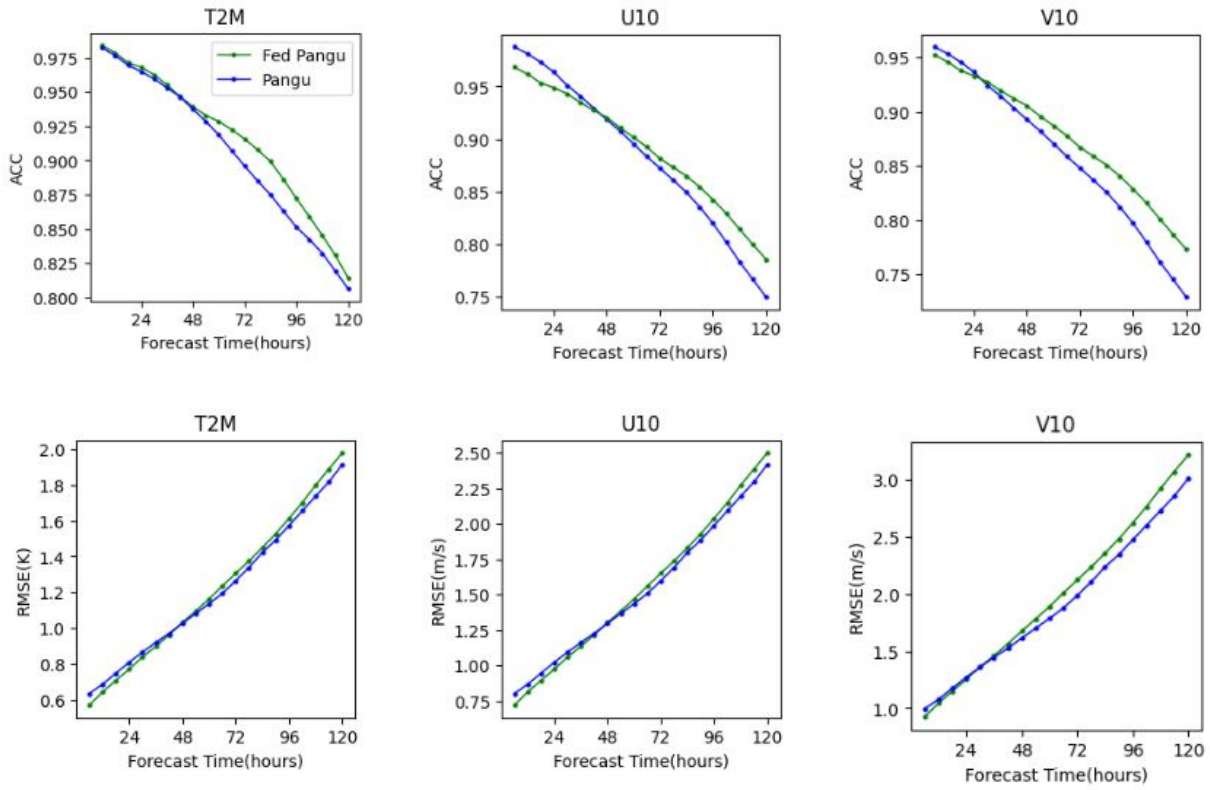

Figure 5.9 Comparison of Federated Pangu-Weather and Pangu-Weather in the ***n.hem*** regions

