Exploring Machine Learning Approaches to Precipitation Prediction: Post Processing of Daily Accumulated North American forecasts

by

Rushil Goomer

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

in the Department of Applied Computer Science

Rushil Goomer, 2023
University of Winnipeg

Exploring Machine Learning Approaches to Precipitation Prediction: Post Processing of Daily Accumulated North American forecasts

by

Rushil Goomer

Supervisory Committee

Dr. S. Ramanna, Supervisor
(Department of Applied Computer Science)

Dr. C. Valdemerra, Member
(Department of Applied Computer Science)

Dr. K. Kotecha, External Member
(Symbiosis Centre for Applied Artificial Intelligence (SCAAI), Symbiosis International, Pune, India)

ABSTRACT

This thesis presents recent work on exploring machine learning (ML) and deep learning (DL) models to improve the accuracy of 24 hour precipitation forecasts. Leveraging a comprehensive North American dataset of precipitation values from Numerical Weather Prediction (NWP) models and secondary meteorological features, the research showcases the need of ML techniques in post-processing NWP precipitation predictions. The evaluation reveals remarkable performance improvements over baseline model, with certain ML models achieving a 15% reduction in Mean Absolute Error (MAE), a 5% decrease in Root Mean Squared Error (RMSE), a 45% reduction in Median Absolute Error (MdAE), and a 50% decrease in Relative Bias (RB). Convolutional Neural Networks (CNN) and Gradient Boosting Regressor (GBR) emerged as top performers, demonstrating their proficiency in accurately predicting daily precipitation.

Keywords: Numerical Weather Prediction (NWP), Precipitation Forecasting, Machine Learning, Neural Networks, Gradient Boosting, Graph Neural Networks, Weather Forecast, Post-processing, Meteorological Features.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

As I wrap up this research endeavor, I want to express my profound appreciation to Dr. Sheela Ramanna and Scott Kehler for their unwavering support and guidance from the beginning to the very end. Without them, none of this would have been achievable. I am immensely thankful!

My academic journey at the University of Winnipeg has been exceptionally enriching and satisfying, thanks to various entities, administrators, instructors, colleagues, and friends. I wish to extend my gratitude to the Natural Sciences and Engineering Research Council of Canada Alliance Grant with Weatherlogics Inc. for their substantial financial backing and the opportunities they provided. I also want to express my appreciation to my former professors, Prof. Simon Liao and Prof. Christopher Henry, as well as all the esteemed members of the Applied Computer Science Department who played a pivotal role in my academic success. Special thanks are due to Cenker Sengoz and Paul Pries for their valuable contributions to this research. I'd like to acknowledge the team at Weatherlogics Inc., a technology company that transforms weather data into actionable information for businesses. Additionally, I want to thank the examining committee, including Dr. Ramanna, Dr. Valdemerra and Dr. Kotecha, for their time and dedication in evaluating this work.

Finally, my heartfelt thanks go to my parents, family and friends for standing by me through all the ups and downs.

Rushil Goomer

DEDICATION

To God and my awesome Parents...

Chapter 1

Introduction

Numerical Weather Prediction (NWP) refers to using mathematical models to process weather data to make forecasts [3]. It is a cornerstone of modern meteorology, playing a pivotal role in providing forecasts that influence a myriad of sectors, from agriculture and transportation to emergency management and climate research. In a world ever more vulnerable to extreme weather events, accurate and timely weather forecasts are of paramount importance.

Advancements in NWP have undoubtedly revolutionized our ability to anticipate and prepare for atmospheric phenomena. As the field of meteorology evolved, so did the recognition of the inherent challenges in NWP, particularly in accurately predicting complex meteorological phenomena like precipitation [4]. This is due to the intricate nature of the processes governing precipitation, the inherent uncertainties in atmospheric dynamics, and the limitations of numerical models. NWP has its roots in Princeton in the 1950's [5], when computer technology and our understanding of atmospheric physics converged to usher in a new era of meteorology. By simulating the behavior of the atmosphere through complex mathematical equations, NWP models have enabled meteorologists to forecast weather conditions with increasing accuracy. These models consider a multitude of variables, such as temperature, pressure, wind speed, and humidity, to generate predictions about the state of the atmosphere at various points in the future. It was not until the 1970's when supercomputing power became more widely accessible that tackling the entire array of complex mathematical equations became a practical possibility [6]. Operational NWP centers now provide predictions covering a broad spectrum of time ranges, from very short-range forecasts at kilometer-scale resolutions, to global seasonal forecasts at resolutions of tens of kilometers [2].

As NWP technology continues to advance, it holds the promise of providing more accurate, reliable, and timely weather forecasts, thereby benefiting numerous sectors and helping society adapt to a world increasingly influenced by changing climate patterns and extreme weather events. In the following sections, we delve deeper into the motivations driving our research to improve precipitation forecasts by integrating machine learning (ML) and deep learning (DL) into post-processing of NWP models. Post processing NWP models, refers to a set of techniques used to refine and improve the output of NWP models.

1.1 Problem Statement and Motivation

Despite the remarkable progress made in NWP, it is incredibly difficult, if not impossible to mathematically represent every atmospheric process analytically [2]. To handle this, processes are discretized both in space and time. Discretization of process introduces a *resolved and unresolved* scales of motion within the equations. In other words, it divides the equations into components that can be accurately simulated (resolved) and others that are approximated (unresolved). These approximated unresolved processes require a form of *parameterization* to describe how they interact with the processes that are resolved. Parameterization account for radiation, convection, and diffusion within the atmosphere and at the boundary between the atmosphere and the Earth's surface. They are frequently influenced by relatively small spatial scales [7]. Figure 1.1 derived from [2] visually represents unresolved processes and their areas of significance. Even though they are not fully resolved, unresolved processes significantly impact heat and momentum balances at the grid scale and play a vital role in achieving predictive accuracy [8]. Inherent complexities of the problem entails introducing simplifications to make the numerical solution more manageable and to reduce the overall complexity of the equation set. Although with limited success, Richardson in [9] was the first to demonstrate these simplifications in 1920's.

Precipitation is a complex meteorological phenomenon characterized by a wide range of spatial and temporal scales, making it inherently difficult to predict. Traditional NWP models often struggle with capturing the subtle variations and abrupt changes in precipitation patterns. Moreover, the unresolved approximated processes discussed above result in significant biases and uncertainties in their forecasts. This persistence of errors in precipitation predictions can have profound

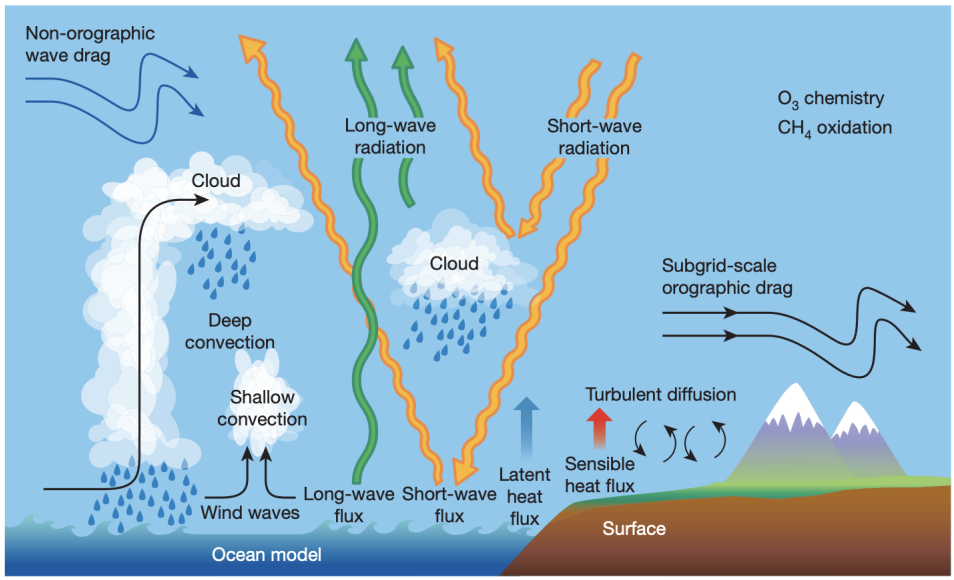

Figure 1.1: Figure from [2] visualizing unresolved physical processes of weather prediction.

consequences for both short-term weather forecasting and long-term climate modeling. It is crucial to explore alternative approaches to improve the precision and reliability of precipitation forecasts. Data assimilation and post-processing, the practice of refining NWP model outputs using additional data or statistical techniques, has emerged as a promising avenue for reducing uncertainties and enhancing the accuracy of NWP predictions [10]. The introduction of a plethora of ML and DL algorithms in the recent past have demonstrated their potential to mitigate the shortcomings of traditional NWP models, making it a subject of increasing interest in the meteorological community [11, 12].

1.2 Previous Work

In this section, we will discuss our previous work [1] on *Machine Learning Approaches to Improve North American Precipitation Forecasts*, which serves as the foundation for the research presented in this thesis. Our approach involved employing an ensemble forecasting strategy, consolidating 6 months (Dec 2021 - May 2023) of NWP precipitation forecasts from multiple meteorological agencies in Canada, the United States, and Europe. For the development of a machine learning assisted weather model, we conducted experiments utilizing five distinct machine learning

techniques: Multiple Linear Regression (MLR), Random Forest Regression (RFR), Gadient Boosted Regression (GBR), Fully Connected Neural Networks (NN), and Convolutional Neural Networks (CNN). These methodologies were explored alongside an existing averaging model that was already in use. Our findings demonstrated that ML approaches can indeed enhance the accuracy and reliability of NWP precipitation predictions. The most significant improvements were observed with neural network variants (NN and CNN), with 17% improvement in the mean absolute error, 3% in the root mean squared error and 5% in the maximum error. One of our objective was to train and deploy an optimal machine learning-based weather model for real-time precipitation forecasting that would surpass the performance of the baseline model. All of the five ML models, offered under one hour of training time and near-real-time prediction capabilities.

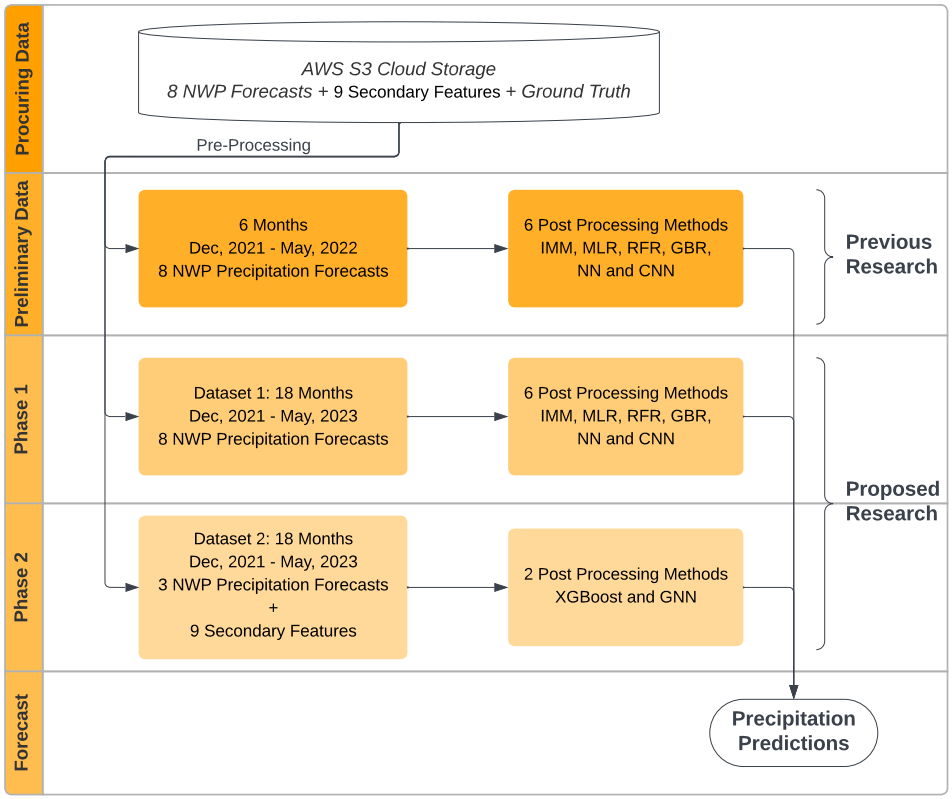

Figure 1.2: Overview of the previous and proposed research.

1.3 Proposed Research

We take advantage of the substantial growth of our dataset, now encompassing 18 months of data from December 2021 to May 2023. We have partitioned the dataset, utilizing 15 months for training (Dec 2021 up to March 2023) and allocated the subsequent months (March 2023 up to June 2023) for testing. Following convention of our previous work, we use the term *Weather Models (WMs)* to refer the NWP forecasts employed as inputs and the term *trained ML models* for the machine learning models developed. Figure 1.2 summarizes our previous and proposed research.

Longer temporal scope is integral to our research, since it gives the opportunity to train even more robust ML algorithms and also evaluate the significance of secondary features like temperature, wind, location, etc. To facilitate this analysis, we conduct experiments in two phases:

- **Phase 1:** In this phase we evaluate how, trained ML models employed in our previous work, evolve when dealing with a large temporal scale, providing a basis for comparison. For this *Dataset 1* is curated, which retains the same set of input features as the smaller dataset from our previous work, consisting of precipitation values from the 8 Weather Models (WMs) as shown in Figure 1.3.

- **Phase 2:** In this phase we will examine the impact of the secondary features, given the larger time span available for training. We curate *Dataset 2* as shown in Figure 1.3, which encompasses a subset of features from *Dataset 1*, i.e., the 3 best performing WMs, and introduce 9 additional secondary features discussed in section 4.5. We also deploy two additional Machine Learning models: XGBoost (XGB) and Graph Neural Networks (GNN), exclusively for *Dataset 2*.