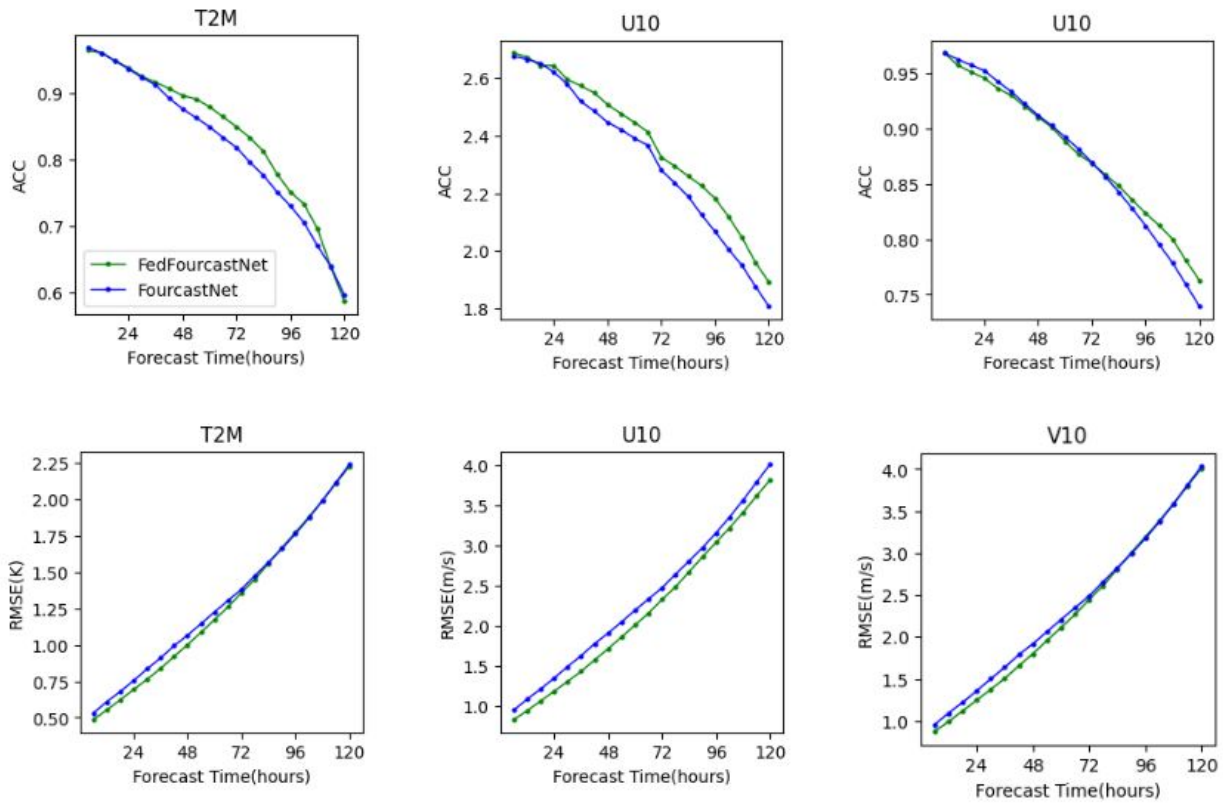

Figure 5.10 Comparison of Federated FourcastNet and FourcastNet in the ***n.hem*** region

CHAPTER 6 CONCLUSION

Our primary contribution in this thesis is demonstrating that the federated approach, commonly known as federated learning, can enhance generalization in weather forecasting. While our study focuses on three output variables and two well-known models, this methodology is adaptable and can be extended to other models and additional atmospheric variables.

6.1 Summary of Works

In this research, we have explored the potential of federated learning to improve generalization in weather forecasting by addressing the challenges posed by decentralized, heterogeneous datasets collected across diverse climatic regions. Building on a comprehensive review of traditional and machine-learning-based methods for weather prediction, this thesis highlights the limitations of centralized data aggregation when dealing with non-IID data, a challenge that is particularly significant in the context of spatially distributed meteorological information. Through an in-depth investigation into aggregation techniques within federated learning, we focused on methods such as FedAvg, FedProx, and SCAFFOLD analyzing their effectiveness in harmonizing predictions across regions with distinct weather patterns.