Along with the regression metrics used in our previous work, we have introduced an additional assessment based on confusion matrices which enables us to gauge how ML models perform at different precipitation levels, offering a more nuanced understanding of their predictive capabilities. Furthermore, we incorporate Shapley values [13] into our analysis to investigate the contribution of each feature to the prediction. This method provides valuable insights into the importance of individual features and their impact on the overall forecast.

In this thesis,the objective is to provide a comprehensive exploration of the potential enhancements that ML and DL algorithms can bring to NWP precipitation

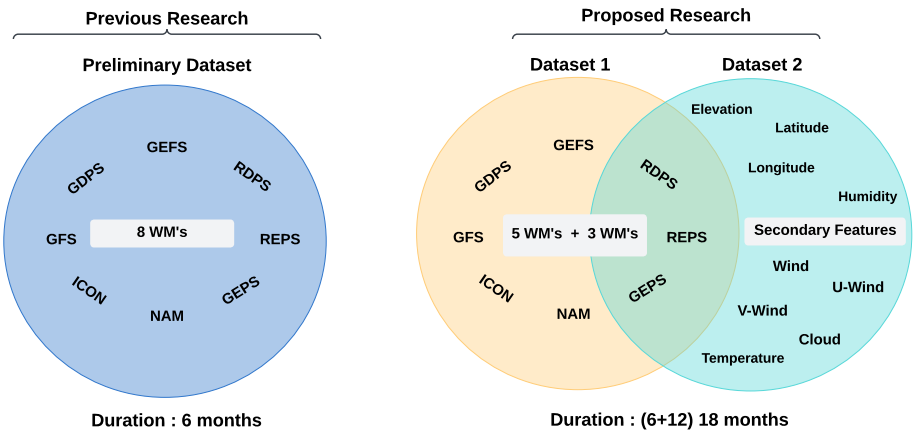

Figure 1.3: Venn diagram defining the datasets provided by Weatherlogics Inc. used in previous and proposed research.

predictions. Through an in-depth analysis of *Dataset 1 and Dataset 2*, along with the introduction of XGBoost and GNN models for Dataset 2, we anticipate shedding new light on the role of secondary features and their potential to further refine precipitation forecasts. Our evaluation metrics, including the Confusion Matrix and Shapley values, will contribute to a more holistic understanding of the performance and feature importance of ML models in this critical domain. This thesis, delves deeper into the exploration of ML and DL algorithms with a larger dataset, aiming to refine and expand the methodologies to achieve even greater precision and reliability in precipitation predictions.

1.4 Thesis Outline

Chapter 2 introduces related work in the field of weather prediction and post-processing NWP models. Chapter 3 lays the foundation of the relevant methodologies and metrics used to train as well as evaluate the machine learning models. Chapter 4 discusses the procuring and pre-processing the datasets used throughout the research. This chapter explains the process of feature selection for Dataset 1 and Dataset 2. The results and visualization of our experiments with the two datasets are contained in Chapter 5. Finally, Chapter 6 concludes on the findings of the previous chapters.

Chapter 2

Related Works

This section presents research related to machine learning techniques used for enhancing weather predictions in general like temperature, humidity and wind, among others. We also discuss research related to post processing forecasts generated by Numerical Weather Prediction (NWP) models with the utilization of machine learning and deep learning. Machine learning have long been used in the prediction of various weather parameter, it can also be applied to address complex meteorological challenges, such as predicting drought conditions, extreme weather events, and climate patterns. Post processing NWP models, refers to a set of techniques used to refine and improve the output of NWP models. These models, while valuable, often exhibit biases and errors in their forecasts. Post processing involve statistical corrections and adjustments to model outputs, aiming to provide more accurate and reliable weather predictions.

2.1 Weather Prediction and Machine Learning

Deo and Sahin's research [14] employs Artificial Neural Networks (ANN) to predict the Standardized Precipitation and Evapotranspiration Index (SPEI) in eastern Australia, incorporating various hydro-meteorological parameters and climate indices to create a robust predictive model. Le in research [15] investigates the application of Recurrent Neural Networks (RNNs) to project drought conditions in California. By considering the Palmer Z Index data and the influence of the 2015-2016 El Niño event, this study demonstrates a significant correlation between projected and observed drought conditions, highlighting the utility of RNNs in this context. Haidar [16]

proposes a deep convolutional neural network (CNN) for monthly rainfall forecasting in eastern Australia, achieving promising results in comparison to conventional forecasting models.

Yajing Wu et al. in research [17] proposes quantitative precipitation estimation based on graph convolutional regression networks. The estimation method converts the rain gauge network into a graph and uses GCRNs to learn two types of relationships: the nonlinear relationship between radar reflectivity and rainfall rate, as well as the spatial correlation among rain gauges. Jiahui Xu et al. in research [18] proposes HighAir, a hierarchical graph neural network-based method for air quality forecasting. HighAir constructs a city-level graph and station-level graphs to capture the spatial dependencies of air quality at different granularities. HighAir uses weather data and an encoder-decoder architecture on the dataset of Yangtze River Delta city group, and shows that it outperforms existing methods. Moreover, study [19] employs XGBoost machine learning to enhance the accuracy of extreme rainfall forecasts, improving hydrological predictions for flood inundation modeling. Results indicate improved performance for localized rainfall events, especially during heavy rainfall and typhoon events. The work presented in [20], uses CNNs to predict total column water vapour and other key atmospheric variables with six-hour time resolution. Their model is applicable for subseasonal-to-seasonal forecasting at lead times from two to six weeks.

Bochenek et al. in research [12] presents a comprehensive review of machine learning applications used in climate forecasts and numerical weather prediction. Piyush Joshi et al. in research [21] explores the use of Artificial Neural Networks (ANN) for precipitation forecasting in the Western Himalayan region using satellite images. The ANN model trained on infrared and water vapor images shows promise in qualitative and quantitative precipitation forecasting, with skill scores indicating its operational potential. Barrera et al. in research [22] does a comparative analysis of machine learning algorithms for rainfall prediction using time-series data. These studies evaluate LSTM, Stacked-LSTM, Bidirectional-LSTM, XGBoost, and AutoML models. Bidirectional-LSTM and Stacked-LSTM with two hidden layers emerge as the top-performing models in this context.

2.2 Post Processing NWP models with Machine Learning

To remove uncertainties and improve upon NWP models, researchers have developed several strategies, the most common being employing ensembles of multiple models [23, 24, 25, 26]. Buizza et al. in research [27] studied the effect of NWP ensemble sizes on its prediction accuracy, having different physical parameters as inputs to the same NWP model. Elizabeth Ebert in research [28] evaluated the use of an ensemble of seven independent NWP models and compared them to its individual members, over Australia. The ensemble models reduce the uncertainties of a single model but fail to map the non-linear relationships between the model output and real-world observations.

Krasnopolsky et al. in research [29] develop a nonlinear multimodel ensemble approach using neural networks to improve 24-hour precipitation forecasts. This approach significantly reduces biases and improves forecast accuracy compared to other methods. The NN multi-model ensemble showed significant improvements in reducing high bias at low precipitation levels, reducing low bias at high precipitation levels, and sharpening features to make them closer to the observed ones. Gagne et al. in research [30] leverages machine learning techniques to enhance storm-scale ensemble probabilistic precipitation forecasts. Logistic regressions and random forests correct for biases and improve forecast reliability, providing valuable insights into precipitation predictions. The study [31] propose an ANN to establish relationships between NWP ensemble forecast and observed 7-day precipitation accumulations. The study demonstrates that the ANN model compares favorably with a state-of-the-art postprocessing technique developed for medium-range ensemble precipitation forecasts.

Casper Sonderby et al. in research [32] introduces MetNet, a neural network that excels in high-resolution precipitation forecasting. It outperforms numerical weather prediction at extended lead times, showcasing the potential of deep learning in weather forecasting. Research [33] extends the work presented in [32] and presents Metnet2, a neural network for large-scale precipitation forecasting. This model surpasses state-of-the-art physics-based models, emphasizing the shift towards efficient forecasting using neural networks. Yun Fan et al. in research [34] propose a statistical post-processing method utilizing neural networks to enhance NOAA CFS week 3-4 forecasts for precipitation and temperature. Their results demonstrate significant improvements

in forecast accuracy compared to traditional methods. Fan et al. [35] conduct a comparative study of merging approaches for regional precipitation estimation, using satellite and reanalysis data. Four different methods, including neural networks, are employed, highlighting the potential for improved regional precipitation estimates. Cho et al. [36] introduce an ensemble learning approach combining deep learning and statistical methods for post-processing numerical weather prediction models. This approach enhances the accuracy of next-day maximum air temperature forecasts, critical for thermal disaster preparation. Frnda et al. in [37] focus on enhancing short-term forecasts of air temperature and precipitation by ECMWF using a neural network-based calibration model. Real observations from weather stations contribute to model training, showing promising improvements in forecast accuracy.

Li et al. in research [38] and Jha et al. in [39] used bayesian post-processing techniques discussed in [40] to combine multiple NWP forecasts reducing errors in rainfall prediction. Zhou et al. in [41] introduce QPFNet, a deep learning model using basic meteorological variables to forecast precipitation. QPFNet outperforms ECMWF's high-resolution model, demonstrating its potential for improved precipitation forecasting. In [42], authors introduce a 3D neural network (termed Pangu-Weather) for medium-range global weather forecasting by including height information as a new third dimension. The AI system includes a hierarchical temporal aggregation algorithm that involves training a series of models with increasing forecast lead times on 39 years of global reanalysis weather data.

In [43], graph neural networks are used to model a dense physical weather forecasting system which is a hybrid physics and machine learning system. In this model, 6-hour changes across 78 atmospheric variables on a spatial grid of 1-degree latitude/longitude and a 13-level pressure grid is considered. However, this model does not consider precipitation. The model is trained on the ERA5 reanalysis dataset from the European Centre for Medium-Range Weather Forecasts (ECMWF) as well as a subset of the Global Forecast System (GFS). Lam et al. in [44] refers to a research study conducted by a team at Google DeepMind introduces a ML based method called *GraphCast* for global weather forecasting. GraphCast is trained directly from ERA5 data and significantly outperforms traditional numerical weather prediction systems in terms of accuracy and efficiency. This represents a key advancement in weather forecasting and demonstrates the potential of graph based machine learning in modeling complex dynamical systems. Dong et al. in research [45] used XGBoost to bias correct the Global Ensemble Forecast System forecasts. The study used data

from 689 meteorological stations across seven different climatic regions in China. XGBoost outperformed forecasting precipitation for 1-8 days ahead.

Chapter 3

Preliminaries

3.1 Introduction

This chapter explores the integration of various ML techniques, including linear regression, ensemble methods (Random Forest, Gradient Boosting Regression and XGBoost), neural networks (Simple Neural Networks, Convolutional Neural Networks and Graph-based Neural Networks), into the post-processing of NWP precipitation predictions. In the upcoming sections, we will explore how each machine learning technique works and how we can use them to make weather forecasts better. We will also introduce the metrics used to evaluate the performance of the input weather models and the machine learning approaches.

3.2 Input Means Model (Baseline)

The customary method used in practice to post-process these weather models is to simply average the inputs from multiple WMs together denoted by Eqn 3.1, in-turn reducing the biases in the forecast of a particular WM. Throughout this thesis, we will refer to this method as the Input Means Model (IMM) which will also serve as the baseline method of post-processing.

$$\hat{y}_{\text{IMM}} = \frac{\sum_{i=1}^{N} WM_i}{N}. \tag{3.1}$$

Here, N is the number of input weather model forecasts, WM_i is the input from the i^{th} weather model and $\hat{y}_{\text{IMM}}$ is the averaged prediction. However, IMM has certain limitations that need to be considered. It assumes that each WM contributing to the

average is equally weighted. In reality, some models may have a better track record and should be given more weight, while others might perform poorly and should be down-weighted or excluded. Additionally, the IMM does not account for the fact that different models may have different biases for different weather conditions or regions. In essence, while the IMM serves as a simple and intuitive baseline for post-processing WM forecasts, it lacks the sophistication required to address the complexity of real-world weather prediction challenges.

3.3 Machine Learning Fundamentals

Machine learning can be divided broadly into three sub categories, supervised, unsupervised and reinforcement learning. In this research, we will be focusing on the first category, supervised learning wherein weather models are trained on labeled datasets, and each data point is paired with a corresponding output label. As defined in section 2.2.1 of research [46], the goal of supervised learning is to find a mapping function $f : \mathbb{R}^D \rightarrow \mathbb{N}$ for classification or $f : \mathbb{R}^D \rightarrow \mathbb{R}$ for regression where input vector $\mathbf{x}$ is D dimensional and the output of the function $f(\mathbf{x})$ returns a real number for regression and a label for classification problems. The learning process involves minimizing a selected loss function like mean squared error during training, which guides the model towards making accurate predictions with small iterations.

3.4 Multiple Linear Regression

Multiple linear regression (MLR) is one of the most commonly used statistical machine learning tool available. It is a supervised machine learning algorithm that computes the linear relationship between a dependent variable ($\hat{y}_{MLR}$) and one or more independent features denoted by (WM_i) as shown in Eqn 3.2.

$$\hat{y}_{MLR} = \beta_0 + \sum_{i=1}^{N} \beta_i \star WM_i. \tag{3.2}$$

Here, N is the number of weather model forecasts input to the MLR model. Training the linear regression model involves estimating the coefficients (β_i) and the intercept (β_0) with a goal to minimize the sum of squared errors (SSE) between the actual and predicted values.

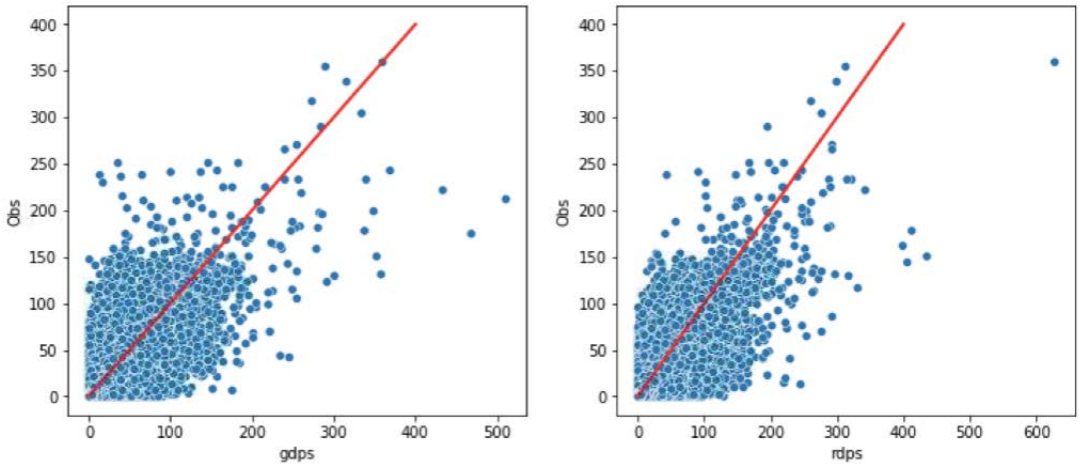

Figure 3.1: Non-Linear relationship between the precipitation forecasts from GDPS, RDPS WM's and Observed values.

MLR assumes that the input features WMs are not highly correlated and the relationship between the attributes and the observed values is a linear one. Which might not always be the case as demonstrated in Figure 3.1.

Accurate identification and modeling of these relationships are crucial in machine learning for making precise predictions and understanding the underlying patterns in data. If these assumptions are not met, the linear regression algorithm is unable to produce reliable predictions. Careful consideration of these assumptions is essential when choosing MLR to post-process WMs precipitation forecasts.

3.5 Ensemble Methods

Ensemble methods, a very popular method in machine learning, revolves around a concept of combining multiple individual weak models and gather their collective wisdom to enhance predictive accuracy. In this study, the core building blocks of the ensemble methods employed are decision trees. Decision trees discussed in [47] are helpful because they can handle complex decision-making processes and are easy to understand, like a series of *if-else* questions, making them valuable for weather forecasting, an example decision tree is illustrated in Figure 3.2.

The upcoming sub-sections, will discuss the most popular ensemble methods, particularly Random Forest, Gradient Boosting Regression (GBR), and XGBoost, and explore their applications in post-processing WMs precipitation forecasts.

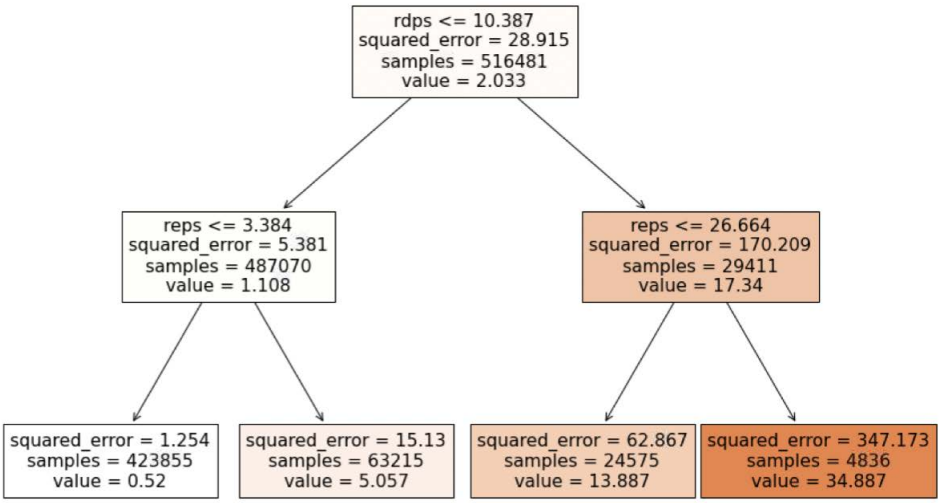

Figure 3.2: Sample decision tree diagram with depth 2 used for predicting the value of precipitation where sample refers to the data points covered within that leaf and RDPS, REPS are the input WM's.

3.5.1 Random Forest

Random Forest regression illustrated by Leo Breiman in [48] is a robust ensemble learning method that excels in capturing complex relationships within data, making it particularly valuable for improving WMs forecasts. Several decision trees are constructed independently, each based on a subset of training samples, and their individual predictions are later combined by taking an average as depicted in Figure 3.3.

Mathematically a random forest can be represented as Eqn 3.3.

$$RF(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N} T_n(\mathbf{x}), \tag{3.3}$$

where, $RF(\mathbf{x})$ represents the final prediction for the random forest for an input vector $\mathbf{x}$, N represents the total number of trees in the forest and $T_n(\mathbf{x})$ is the prediction of the n^{th} decision tree. The ensemble approach promotes robustness and minimizes overfitting by introducing randomness during both the data sampling (bootstrapping) and feature selection processes for each tree. This diversity among trees helps reduce variance and increases the model's generalization performance.

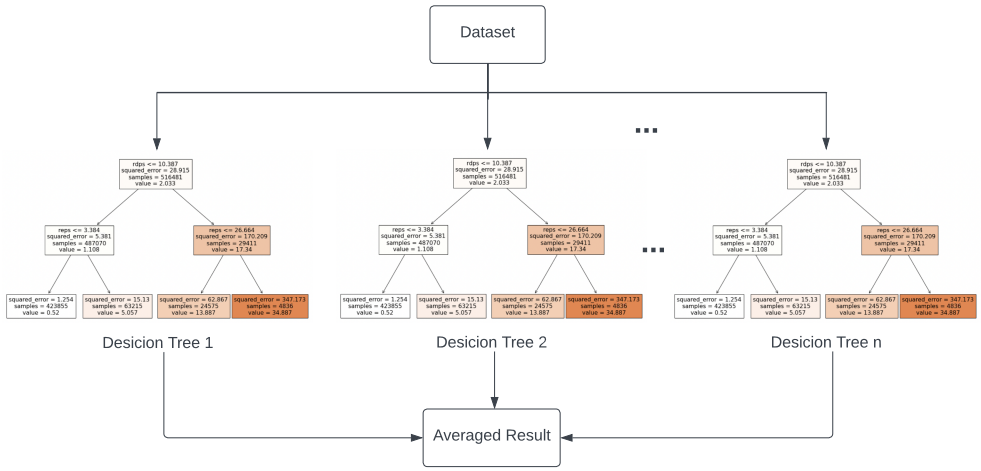

Figure 3.3: Random Forest ensemble model consisting of multiple decision trees, with the parameter n denoting the number of trees in the ensemble.

3.5.2 Gradient Boosting Regression (GBR)

Gradient Boosting is another powerful ensemble learning technique that has gained popularity in machine learning. The concept of gradient boosting was first introduced by Jerome Friedman in 2001 in his paper titled *Greedy Function Approximation: A Gradient Boosting Machine* [49]. GBR builds an ensemble of weak decision trees, similar to a Random Forest as shown in Figure 3.4. However, the trees are trained sequentially in order to correct errors made by previously constructed weak trees.

Training data D containing input vector $\mathbf{x} = (x_0, \dots x_n)$ and target vector $\mathbf{y} = (y_0, \dots y_n)$ is given as $D = \{(x_i, y_i)\}_{i=1}^{n}$, where N is the number of training examples, and x_i is an i^{th} object with a corresponding true label y_i. In this iterative process, we train M iterations of weak learners sequentially, each denoted as $f_m(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$, where $f_m(\mathbf{x})$ is the prediction of the m^{th} weak learner and the prediction of the ensemble is denoted as $F(\mathbf{x}) = \sum_{m=1}^{M} f(\mathbf{x})$. The first prediction, $f_0(\mathbf{x})$ is initialized by finding the argument that minimizes the loss function L, and is given by $f_0(\mathbf{x}) = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} L(y_i, f(x_i))$. For every iteration or boosting round $m = 1$ to M, the algorithm proceeds by calculating residuals r_{im}, given by Eqn 3.4.

$$r_{im} = -\left[\frac{\delta L(y_i, F(x_i)}{\delta F(x_i)})\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})}. \tag{3.4}$$

Subsequently, a weak learner is trained to predict these residuals r_{im}. The goal of this

step is to find a model $f_m(\mathbf{x})$ given in Eqn 3.5, that captures the corrections needed to reduce the total loss.

$$f_m(\mathbf{x}) = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} L(r_{im}, f(x_i)). \tag{3.5}$$

The ensemble model is updated by adding a scaled version of the new weak learner. The scaling factor, represented by ϑ, is called the learning rate and is typically a small positive value $\vartheta \in \{0, 1\}$. The updated ensemble prediction is given by $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \vartheta f_m(\mathbf{x})$. After all the M boosting rounds are completed, the final ensemble model is given as $F(\mathbf{x}) = F_M(\mathbf{x})$.

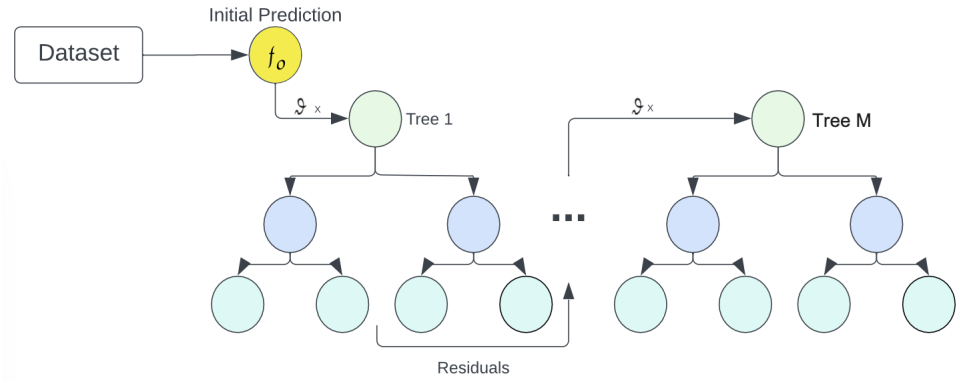

Figure 3.4: This figure illustrates a Gradient Boosting ensemble model. With the parameter M, indicating the number of trees in the ensemble

This approach is especially powerful because it allows you to, effectively handle complex, non-linear relationships within the data by iteratively adjusting and improving the functions. It's like solving a puzzle where each piece makes the picture clearer.

3.5.3 Extreme Gradient Boost (XGboost)

XGBoost, short for Extreme Gradient Boosting, is a state-of-the-art ensemble learning algorithm that has gained immense popularity in various machine learning applications, including the enhancement of NWP precipitation forecasts. Introduced by Tianqi Chen [50], XGBoost combines the strengths of gradient boosting and ensemble techniques, making it a powerful tool for capturing complex relationships within data. XGBoost builds upon the principles of gradient boosting as discussed

in the above subsection, additionally incorporates regularization terms into its loss function. This helps control the complexity of individual trees and encourages the selection of relevant features. In the XGBoost's introductory paper [50] the objective function Obj^t, for the i^{th} tree instance and at the t^{th} iteration, is given by Eqn 3.6.

$$Obj^t = \sum_{i=1}^{n} L(y_i, F_{t-1}(x_i) + F_t(x_i)) + \Omega(F_t), \tag{3.6}$$

where $\Omega(F_t) = \gamma T + \frac{1}{2}\lambda||w||^2$ and F_t corresponds to t^{th} tree structure and Ω is the regularization parameter that penalizes the individual tree in the model. Ω comprises of adjustable parameters γ and λ, which regulates the number of tree leaves T and learnt weights w, to avoid over-fitting. With a higher value of λ the optimal output value $F_t(x_i)$ would be closer to 0.

Solving for the optimal output value, XGboost applies the second-order Taylor approximation to Eqn 3.6 and the objective function is later differentiated, which means we can ignore the constant values in the Taylor approximation resulting in Eqn 3.7.

$$Obj^t = \sum_{i=1}^{n} [g_i F_t(x_i) + \frac{1}{2} h_i F_t^2(x_i)] + \Omega F_t(x_i), \tag{3.7}$$

where $g_i = \frac{\delta L(y_i, F_{t-1}(x_i))}{\delta F_{t-1}(x_i)}$ and $h_i = \frac{\delta^2 L(y_i, F_{t-1}(x_i))}{\delta^2 F_{t-1}(x_i)}$ are the first and second order derivatives of the loss function L respectively. Substituting the value of Ω and the Eqn 3.7 can be rewritten as Eqn 3.8

$$Obj^t = \sum_{i=1}^{n} g_i(F_t(x_i)) + \frac{1}{2}\sum_{i=1}^{n} h_i(F_t^2(x_i)) + \gamma T + \frac{1}{2}\lambda||w||^2. \tag{3.8}$$

To minimize the objective function Eqn 3.8 we set the derivative equal to 0 resulting in the optimum output value for the t^{th} iteration $F_t(x_i)$ shown in Eqn 3.9

$$F_t(x_i) = -\frac{\sum_{i=1}^{n} g_i}{\sum_{i=1}^{n} h_i + \lambda}. \tag{3.9}$$

XGBoost can also incorporate sample weights that assigns different weights to individual data points during the training process. It allows you to emphasize the importance of certain data points by assigning them higher weights, which can be useful when dealing with imbalanced datasets, discovered in this research. Regularization terms and gradient-based optimization make XGBoost a powerful and

robust algorithm for a wide range of machine learning problems. It is designed for parallel and distributed computing, allowing it to efficiently utilize multiple processors and even use a gpu for faster processing. This is especially beneficial when dealing with large datasets and complex models like the one used in this research.