The pre-training phase enabled an effective initialization of our federated model, providing a shared basis for each client while accommodating regional data variability. Our comparative analysis demonstrated that aggregation methods designed to counter client drift, such as SCAFFOLD, significantly improved model accuracy and convergence speed in federated settings with non-IID data. By applying these approaches across geographically and climatically diverse regions —the Northern Hemisphere, tropics, and Southern Hemisphere— we observed that the optimized aggregation techniques notably enhanced the predictive capacity of surface variables such as 2-meter temperature, 10m U-wind component, and 10m V-wind component, reducing forecasting error while increasing resilience over successive forecasting intervals. We also show that a model generalizes better out-of-distribution and in-distribution by using federated learning.

Overall, this study contributes to bridging the gap between federated learning paradigms and real-world applications in environmental science, highlighting the adaptability of federated learning in meteorological forecasting. The outcomes underscore the importance of tailored aggregation strategies in federated settings and offer a pathway for future work to further adapt these methods for a broader range of atmospheric variables and extended multi-modal

data sources, ultimately aiming to enhance the accuracy and accessibility of weather forecasts across regions.

6.2 Limitations

While this research advances the use of federated learning for weather forecasting, certain limitations present both challenges and opportunities for improvement. Firstly, federated learning relies on heterogeneous, decentralized data, which complicates model training, as regions often vary significantly in data distribution, density, and feature characteristics. This heterogeneity necessitates complex, customized aggregation methods to manage client drift and ensure consistent performance across regions. Although methods like SCAFFOLD mitigate some of these issues, aggregation methods do not include physics concepts that could help better aggregate regional models.

Another limitation stems from computational constraints. Running federated learning experiments, particularly with larger models and smaller resolutions or on limited computational resources, can be resource-intensive and slow. This issue was partially addressed through model pre-training and reduced spatial resolution, yet further optimization and resource allocation are needed for broader scalability. Additionally, while our approach effectively predicts short-term surface variables extending it to other critical variables or forecasting longer-term intervals may require refinements in both the model architecture and the aggregation approach to maintain predictive accuracy.

6.3 Future Research

Building on the findings of this research, future work could pursue several directions to further enhance the use of federated learning for weather forecasting. Firstly, developing more sophisticated aggregation techniques tailored for non-IID and imbalanced datasets remains a priority in this particular context, i.e. developing aggregation techniques taking into consideration the physics nature of weather. Emerging approaches, such as personalized federated learning, which adapts the global model to each client's specific data distribution, could offer a more accurate, locally optimized solution in cases of extreme data heterogeneity. Additionally, integrating adaptive learning rates or fine-tuning parameters based on client characteristics may further improve model convergence and accuracy in complex environmental applications.

Another avenue for future research lies in the expansion of the model's predictive capabilities to cover a wider range of meteorological variables and extended time horizons. Addressing

multi-modal data fusion within the federated framework, such as combining satellite, radar, and ground-based measurements, could offer a more comprehensive view of atmospheric dynamics.

To ensure more robust generalization, access to high-resolution datasets and real-time data, particularly for underrepresented regions, would also enhance the model's reliability and extend its global applicability.

We could also try a distributed architecture with more regions as the one proposed in the ECMWF scorecards including about 17 sub-regions.

REFERENCES

[1] N. Jarecki, "Shallow vs deep neural network - neural network ai, hd png download - kindpng." [Online]. Available: https://www.kindpng.com/imgv/TTmhioT_shallow-vs-deep-neural-network-neural-network-ai/

[2] Á. Peris, M. Domingo, and F. Casacuberta, "Interactive neural machine translation," *Comput. Speech Lang.*, vol. 45, pp. 201–220, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:11613558

[3] apmonitor, "Dynamic optimization." [Online]. Available: https://apmonitor.com/do/index.php/Main/LSTMNetwork

[4] J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli, P. Hassanzadeh, K. Kashinath, and A. Anandkumar, "Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators," 2022. [Online]. Available: https://arxiv.org/abs/2202.11214

[5] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian, "Pangu-weather: A 3d high-resolution model for fast and accurate global weather forecast," 2022. [Online]. Available: https://arxiv.org/abs/2211.02556

[6] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, W. Hu, A. Merose, S. Hoyer, G. Holland, O. Vinyals, J. Stott, A. Pritzel, S. Mohamed, and P. Battaglia, "Graphcast: Learning skillful medium-range global weather forecasting," 2023. [Online]. Available: https://arxiv.org/abs/2212.12794

[7] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," 2023. [Online]. Available: https://arxiv.org/abs/1602.05629

[8] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2020. [Online]. Available: https://arxiv.org/abs/1812.06127

[9] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," 2021. [Online]. Available: https://arxiv.org/abs/2103.16257

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[11] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10. Madison, WI, USA: Omnipress, 2010, p. 807–814.

[12] X. Garcia-Santiago, S. Burger, C. Rockstuhl, and P.-I. Schneider, "Bayesian optimization with improved scalability and derivative information for efficient design of nanophotonic structures," *Journal of Lightwave Technology*, vol. 39, no. 1, p. 167–177, Jan 2021. [Online]. Available: http://dx.doi.org/10.1109/JLT.2020.3023450

[13] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," 2020.

[14] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990. [Online]. Available: https://www.sciencedirect.com/science/article/pii/036402139090002E

[15] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[16] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[18] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," 2013.

[19] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Sparse binary compression: Towards distributed deep learning with minimal communication," 2018. [Online]. Available: https://arxiv.org/abs/1805.08768

[20] J. Konečný, "Stochastic, distributed and federated optimization for machine learning," 2017. [Online]. Available: https://arxiv.org/abs/1707.01155

[21] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A hybrid approach to privacy-preserving federated learning," 2019. [Online]. Available: https://arxiv.org/abs/1812.03224

[22] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," 2021. [Online]. Available: https://arxiv.org/abs/2003.00295

[23] S. Ji, S. Pan, G. Long, X. Li, J. Jiang, and Z. Huang, "Learning private neural language modeling with attentive aggregation," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Jul. 2019. [Online]. Available: http://dx.doi.org/10.1109/IJCNN.2019.8852464

[24] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," 2021. [Online]. Available: https://arxiv.org/abs/1910.06378

[25] N. Rieke, J. Hancox, W. Li, F. Milletarì, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, S. Ourselin, M. Sheller, R. M. Summers, A. Trask, D. Xu, M. Baust, and M. J. Cardoso, "The future of digital health with federated learning," *npj Digital Medicine*, vol. 3, no. 1, Sep. 2020. [Online]. Available: http://dx.doi.org/10.1038/s41746-020-00323-1

[26] M. Sheller, B. Edwards, G. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. Colen, and S. Bakas, "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data," *Scientific Reports*, vol. 10, 07 2020.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[28] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," 2019. [Online]. Available: https://arxiv.org/abs/1811.03604

[29] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," 2019. [Online]. Available: https://arxiv.org/abs/1902.04885

[30] Z. Wang, J. Xiao, L. Wang, and J. Yao, "A novel federated learning approach with knowledge transfer for credit scoring," *Decis. Support Syst.*, vol. 177, no. C, Mar. 2024. [Online]. Available: https://doi.org/10.1016/j.dss.2023.114084

[31] C. M. Lee, J. Delgado Fernandez, S. Potenciano Menci, A. Rieger, and G. Fridgen, "Federated learning for credit risk assessment," 01 2023.

[32] H. Zhang, J. Bosch, and H. H. Olsson, "End-to-end federated learning for autonomous driving vehicles," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.

[33] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4734–4746, 2020.

[34] R. Valente, C. Senna, P. Rito, and S. Sargento, "Federated learning framework to decentralize mobility forecasting in smart cities," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, 2023, pp. 1–5.

[35] G. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. Holden-Day, 1976.