3.6 Neural Networks

In recent years, neural networks (NNs) have emerged as powerful tools in the field of machine learning, demonstrating remarkable capabilities in various applications, including natural language processing, image recognition, and, notably, weather forecasting [51, 52, 53, 54, 55]. Neural networks draw inspiration from the structure and functioning of the human brain, consisting of highly interconnected neurons also known as perceptrons [56] that process the incoming information.

As depicted in the Figure 3.5, each perceptron P multiplies the weights $\mathbf{w} = (w_0, \dots w_n)$ with their corresponding inputs $\mathbf{x} = (x_0, \dots x_n)$ to have a weighted sum and add a learnable bias (b) to get the intermediary value $I = \sum_{i=0}^{n} \mathbf{W}_i \times \mathbf{x}_i + b$.

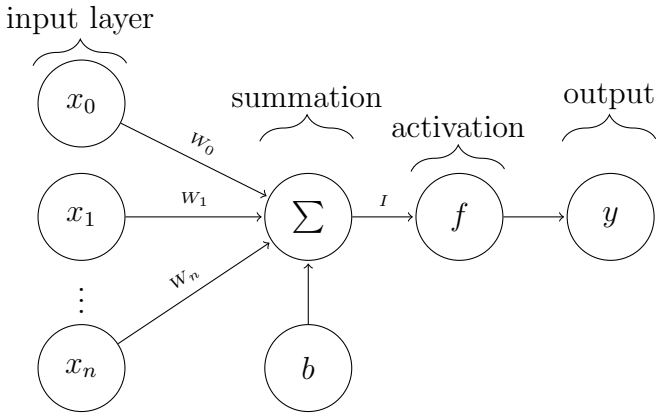

Figure 3.5: Figure provides an illustration of a Perceptron, which is one of the fundamental building blocks of artificial neural networks. Where, $\mathbf{x} = (x_0, \dots x_n)$ is the input vector, $\mathbf{w} = (w_0, \dots w_n)$ is weight vector and b is the bias.

The intermediary value is then passed to an activation function (f) associated with each neuron, which introduces non-linearity into the model. A common activation function for regression is ReLU (Rectified Linear Unit), that can be defined as $max(0, I)$. The activation function determines the neuron's final output y, as shown in Eqn 3.10.

$$Output(y) = f(0, \sum_{i=0}^{n} \mathbf{W}_i \times \mathbf{x}_i + b). \tag{3.10}$$

Although a single perceptron is incredibly powerful to replicate any linearly separable functions. Many real-world problems are not linearly separable, a single perceptron struggles to capture complex, nonlinear relationships in data [57]. This means they cannot learn or compute functions like XOR or parity, which are not linearly separable. Additionally, perceptrons themselves have no learning capacity beyond simple weight and bias adjustments.

3.6.1 Neural Networks (NN)

An interconnected multiple-layer perceptron, or a fully connected neural networks, overcomes the limitations of single perceptrons. It consists of multiple interconnected layers of neurons (perceptrons) with nonlinear activation functions. These neurons are organized into layers, typically divided into three primary types as depicted in the Figure 3.6. The input layer receives raw data as input, which could be numerical values, images, or any other type of data relevant to the problem at hand which are then passed to the hidden layer. The hidden layers are the intermediate layers between the input and output layers. They play a crucial role in capturing complex patterns and features within the data. The final layer or the output layer, produces the network's predictions or classifications based on the patterns learned in the hidden layers.

Neural Networks learn through a process known as training, during which they adjust their internal parameters (weights and biases) to minimize a custom loss function. The learning process or training can be divided into two major phases forward propagation and backpropagation.

Forward Propagation

When a neural network is created, it is generally initialized with random values, i.e. the weights of each perceptron is a matrix with random numbers $\mathbf{W} \in \mathbb{R}$. During forward propagation, data is fed through the network from the input layer to the output layer via the hidden layers. Neurons in each layer perform weighted summations of their inputs, pass the result through the activation function as shown in Eqn 3.10, and forward the processed data to the next layer. This process continues

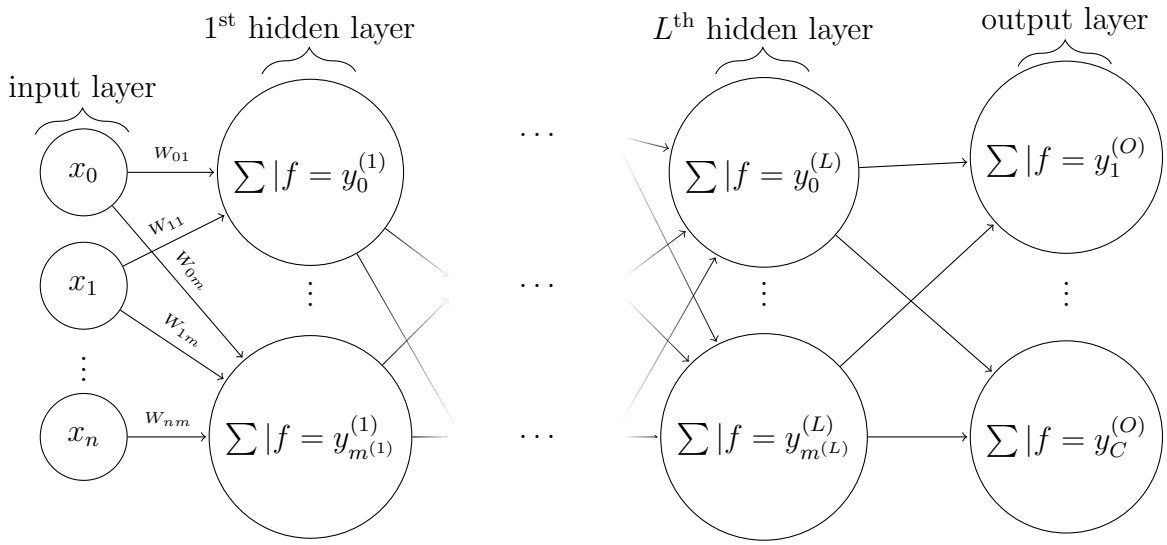

Figure 3.6: Network graph of a $(L+1)$-layer perceptron with $N+1$ input units, W_{ij} is the weights associated with neuron i and j and C are the number of output units. The l^{th} hidden layer contains $m^{(l)}$ hidden units.

until the output layer produces predictions. After the predictions are made, the randomly assigned weights are now adjusted by a process known as backward-propagation or Backpropagation.

Backward Propagation

Backward propagation introduced by [58] is the heart of neural network training. It involves calculating the gradients of the loss function with respect to the network's parameters (weights and biases) as shown in Figure 3.7.

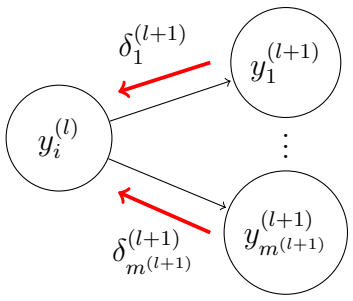

Figure 3.7: Backpropagation of errors through the network.

.

The parameter (θ_i^l) for the i^{th} neuron in layer l of the neural network, are learnt

to minimize the loss function (L). Inferring back-propagation defined in [58], partial derivative of the loss function (L) with respect to each parameter (θ_i^l) is computed with the help of chain rule [59] shown in Eqn 3.11.

$$\frac{\delta(L)}{\delta(\theta_i^l)} = \frac{\delta(L)}{\delta(y_i^l)} \frac{\delta(y_i^l)}{\delta(\theta_i^l)}, \tag{3.11}$$

where y_i^l is the output of the i^{th} neuron in layer l and the derivative $\frac{\delta(L)}{\delta(\theta_i^l)}$ can be intuitively seen as the effect of a parameter on the overall loss. Eqn 3.12 gives us the updated parameters by applying a learning rate to prevent overfitting.

$$new\_\theta_i^l = \theta_i^l - (learning\_rate \times \frac{\delta(L)}{\delta(\theta_i^{l+1})}). \tag{3.12}$$

After the parameter update forward and backward propagation cycles are repeated until a desirable output is achieved. This looping through forward and backward propagation is also popularly known as an epoch.

While NN have demonstrated significant success in a wide range of applications, they also come with certain challenges and considerations that are important to address. Precipitation data is inherently spatial in nature. However, the NN model that is adopted does not consider spatial information. As a result, data points that are spatially close are treated no different that those that are distant. This leads to a loss of spatial context which is quite important in many real-world scenarios.

3.6.2 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) introduced by [60] have emerged as a powerful class of deep learning models that excel in various computer vision tasks, including image classification, object detection, and semantic segmentation [61, 62, 63, 64]. Their ability to automatically learn spatial/grid-like features from data makes them a compelling choice for improving the quality of WM's precipitation forecasts which are spatial in nature. At their core, CNNs are composed of layers similar to NN, additionally CNN's use convolution and pooling layers defined below to extract spatial context from the input grid or image as shown in Figure 3.8.

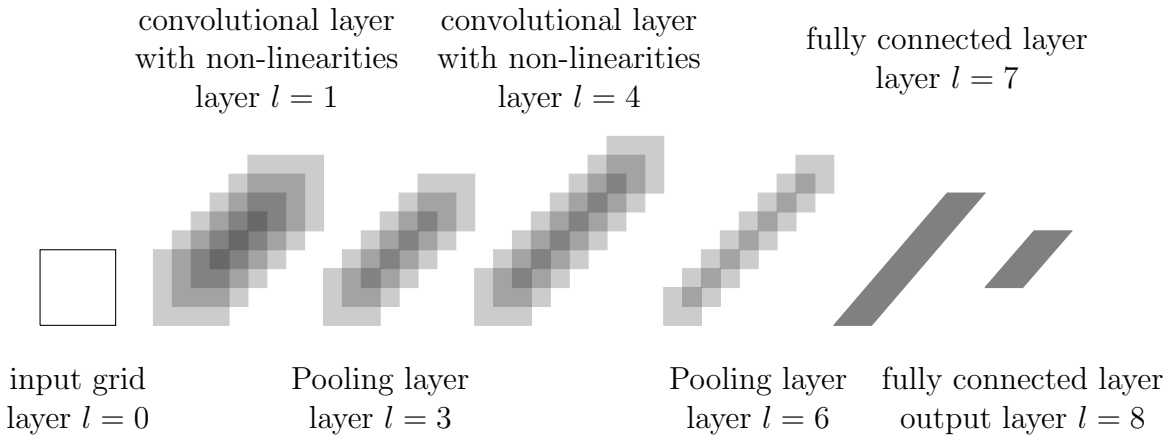

Figure 3.8: The architecture of the original convolutional neural network, as introduced by LeCun et al. (1989). The convolutional layers already include non-linearities and, thus, a convolutional layer actually represents two layers.

Convolution Layer

Convolution layers apply a set of learnable weights or filters (f) to the input data, allowing to detect local patterns and features. The convolution operation is particularly effective at capturing spatial information. It achieves this by systematically overlaying a matrix of weights over each element and its neighbors in the grid and later taking a weighted sum. The convolution operation can be done in parallel over each element in the image essentially smoothening the entire image as depicted in the Figure 3.9.

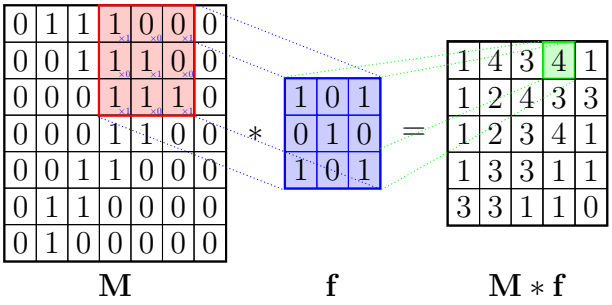

Figure 3.9: Convolution operation on the matrix **M** and filter **f** of size 3x3.

This capability is especially valuable when working with precipitation data. For instance, if there is a sudden spike in precipitation values at a specific location on the grid, potentially resulting from an incorrect reading, applying a convolution layer

would simply blur the spike and help neural networks to generalize better. The weights of the filter are learnt during the back-propagation, whereas the filter size is tune-able hyper-parameter set before the training. As the convolution operation is dependent on the neighboring pixels, the edges of the matrix M are padded with 0's for this research. It helps with keeping the dimensions of the image after convolution operations consistent.

Pooling Layer

Pooling layers downsample the spatial dimensions of the data, reducing its computational complexity while retaining important information. The pooling layer summarizes the features present in a region of the feature map generated by a convolution layer. Common pooling operations include max pooling and average pooling. The pooling filter works similarly to a convolution filter by overlapping on the grid, but instead of taking a weighted sum, it essentially takes the maximum of the grid elements within that max-pool filter. Further operations in the CNN are performed on the pooled features, instead of the precisely positioned features generated by the convolution layer. As a result, the model is more resilient to variations in the position of the features in the input grid.

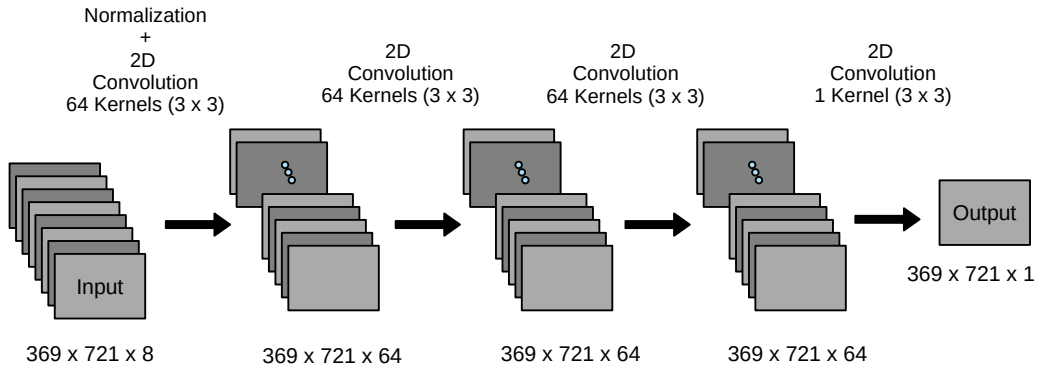

Figure 3.10: Figure describes a CNN architecture used in this study.