[36] E. Dritsas, M. Trigka, and P. Mylonas, "A multi-class classification approach for weather forecasting with machine learning techniques," in *2022 17th International Workshop on Semantic and Social Media Adaptation Personalization (SMAP)*, 2022, pp. 1–5.

[37] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. kin Wong, and W. chun Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," 2015. [Online]. Available: https://arxiv.org/abs/1506.04214

[38] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W. kin Wong, and W. chun Woo, "Deep learning for precipitation nowcasting: A benchmark and a new model," 2017. [Online]. Available: https://arxiv.org/abs/1706.03458

[39] Z. Gao, X. Shi, H. Wang, Y. Zhu, Y. Wang, M. Li, and D.-Y. Yeung, "Earthformer: Exploring space-time transformers for earth system forecasting," 2023. [Online]. Available: https://arxiv.org/abs/2207.05833

[40] C. Bai, F. Sun, J. Zhang, Y. Song, and S. Chen, "Rainformer: Features extraction balanced network for radar-based precipitation nowcasting," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 01 2022.

[41] T. Nguyen, J. Brandstetter, A. Kapoor, J. K. Gupta, and A. Grover, "Climax: A foundation model for weather and climate," 2023. [Online]. Available: https://arxiv.org/abs/2301.10343

[42] E. Kalnay, M. KANAMITSU, R. KISTLER, W. Collins, D. DEAVEN, G. LS, M. IREDELL, S. Saha, G. White, J. WOOLLEN, Y. Zhu, M. Chelliah, W. EBISUZAKI,

W. Higgins, J. Janowiak, K. C, C. ROPELEWSKI, J. Wang, and A. LEETMAA, "The nmc/ncar 40-year reanalysis project. bull am meteorol soc," *Bulletin of the American Meteorological Society.*, vol. 77, 03 1996.

[43] G. Huffman, D. Bolvin, D. Braithwaite, K. Hsu, R. Joyce, C. Kidd, E. Nelkin, S. Sorooshian, E. Stocker, J. Tan, D. Wolff, and P. Xie, *Integrated Multi-satellite Retrievals for the Global Precipitation Measurement (GPM) Mission (IMERG)*, 04 2020, pp. 343–353.

[44] S. Saha, S. Moorthi, H.-L. Pan, X. Wu, J. Wang, S. Nadiga, P. Tripp, R. Kistler, J. Woollen, D. Behringer, H. Liu, D. Stokes, R. Grumbine, G. Gayno, J. Wang, Y.-T. Hou, H.-Y. Chuang, H.-M. Juang, J. Sela, and M. Goldberg, "The ncep climate forecast system reanalysis," *Bulletin of The American Meteorological Society - BULL AMER METEOROL SOC*, vol. 91, 08 2010.

[45] D. Waliser, P. Gleckler, R. Ferraro, K. Taylor, S. Ames, J. Biard, M. Bosilovich, O. Brown, H. Chepfer, L. Cinquini, P. Durack, V. Eyring, P.-P. Mathieu, T. Lee, S. Pinnock, G. Potter, M. Rixen, R. Saunders, J. Schulz, and M. Tuma, "Observations for model intercomparison project (obs4mips): status for cmip6," *Geoscientific Model Development*, vol. 13, pp. 2945–2958, 07 2020.

[46] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, A. Simmons, C. Soci, S. Abdalla, X. Abellan, G. Balsamo, P. Bechtold, G. Biavati, J. Bidlot, M. Bonavita, and J.-N. Thépaut, "The era5 global reanalysis," *Quarterly Journal of the Royal Meteorological Society*, vol. 146, 06 2020.

[47] S. Rasp, P. D. Dueben, S. Scher, J. A. Weyn, S. Mouatadid, and N. Thuerey, "Weatherbench: A benchmark data set for data-driven weather forecasting," *Journal of Advances in Modeling Earth Systems*, vol. 12, no. 11, Nov. 2020. [Online]. Available: http://dx.doi.org/10.1029/2020MS002203