The training of a CNN is quite similar to the NN, the inputs are first passed to convolutional layer followed by a element wise activation layer that introduces non

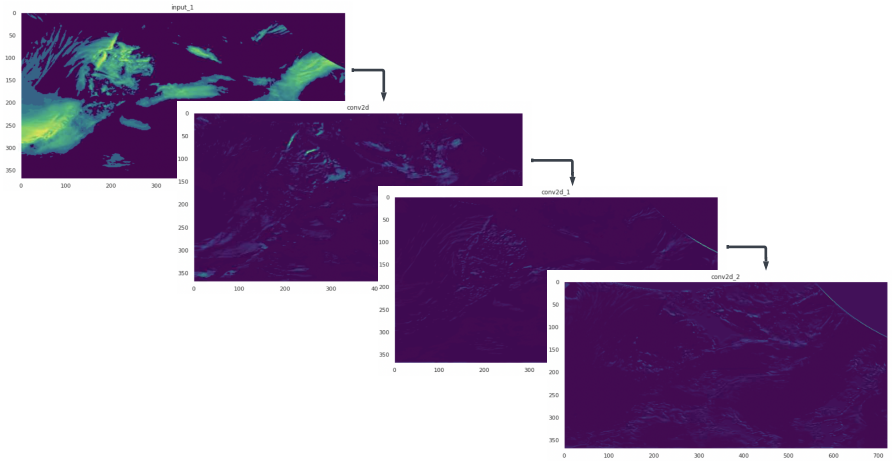

Figure 3.11: Figure visualizes the outputs from the three convolution layers.

linearity into the model as shown in Figure 3.10. Then the outputs are pooled to get a overview and reduce dimensionality. After this, depending on the underlying complexity, the outputs are fed through multiple convolution and pooling layers that helps identify hierarchical features in the data. Lower layers tend to capture simple features like edges, corners, and textures, while higher layers combine these features to represent more complex and abstract concepts. This hierarchical feature learning makes CNNs particularly suitable for improving WMs, as precipitation forecasting often involves recognizing intricate spatial patterns at various scales as depicted in Figure 3.11.

For image classification tasks the outputs of the CNN and Pooling layers are flattened and fed into a NN for further processing. As mentioned above the weights of the convolution filters are learnable, hence backpropagation is applied to learn these weights and improve the model accuracy. The ability of convolutional layers to perform local, grid-based operations makes them a valuable tool for handling gridded weather data. The convolution layers recognize local features indicative of precipitation, such as cloud formations, temperature gradients, and wind patterns. By learning and extracting these patterns, CNNs can contribute to more accurate precipitation forecasts.

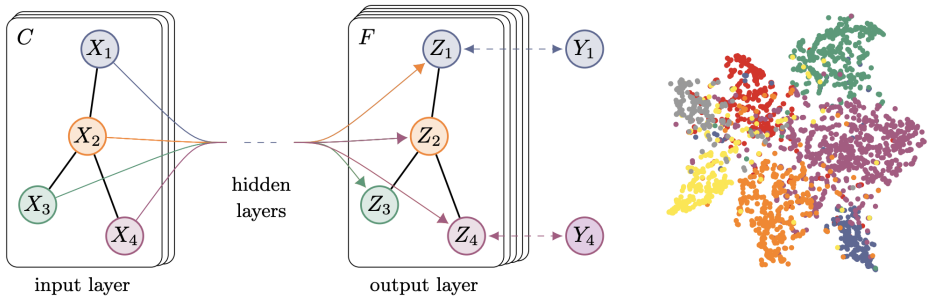

Figure 3.12: Left: Schematic depiction of multi-layer Graph Convolutional Network (GNN) for semisupervised learning with C input channels and F feature maps in the output layer. The graph structure (edges shown as black lines) is shared over layers, labels are denoted by Y_i. Right: t-SNE (Maaten and Hinton, 2008) visualization of hidden layer activation of a two-layer GNN trained on the Cora dataset (Sen et al., 2008) using 5% of labels. Colors denote document class.

3.6.3 Graph Convolutional Neural Networks (GNN)

The paper *Semi-Supervised Classification with Graph Convolutional Networks* [65] by Thomas N. Kipf and Max Welling presents a novel approach for semi-supervised learning on graph-structured data. This approach, based on an efficient variant of convolutional neural networks which operate directly on graphs as shown in the example Figure 3.12 given by [65], can be adapted for node level regression tasks, such as predicting precipitation. GNNs learn hidden layer representations that encode both local graph structures and features of nodes. This makes them particularly suitable for tasks where spatial relationships between nodes play a crucial role. In the context of predicting precipitation, each node in the graph represents a weather station and the edges represent the spatial relationships between nodes such as proximity. An adjacency matrix (A) is used to represent if a connection between nodes i, j of a graph exists, where $A_{ij} = 1$ if node i and j are connected and 0 otherwise. Adjacency matrix is a square matrix of size $N * N$, where N is the number of nodes (grid points). The node features are represented as another matrix X of size $N * D$ where D is the number of features (WM forecasts and other meteorological variables) and each row of X represents the feature vector of a node.

GNN also comprises of convolution layers similar to a convolution neural network but instead of working on a matrix/grid-like structure, GNN operates on graph structures. A convolution layer (H^{l+1}) is described by the Eqn 3.13, propagates

information from neighboring nodes connected through the graph edges.

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{l}W^{l}), \tag{3.13}$$

where, $\tilde{A} = A + I$ denotes the adjacency matrix with inserted self-loops I and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ is the diagonal degree matrix. The adjacency matrix can include other values than 1 and 0 representing edge weights like proximity information between two points. H^0 is the original inputs feature matrix X as $H^0 = X$, with dimension as $N * D$, and H^l indicates the l^{th} layer hidden representation of graph. W^l are the layer specific weight matrix learnt during the back-propagation phase. While the original paper focuses on semi-supervised classification, the underlying principle of GNNs is applicable to regression tasks as well. The key difference lies in the final layer of the network and the loss function. For regression tasks, a linear activation function like ReLU can be used in the final layer to output continuous values. In our research, a weighted mean squared error (MSE) is used as the loss function.

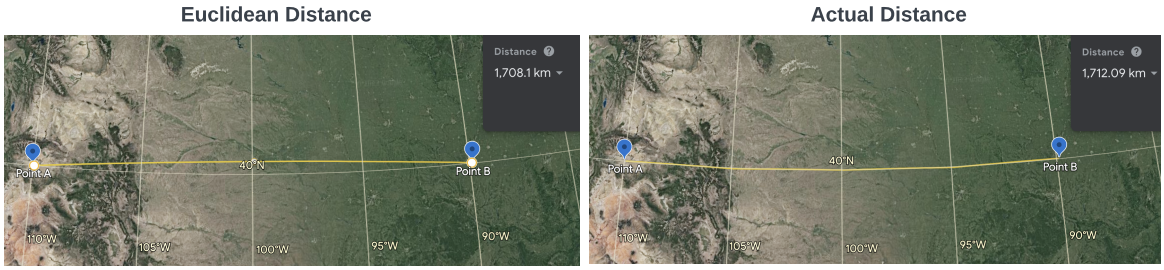

Figure 3.13: Distance measured between (40'N, 110'W) and (40'N, 90'W) assuming an euclidean coordinated system vs actual distance

In meteorology and many other geo-spatial applications, positional coordinate information is crucial for understanding and predicting patterns. GNN's offer several advantages when working with geospatial data. Unlike traditional CNNs, which operate on rigid structured grids, GNN's can adapt to irregular graph structures. This adaptability is crucial when working with latitude and longitude data, where distances between locations cannot be mapped in a euclidian space as shown in Figure 3.13. GNNs can learn to weigh edges differently, giving more importance to nearby locations while accounting for the Earth's curvature. Additionally in weather prediction, missing data is a common issue due to the sparse distribution of weather stations. GNNs offer the flexibility to modify or cut edges in the graph to ignore

nodes with missing data. This allows the network to make predictions even when data is unavailable for certain locations.

As mentioned in [65] GNNs can capture global context information even a few layers, allowing them to model large-scale weather patterns and interactions between distant locations. This enables the network to learn complex spatial dependencies, which is particularly useful for tasks like NWP precipitation forecasting where spatial distances and relationships are extremely essential.

Table 3.1: Metrics.

Name	Formula
MAE	$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\lvert\hat{y}_i - y_i\rvert$
RMSE	$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$
MdAE	$\text{MdAE} = \text{median}_{i=1}^{n}\lvert\hat{y}_i - y_i\rvert$
MaxE	$\text{MaxE} = \max_{i=1}^{n}\lvert\hat{y}_i - y_i\rvert$
CC	$\text{CC}(X,Y) = \frac{\text{Cov}(X,Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$
RB	$\text{Relative Bias} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{\sum_{i=1}^{n}(y_i)}$
POD	$\text{POD} = \frac{\text{True Positives}}{\text{True Positives+False Negatives}}$
FAR	$\text{FAR} = \frac{\text{False Positives}}{\text{True Positives+False Positives}}$
CSI	$\text{CSI} = \frac{\text{True Positives}}{\text{True Positives+False Negatives+False Positives}}$
CM	CM $\ni$ True Positives, False Positives, False Negatives, True Negatives

3.6.4 Evaluation Metrics

In this research, we considered a number of metrics to evaluate our trained ML models. This section provides an introduction to the performance metrics utilized, which are outlined in Table 3.1.

- Mean Absolute Error (MAE) [66] provides an understanding on the average error each model makes across all observations. It is defined as the average

absolute difference between the predicted and actual values, where n is the number of observations, $\hat{y}_i$ is the predicted output and y_i is the actual output for the i^{th} observation.

- Root Mean Squared Error (RMSE) [66] penalizes larger errors. It is defined as the square root of the mean squared error (MSE), where n is the number of observations, $\hat{y}_i$ is the predicted output and y_i is the actual output for the i^{th} observation. RMSE is the average squared difference between the predicted and actual values.

- Maximum Error (MaxE) provides an insight into the worst-case performance scenario. It is defined as the maximum absolute difference between the predicted and actual values, where n is the number of observations, $\hat{y}_i$ is the predicted output and y_i is the actual output for the i^{th} observation.

- Median Absolute Error (MdAE) is a more robust measure with respect to outlier values. It is calculated by taking the median of the absolute differences between the predicted and actual values.

- Pearson Correlation Coefficient (CC) describes the level of association between the predicted and actual output values, where X and Y are predicted and actual output values respectively, $\text{Cov}(X, Y)$ is the covariance of the two variables X, Y, $\text{Var}(X)$ is the standard deviation of X and $\text{Var}(Y)$ is the standard deviation of Y. A CC value of 1 indicates a strong positive relationship, a CC value of -1 indicates a strong negative relationship, and a CC value of 0 indicates no relationship at all.

- Relative Bias (RB) gives insight on whether a model tends to either over-estimate or under-estimate the output (e.g., precipitation value). It is defined as the average difference between the predicted and actual values relative to the mean of the actual values, where n is the number of observations, $\hat{y}_i$ is the predicted output and y_i is the actual output for the i^{th} observation.

- Probability of Detection (POD) measures how well the event of interest (precipitation) is detected. It is a measure of the ability of a classification model to correctly predict the presence of a particular class (in our case rainfall amounts). *True Positives* are the number of observations where both the predicted and actual values are same, and *False Negatives* are observations

where a prediction wrongly indicates that an event did not occur, when in fact it did.

- False Alarm Ratio (FAR) measures the rate of erroneous precipitation forecasts. True Positives are the number of samples where both the predicted and actual values are same and False positives are instances where a test or prediction wrongly indicates that an event or condition has occurred.

- Critical Success Index (CSI) is a metric which essentially combines POD and FAR metrics into a single score. Specifically, it measures the ratio for the correctly predicted precipitation events to the sum of hits, misses and false alarms.

- Confusion Matrix (CM) Confusion matrices are a valuable tool for assessing the performance of models when it comes to predicting precipitation categories, such as Nil, Light, Moderate, and Heavy. These matrices provide a detailed breakdown of how well a model classifies observations into these categories. In the context of precipitation prediction, a confusion matrix helps us understand not only the accuracy of the model's predictions but also its ability to distinguish between different levels of precipitation. The insights gained from confusion matrices are instrumental in evaluating the model's ability to differentiate between various precipitation intensities.

Chapter 4

Data Procuring, Curating, and Preparing

The quality and reliability of any weather prediction system are inextricably tied to the integrity of the data that underpins it. This chapter elucidates the meticulous processes undertaken in gathering and pre-processing the data that fuels this research. Specifically, we explore the selection of eight distinct weather models along with their meteorological features, each contributing a unique perspective on atmospheric conditions, and the intricate steps undertaken to ensure the compatibility and consistency of the data from these models.

4.1 Geographical Area Covered

We consider a geographical area that covers the majority of the continental USA and Canada, along with the surrounding region, as shown in Figure 4.1. Specifically, it is confined to the 24th parallel from the south, 70th parallel from the north, 218th meridian from the west, and 308th meridian from the east. The area is gridded at an increment of 0.125 degrees, starting from the origin at the 24th parallel and 218th meridian. This results in potentially useful 369 rows and 721 columns, 266,049 grid units to forecast the daily accumulated precipitation.

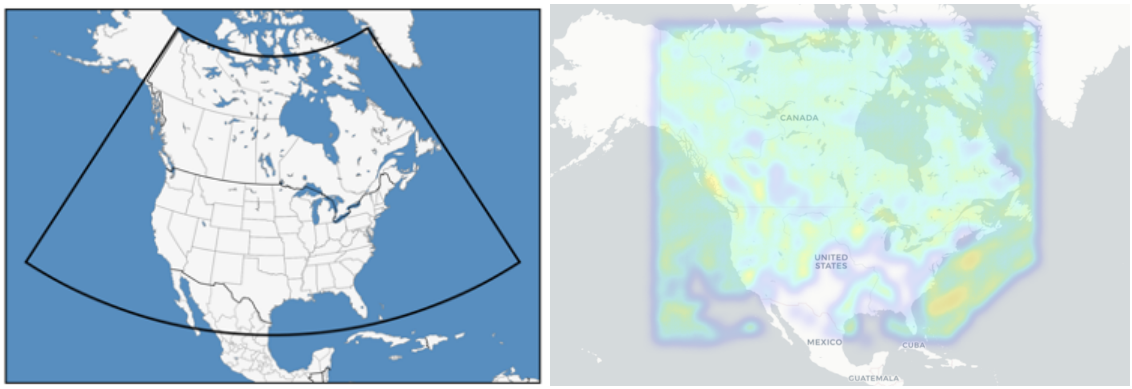

Figure 4.1: The region of study

4.2 Input Weather Model Details

Table 4.1 presents the spatial and temporal attributes of the samples of each of the WMs as well as their original accumulation periods for precipitation forecasts. The grid spacing in Table 4.1 represents the WMs after they were regridded for this study. The WMs natively use various different grid spacings, so they were regridded to a common grid to make processing easier. The feature of interest is the daily accumulated precipitation which is considered as a *primary input feature* as well as the *target output* for the ML models. Dozens of other variables ranging from visibility to soil temperature and convective available potential energy are also predicted by different subsets of these WMs. However, many of them are intermittent across the time and space domains so they are not available for wide-scale experimentation. We will, however, consider the persistent ones as potential secondary features. In the following subsections, we present a brief description of each of the WMs considered.

GDPS

The Global Deterministic Prediction System (GDPS)[1] is a WM that is used for global data assimilation and medium range forecasting. It is developed by the Meteorological Service of Canada (MSC) at the Canadian Meteorological Centre (CMC). The version that is used in this study (v8.0) was released in December, 2021. It provides forecasts two times a day for a lead time of ten days with three-hourly increments. The forecasts

[1] `https://collaboration.cmc.ec.gc.ca/cmc/cmoi/product_guide/docs/tech_specifications/tech_specifications_GDPS_e.pdf`

Table 4.1: Input weather models' properties.

WM	Spatial Resolution	Temporal Period	Original Accumulation Period
GDPS	0.125°	24 h	Running total
GEFS	0.125°	24 h	6-hourly
GEPS	0.125°	24 h	6-hourly
GFS	0.125°	24 h	Running total
ICON	0.125°	24 h	Running total
NAM	0.125°	24 h	12-hourly
RDPS	0.125°	24 h	Running total
REPS	0.125°	24 h	Running total

are made on Yin-Yang horizontal grid with a horizontal grid spacing of 0.135 degrees (15 km). It covers a range of variables including precipitation, wind gusts, humidity, cloud cover, temperature, wind speed and wind directions.

GEFS

The Global Ensemble Forecast System (GEFS)[2] is a WM created by the United States National Centers for Environmental Prediction (NCEP), a branch of National Oceanic and Atmospheric Administration (NOAA). It has a horizontal grid spacing of 0.125 degrees (25 km) and a forecast lead time of sixteen days (384 hours) with an output timestep of three hours. The forecasts are made four times a day. In our work, version 12.0 is employed, which was released in September 2020. Its suite of variables include temperature, humidity, wind speed and direction, precipitation and cloud cover amongst others. Unlike GEFS, this WM does not make a single deterministic forecast but rather, probabilistic forecasts based on a range of ensemble members each of which works with a marginally perturbed set of inputs, resulting in a probabilistic distribution to account for the intrinsic uncertainty of the weather conditions. This particular WM uses 30 + 1 ensemble members (one is used for control) and we considered their mean output as the feature, in our study.

[2] https://www.emc.ncep.noaa.gov/emc/pages/numerical_forecast_systems/gefs.php

GEPS

The Global Ensemble Prediction System (GEPS)[3] is another WM developed by the MSC at CMC, Canada. Like GEFS, it is an ensemble WM. It has 20 + 1 perturbed members. Its forecasts have a lead time of sixteen days, and the forecasts are executed two times a day with a timestep of three hours. It has a horizontal grid spacing of 0.35 degrees (39 km). The variables covered by this WM include precipitation, wind speed and direction, temperature and humidity. We used version (v7.0) which was released in December, 2021.

GFS

The Global Forecast System (GFS)[4] is a global WM created by the NCEP of the United States as part of its suite of numerical tools. It is widely used in the meteorological community and provides detailed forecasts of global weather conditions. It produces forecasts four times a day with a lead time of sixteen days and it has a horizontal grid spacing of 13 km. The first five days have one-hourly forecast periods and afterwards, it increases to three hours. We used version 16 which was implemented in March 2021. Its variables include wind gust, temperature, humidity, wind speed and direction, precipitation and cloud cover.

ICON

ICON (short for the Icosahedral Nonhydrostatic model)[5] is a WM developed by the German weather service, Deutscher Wetterdienst (DWD). This is a global model that uses an icosahedral grid, which is a type of grid that is based on a geometric shape with 20 faces, to represent the earth's surface. The actual global grid is finer, comprised of 2,949,120 triangles and it amounts to a mesh size of 13 km. Forecasts are made four times a day, with a lead time of 180 hours for the runs at 00 and 12 UTC. For the 06 and 18 UTC runs, the lead time is 120 hours. For the first 78 hours, output period is one hour after which it increases to three hours. The variable that was available from this WM was total precipitation forecast.

[3] https://collaboration.cmc.ec.gc.ca/cmc/cmoi/product_guide/docs/tech_specifications/tech_specifications_GEPS_e.pdf

[4] https://www.emc.ncep.noaa.gov/emc/pages/numerical_forecast_systems/gfs.php

[5] https://www.dwd.de/EN/research/weatherforecasting/num_modelling/01_num_weather_prediction_modells/icon_description.html

NAM

The North American Mesoscale Forecast System (NAM)[6] is a WM that provides forecasts for the United States, Canada, and Mexico. It is developed by the NCEP of the United States and it uses a high-resolution model to provide detailed forecasts. The NAM model is typically used for short-range weather forecasting and to support decision-making in industries such as aviation, energy, and transportation. This WM runs 4 times a day and makes forecasts with a lead time of 84 hours with a forecast timestep of 3 hours. Its horizontal grid spacing is 12 km. The suite of variables covered by this WM include wind gust, temperature, humidity, wind speed and direction, and precipitation. The timestep is 3 hours.

RDPS

The Regional Deterministic Prediction System (RDPS)[7] is developed by the Canadian Meteorological Centre to produce detailed weather forecasts for Canada and the United States. It operates on a Limited Area Model (LAM) grid with a size of 1108 by 1082 and a horizontal grid spacing of 0.09 degrees (10 km). We used version 8, which was released in December 2021. Predictions are made 4 times a day and the forecast lead time is 84 hours. The timestep for the forecasts is 300 seconds. The set of variables it supports include precipitation, wind gust, humidity, cloud cover, temperature, wind speed and direction.

REPS

The Regional Ensemble Prediction System (REPS)[8] is the ensemble counterpart of RDPS. It is likewise developed by the CMC and it uses the same grid as RDPS, with a horizontal grid spacing of 0.09 degrees (10 km) covering Canada and United States. Version 4 is used, which was released in December 2021. The ensemble consists of 20 + 1 members. It runs 4 times a day with a forecast lead time of 72 hours. The timestep is 300 seconds. Amongst the forecast variables delivered by the model are precipitation, humidity, temperature, wind speed and direction.

[6] https://www.emc.ncep.noaa.gov/emc/pages/numerical_forecast_systems/nam.php

[7] https://collaboration.cmc.ec.gc.ca/cmc/cmoi/product_guide/docs/tech_specifications/tech_specifications_RDPS_e.pdf

[8] https://collaboration.cmc.ec.gc.ca/cmc/cmoi/product_guide/docs/tech_specifications/tech_specifications_REPS_e.pdf

RDPA - Ground Truth Target

We used the Canadian Regional Deterministic Precipitation Analysis System (CaPA-RDPA)[9] from CMC, as precipitation estimates to represent the ground truth for our precipitation forecasts. This system works on the same grid for the RDPS, covering the United States and Canada. The analyses are executed four times a day (00, 06, 12, 18Z), producing estimates for the preceding six-hour window. The grid spacing is 10 km. RDPA version 5.2.0 is used.

4.3 Dataset Acquisition

The primary data source for our research comprises the datasets made available by Weatherlogics Inc. These datasets are invaluable for our study, as they contain rich meteorological information crucial for post-processing NWP precipitation forecasts using machine learning techniques. To access and retrieve the required data securely, we developed a custom python script designed to interact with the AWS S3 infrastructure. This script facilitated the download of raw datasets stored in the S3 bucket, making them accessible for further analysis and processing. The collected data spans a significant time frame of 18 months, starting from December 2021, up to June 2023. This temporal range was selected to encompass a diverse range of weather patterns and conditions for machine learning models to learn.

4.4 Data Preparation and Augmentation

Data preparation is a critical step in preparing the raw datasets obtained from the AWS S3 bucket for subsequent analysis and machine learning model development. The downloaded files from S3 had a specific structure, where each file contained data for multiple weather models, each with different precipitation accumulation periods, in .grib format. This section outlines the steps taken to pre-process the data and organize it for further analysis.

The first step in data pre-processing involved unzipping the downloaded files from the AWS S3 bucket. These files were in a compressed format and needed to be extracted for further analysis. The data files were stored in the GRIB

[9] https://collaboration.cmc.ec.gc.ca/cmc/cmoi/product_guide/docs/lib/technote_capa_rdpa_e.pdf

(Gridded Binary) format, a widely used format for encoding and efficiently storing meteorological data. Each grib file contains a collection of messages, with each message representing a specific meteorological variable (e.g., precipitation, temperature, humidity) along with information about the variable, grid, and time. To access and manipulate data in grib files, we utilized the pygrib library, a powerful Python module designed for handling grib-formatted data. It simplifies the extraction of data from grib files by providing a pythonic interface to interact with grib files without needing to handle the intricacies of the binary format.

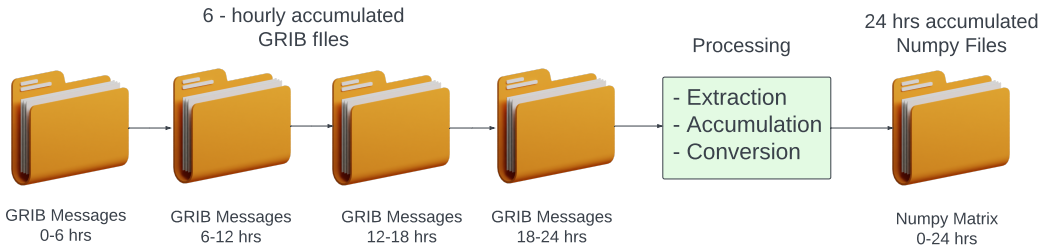

Figure 4.2: Process of consolidating the 24 hourly daily accumulated values

To calculate the 24-hour accumulated precipitation values, we iterated through each date within the dataset, accessing the corresponding grib data files. For each date, we processed data from all the weather models and extracted the relevant precipitation messages. To produce a grid of daily accumulated precipitation, we perform summation across time steps within the 24-hour window as shown in Figure 4.2. In addition to accumulated precipitation, we used pygrib to extract secondary meteorological variables such as temperature, humidity and others, from their respective grib messages. These variables were processed in a similar manner to create 24-hour accumulated values.

Our data acquisition and preprocessing pipeline is summarized in Figure 4.3. For further analysis and machine learning model development, we converted the grib files into a 2D tensor grid with shape $(369, 721)$. For every ML model that permitted data in a tabular format, we created a tensor where each column represented a forecast by an individual WM. This was done by flattening and concatenating daily grid forecasts of respective WMs. In this setup, each row constitutes a data point for a particular location and the rows can be shuffled without regards to latitude, longitude and date. The last column is used for the RDPA ground truth data as the target.

The convolutional neural network architecture utilizes a 3D tensor where, each

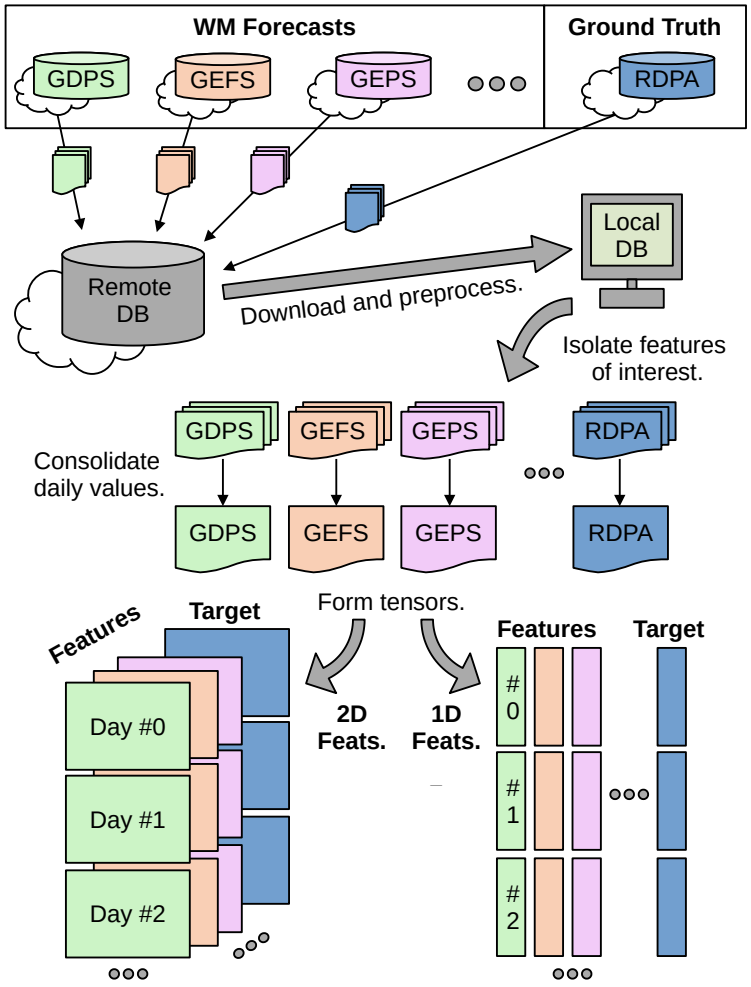

Figure 4.3: Data acquisition & preprocessing pipeline from [1].

slice is the 2D grid forecast by a given WM. In this case, a data sample is composed of the all the 2D grids of 24hr accumulated forecasts by our input WMs as shown in left of Figure 4.4.

Data for the GNN model was generated with PyTorch Geometric, which is a powerful library for handling graph datasets in PyTorch, enabling researchers and developers to work with structured data in the form of graphs efficiently. PyTorch geometric library provides a wide range of tools and utilities for creating custom graph neural network architectures and conducting experiments on graph-based models. It's flexibility, combined with its seamless integration with PyTorch, makes it a preferred choice for anyone working on graph-based deep learning tasks, allowing to harness

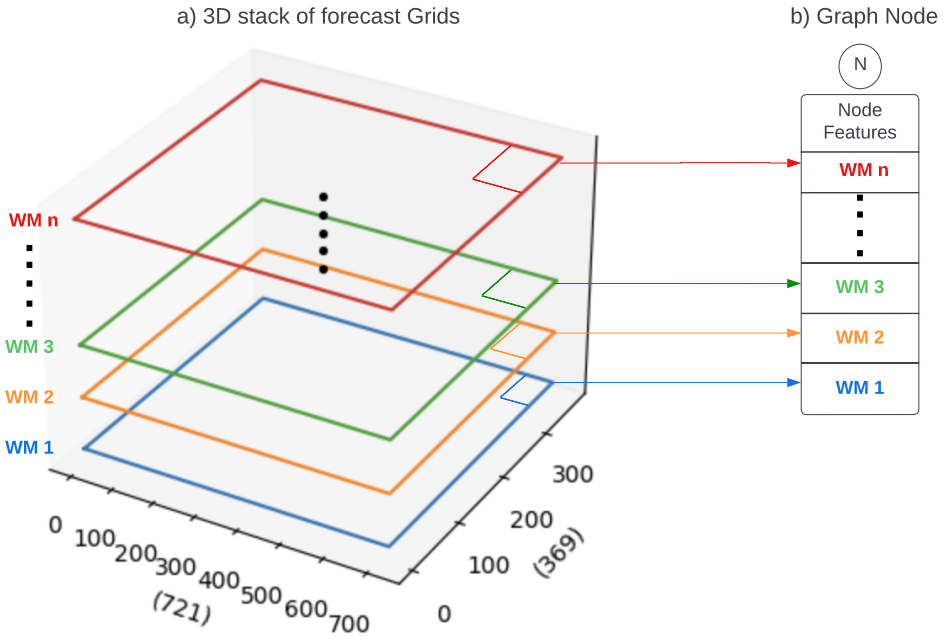

Figure 4.4: Left: 3D tensor made from 2D grid forecast, Right: A single graph node feature derived from the 3D tensor

the full potential of neural networks in the context of structured data represented as graphs. In our case, the node features of a 24hr accumulated graph is defined as the flattened version of 3D tensor used in CNN. A single node feature is made from concatenating a point from a particular location of the grid from each 2D slice as shown in right of Figure 4.4. Bi-directed weighted edges are made between the adjacent points of the 2D grid. The weight of each edge is a normalized inverse of haversine distance [67] D between two points l_1 and l_2 is given by Eqn 4.1. The intuition is to reward datapoints in close proximity to a considered node and punish those points that are farthest from that node.

$$\begin{aligned} I &= \sin^2(\frac{\Delta lat}{2}) + \cos(lat_1) \star \cos(lat_2) \star \sin^2(\frac{\Delta lon}{2}), \\ D &= \frac{\lambda}{r_{earth} \star 2 \star \arctan 2(\sqrt{I}, \sqrt{1-I})}, \end{aligned} \tag{4.1}$$

where, I is an intermediary value, Δlat is the difference in latitude between the two points, Δlon is the difference in longitude between the two points and lat_1 and lat_2

are the latitudes of the two points l_1 and l_2. r_{earth} is substituted by the mean radius of the Earth (6,371 km) and λ is the user-defined normalization hyper-parameter. Through trial and error we found that an optimal value for λ ranges from $4 - 7$. An arbitrary graph network of nodes and edges when overlay over the earth can be visualized in Figure 4.5.

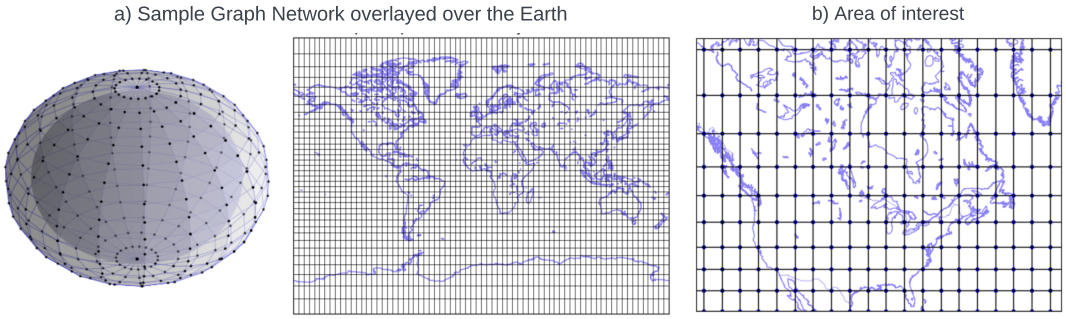

Figure 4.5: Left: A visualization of a sample global graph network, Middle: A flattened 2D graph representation over the Earth, and Right: A flattened graph over a specific region of interest

For all ML models excluding GNN, we have only considered the interaction between two data points either to be same or non-existent. As demonstrated in Figure 4.5 this is clearly not the ideal case when dealing with geometric data, where the curvature of the earth must be incorporated to capture nuanced spatial dependencies. Recent research by Google DeepMind in [44] shows that the 2D grids from various WM's as a graph can help capture these dependencies.

Throughout the experiments, the data set was further cleaned and enhanced as necessary by preprocessing methods such as removing the rows with missing cells, removing the days with missing data or input spectrum normalization, depending on the methodologies used.

4.5 Feature Selection

Alongside daily accumulated precipitation (PR), we also considered twelve secondary features that are present in our data set. The features are summarized in Table 4.2. These include WM-bound features like 2 m air temperature (TM), 2 m relative humidity (RH), U-component of the wind at 10 m above ground (UW), V-component of the wind at 10 m above ground (VW), total cloud cover percentage (CL), wind

gusts at 10 m above ground (GS); the elevation of the ground surface (EL); the spatial features of latitude (LT) and longitude (LN); and the temporal features in forms of varying representations of julian date (JD).

Table 4.2: List of considered input features. Cardinality shows the number of input WMs that can produce this feature. For WM-agnostic geographical and spatio-temporal features it defaults to (1).

Feature	Shorthand	Unit	Cardinality
Daily precipitation	PR	kg / m^2	8
2 metre air temperature	TM	K	7
2 metre relative humidity	RH	%	7
10 metre U wind component	UW	m / s	7
10 metre V wind component	VW	m / s	7
10 metre wind gust	GS	m / s	4
Total cloud coverage	CL	%	4
Elevation	EL	m	1
Latitude	LT	°	1
Longitude	LN	°	1
Julian date	JD	-	1
Julian date (cosine)	Cos(JD)	-	1
Julian date (sine)	Sin(JD)	-	1

Table 4.2 also shows the cardinality of each feature. The primary feature of precipitation is available in every WM, whereas the secondary ones are only available in a subset of them. In any event, because the number of WM-bound features to be considered are proportional to the number of WMs that produce them, experimenting with all of their combinations would prohibitively increase the complexity of the input space. Furthermore, whether they carry the potential to improve the ML models would still have to be assessed, particularly for the secondary features. We therefore opted for a correlation analysis to select the optimal features.

In our previous work, we explored pairwise correlation of RDPA (ground truth) against 123 potential feature form, summarized in Table 4.3. These derived features include the daily mean (average of eight 3-hourly values) and standard deviation of individual WM forecasts as well as the aggregated values of multiple WMs (e.g. cross-model means of daily mean forecasts). Note that daily mean and daily accumulated precipitation are completely correlated so they can be used interchangeably for the purpose of this analysis.

The results of the correlation analysis over a sample set of 3 months is shown

Table 4.3: List of considered feature aggregations.

Aggregation Method	Features to consider (per WM)	# Features
Daily mean	PR, TM, RH, UW, VW, GS, CL	44
Daily std	PR, TM, RH, UW, VW, GS, CL	44
Mean of daily means	PR, TM, RH, UW, VW, GS, CL	7
Mean of daily std	PR, TM, RH, UW, VW, GS, CL	7
Std of daily means	PR, TM, RH, UW, VW, GS, CL	7
Std of daily std	PR, TM, RH, UW, VW, GS, CL	7
Unary (spatial)	LT, LN,	2
Unary (temporal)	JD, cos(JD), sin(JD)	3
Unary (geographical)	EL	1
Gauged Precip.	Ground truth	1
Total		123

Table 4.4: Absolute correlations of 123 feature aggregations against the RDPA ground truth target referred from [1]. Temporal window: (Jan 2022 up to Apr 2022).

Feature	Corr	Feature	Corr	Feature	Corr	Feature	Corr
Ground Truth	1.00	GS std (NAM)	0.34	RH std of mean	0.20	TM std (RDPS)	0.07
PR mean (REPS)	0.94	UW std (GEPS)	0.34	VW mean (GDPS)	0.19	TM std (GEPS)	0.07
PR mean of means	0.93	VW std (NAM)	0.34	VW mean (NAM)	0.19	UW mean (NAM)	0.07
PR mean (RDPS)	0.93	VW mean of std	0.33	VW mean of means	0.19	LT	0.06
PR std (REPS)	0.91	GS mean of means	0.33	VW mean (REPS)	0.19	sin (JD)	0.05
PR mean (GDPS)	0.91	VW std (GEFS)	0.33	VW mean (GEFS)	0.19	UW mean (GFS)	0.04
PR mean (GEPS)	0.91	UW std (REPS)	0.33	VW mean (RDPS)	0.19	UW mean of means	0.04
PR std (RDPS)	0.91	GS mean of std	0.32	VW mean (GFS)	0.19	UW mean (GEFS)	0.04
PR mean of std	0.90	VW std (RDPS)	0.32	CL std (GEFS)	0.19	UW mean (GEPS)	0.04
PR std (GDPS)	0.89	VW std (GDPS)	0.32	TM mean (REPS)	0.18	TM std (GEFS)	0.03
PR mean (GEFS)	0.88	VW std of std	0.32	TM mean (RDPS)	0.18	UW mean (REPS)	0.03
PR mean (ICON)	0.87	GS mean (GFS)	0.31	VW mean (GEPS)	0.18	UW mean (GDPS)	0.03
PR mean (GFS)	0.86	RH mean (GDPS)	0.31	TM mean (GDPS)	0.18	UW mean (RDPS)	0.03
PR std (ICON)	0.85	VW std (GEPS)	0.30	TM mean (GEPS)	0.18	JD	0.03
PR mean (NAM)	0.85	GS mean (RDPS)	0.30	TM mean of means	0.18	cos (JD)	0.02
PR std of std	0.82	GS mean (GDPS)	0.30	TM mean (GEFS)	0.17	CL std (RDPS)	0.02
PR std of mean	0.80	RH mean (RDPS)	0.30	VW std of means	0.16	RH std (RDPS)	0.02
PR std (GEPS)	0.77	VW std (REPS)	0.30	TM mean (NAM)	0.16	RH std (NAM)	0.02
PR std (GEFS)	0.75	GS std (RDPS)	0.29	UW std of means	0.16	RH std (GDPS)	0.02
PR std (GFS)	0.73	RH mean (REPS)	0.29	TM mean (GFS)	0.15	TM std of means	0.02
PR std (NAM)	0.68	GS std (GDPS)	0.29	CL mean of std	0.13	RH std (GEFS)	0.02
CL mean (RDPS)	0.45	UW std of std	0.28	TM std of std	0.13	RH std (GFS)	0.02
CL mean (GDPS)	0.44	CL mean (GEFS)	0.28	RH mean (GEFS)	0.13	RH mean of std	0.01
CL mean of means	0.38	GS std (GFS)	0.28	RH mean (GFS)	0.12	RH std (REPS)	0.00
GS mean (NAM)	0.38	CL mean (GFS)	0.26	CL std of std	0.10	RH std of std	0.00
UW std (NAM)	0.37	RH mean (GEPS)	0.26	EL	0.09	RH std (GEPS)	0.00
UW std (GFS)	0.36	RH mean of means	0.25	LN	0.09	CL std (GDPS)	0.00
UW mean of std	0.36	CL std of mean	0.22	TM std (NAM)	0.09		
UW std (GEFS)	0.36	RH mean (NAM)	0.22	TM std (GDPS)	0.08		
UW std (GDPS)	0.35	GS std of mean	0.21	TM std (REPS)	0.08		
UW std (RDPS)	0.35	CL std (GFS)	0.20	TM mean of std	0.07		
VW std (GFS)	0.34	GS std of std	0.20	TM std (GFS)	0.07		

in Table 4.4. As observed, the most correlated features against the ground truth - RDPA are the different forms of daily precipitation features, dominating the top of

the table, with all of the 8 WMs having a correlation value of more than 0.85 for daily mean.

In our previous work [1], the practical implementation of a proposed approach was one of the highest priority and the improved machine learning approach needed to be resource-light with manageable memory and processing time. In particular, we aimed for solutions which can be deployed on a desktop computer with tensorflow-capable GPU, with a memory consumption of under 16 GB and a processing time of under one hour for training and near real-time for prediction. The early training of the ML models suggested that the secondary features did not result in a major contribution given the limited training data of 4 months (Dec 2021 - Apr 2022). Consequently, in our previous work we only employed 8 daily accumulated precipitation (PR) values as features against our target value of daily precipitation. In this thesis, we extended the use of the same 8 WMs with a larger training dataset of 15 months. Henceforth, we will refer to this expanded preliminary dataset as *Dataset 1*.

Table 4.5: List of features considered in Dataset 1 and Dataset 2.

Dataset 1	Dataset 2
PR (RDPS)	PR (RDPS)
PR (REPS)	PR (REPS)
PR (GEPS)	PR (GEPS)
PR (GDPS)	2 metre air temperature
PR (GEFS)	2 metre relative humidity
PR (GFS)	10 metre U wind component
PR (ICON)	10 metre V wind component
PR (NAM)	PR 10 metre wind gust
	Total cloud coverage
	Elevation
	Latitude
	Longitude

Additionally we will explore the importance and effects of using secondary features with creation of a new dataset, aptly named *Dataset 2*. The aim is to investigate the meteorological features previously left unexplored due to the increased complexity. In Dataset 2, we considered the top three WMs from our correlation analysis and initial testing, which are RDPS, REPS, and GEPS along with the secondary features summarized in Table 4.5. It is worth noting that not all secondary features were available in all of the eight weather models (WMs). Among them, the RDPS WM stands out, as it included all the secondary feature data. Furthermore, the secondary

features from the RDPS model exhibited high correlation with the target values, making it a crucial dataset for our analysis. By capitalizing on the availability of more data, we seek to evaluate the worth of these secondary features in enhancing the predictive capabilities of our machine learning models. Moreover, we have included two powerful machine learning models, XGBoost and GNN, for generating forecasts using Dataset 2. This approach promises to yield valuable insights into the interplay between secondary features and machine learning models, potentially uncovering new avenues for refining our NWP post-processing techniques.

To ensure the validity and robustness of our machine learning models, we divided the collected data into two distinct subsets of training and testing. Data spanning from December 2021 to March 2023, serves as the training data for model development. It encompasses a significant portion of our data and offers the necessary historical context for training. Data with temporal range from March 2023 up to June 2023 has been earmarked for model testing and validation. It offers a more recent dataset for evaluating the model's generalization capabilities and keep the testing season similar to our previous work in [1] for a comprehensive comparison.

Chapter 5

Results and Discussion

In this chapter, we present the outcomes of our experimentation with various machine learning and deep learning models for improving precipitation prediction. Our objective is to assess the performance of each model in predicting precipitation patterns and to gain insights into their strengths and limitations. We employ Dataset 1 in a selection of models, namely IMM, MLR, RFR, GBR, NN, and CNN. Additionally, we leverage the potential of Dataset 2, utilizing it specifically with XGBoost and GNN to explore previously unexplored secondary meteorological features, enhancing our understanding of their predictive capabilities. In the model development process, a two-fold validation approach was employed for parameter tuning. This method involved dividing the dataset into two subsets, allowing for robust validation while optimizing model parameters. The hyper-parameters of the trained ML models can be found in Table A.1 in the appendix section.

This chapter is organized into sections, with each section dedicated to a specific evaluation metric employed in our study, summarized in Table 5.1. Additionally, we provide visualization of some precipitation maps in the appendix in A.1, to compare the predicted precipitation patterns. Similar to our previous work, we studied the results in two separate contexts, spatial and temporal. For the spatial context, we considered our data grid of 369 by 721 cells. For each cell, we combined the results of the validation dates by taking their location wise daily mean, resulting in a 2D map. Then we further consolidated every 14 neighboring cells to create a visually identifiable grid. Fig 5.1 gives a visualization of the mean precipitation for the validation period (March 2023 up to June 2023). For the temporal context, we combined the results across the entire precipitation map of a day and created one daily data point along the time axis.

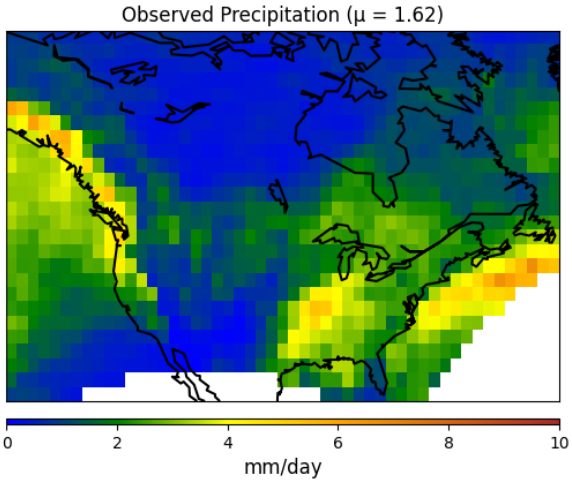

Figure 5.1: Spatial representation of mean precipitation for the validation period (March 2023 up to June 2023).

5.1 Mean Absolute Error (MAE)

MAE measures the average absolute difference between predicted and observed values, providing insights into the models' accuracy. In evaluating the performance of all WMs and trained machine learning models using the Mean Absolute Error (MAE) metric, we observe noteworthy results. Among the input WM, RDPS exhibited the lowest MAE with a value of 0.75 mm/day, followed closely by REPS at 0.76 mm/day. GFS and NAM, on the other hand, exhibited slightly higher MAE values at 1.35 and 1.07, respectively.

Comparing these input WMs models to the baseline Input Means Model (IMM), it is evident that IMM had a MAE of 0.70, demonstrating reasonable accuracy. However, not all the trained machine learning models outperformed IMM in terms of MAE. Notably, the GBR and CNN stood out as the top-performing ML models, achieving MAE values of 0.63 and 0.59 mm/day, respectively. Simple NN's are seen to be the worst performers among other ML models with an MAE of 0.78 mm/day. It is interesting to note that in our previous work [1] NN and CNN's had similar MAE of 0.79 and 0.78 mm/day respectively and GBR with MAE of 0.85 mm/day.

In spatial terms as depicted in Fig 5.4, the models exhibited varying performance across different geographical regions. Generally, lower MAE values were observed in drier zones, particularly in the south-western US and the northern territories. Conversely, higher MAE values were noted in the eastern half of the US, as well as

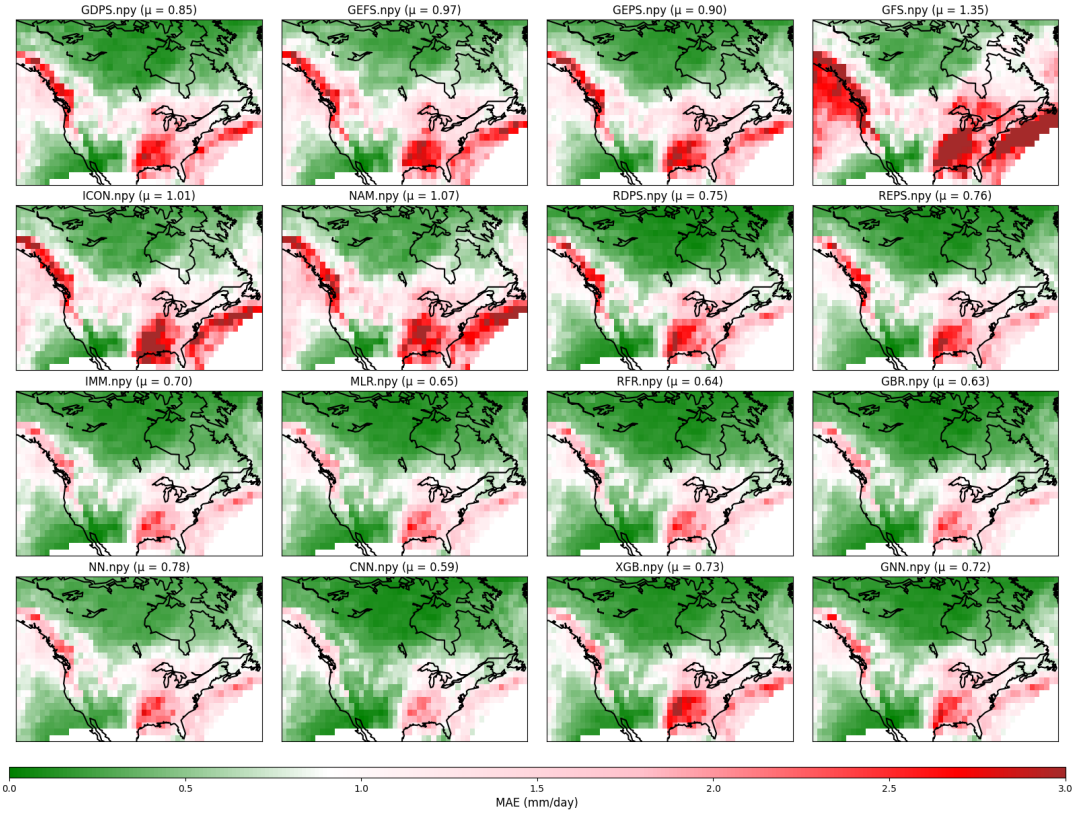

Figure 5.2: Mean absolute error (spatial). Lower is better.

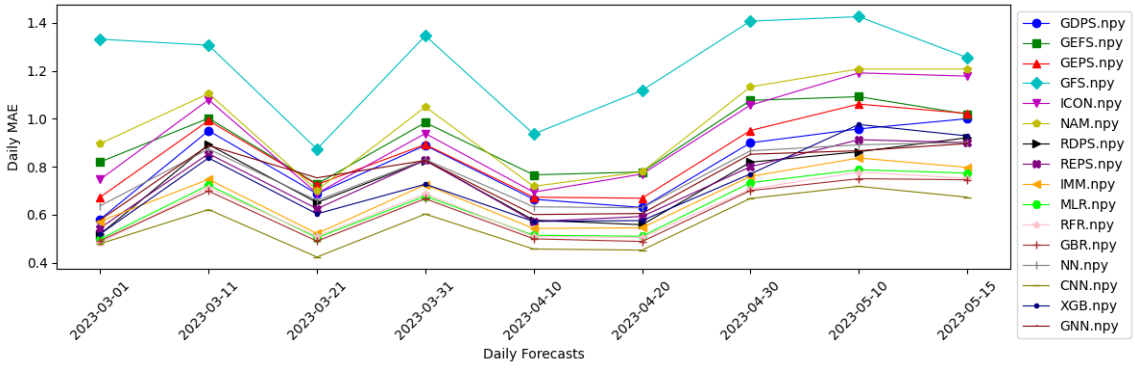

Figure 5.3: Mean absolute error (temporal). Lower is better.

along the Pacific coasts of the northwestern US and British Columbia. Machine learning algorithms introduced improvements, visible in the form of smaller and lighter shades of red on the spatial precipitation distribution map. These results provide an initial glimpse into the models' performance based on MAE, paving the way for further analysis of temporal trends shown in Fig 5.3.

We observe a similar pattern emerged as the spatial performance where, on most days, the trained ML models outperformed the input WMs. CNN can be clearly seen having the lowest MAE, closely followed by GBR. Once again, GFS emerged as the poorest model, followed by NAM and ICON.

5.2 Root Mean Squared Error (RMSE)

RMSE measures the square root of the average squared difference between predicted and observed values, providing insights into the models' accuracy in predicting variability. It is more sensitive to larger errors than MAE. In assessing the performance of both the machine learning models and weather models using the RMSE metric, referring Fig 5.4, we find significant variations in the results. Among the input WM, RDPS and REPS exhibited the lowest RMSE values, at 1.64 and 1.83 mm/day, respectively. IMM demonstrated a RMSE of 1.53 mm/day, indicating reasonable accuracy. All of the trained machine learning models consistently outperformed the input WM in terms of RMSE.

CNN stood out as the top-performing ML models for this metric, achieving RMSE values of 1.45 mm/day, followed by GBR with RMSE 1.46 mm/day. MLR and RFR also demonstrated strong performance with a RMSE of 1.48 mm/day. In contrast, NN, GNN, and XGBoost displayed slightly higher RMSE values ranging from 1.57 to 1.63 mm/day. Previously, CNN and NN had similar performance with RMSE of 1.82 mm/day, and more importantly GBR performed worse with RMSE of 1.88 mm/day.

The spatially distribution of errors is similar to MAE varying performance across different geographical regions. Temporally, we see the machine learning approaches lowering the RMSE as before, with the gradient boosted regression giving the lowest RMSE and CNN following closely behind shown in Fig 5.5. For some days XGBoost does show signs for reducing the overall RMSE.

5.3 Median Absolute Error (MdAE)

MDAE, which measures the median of absolute difference between predicted and observed values, provides valuable insights into the models performance in predicting the weather without getting influenced from outliers. Among the input weather models, RDPS stands out as the top-performer in terms of MDAE with an impressively low value of 0.13 mm/day. GDPS also demonstrate strong performance

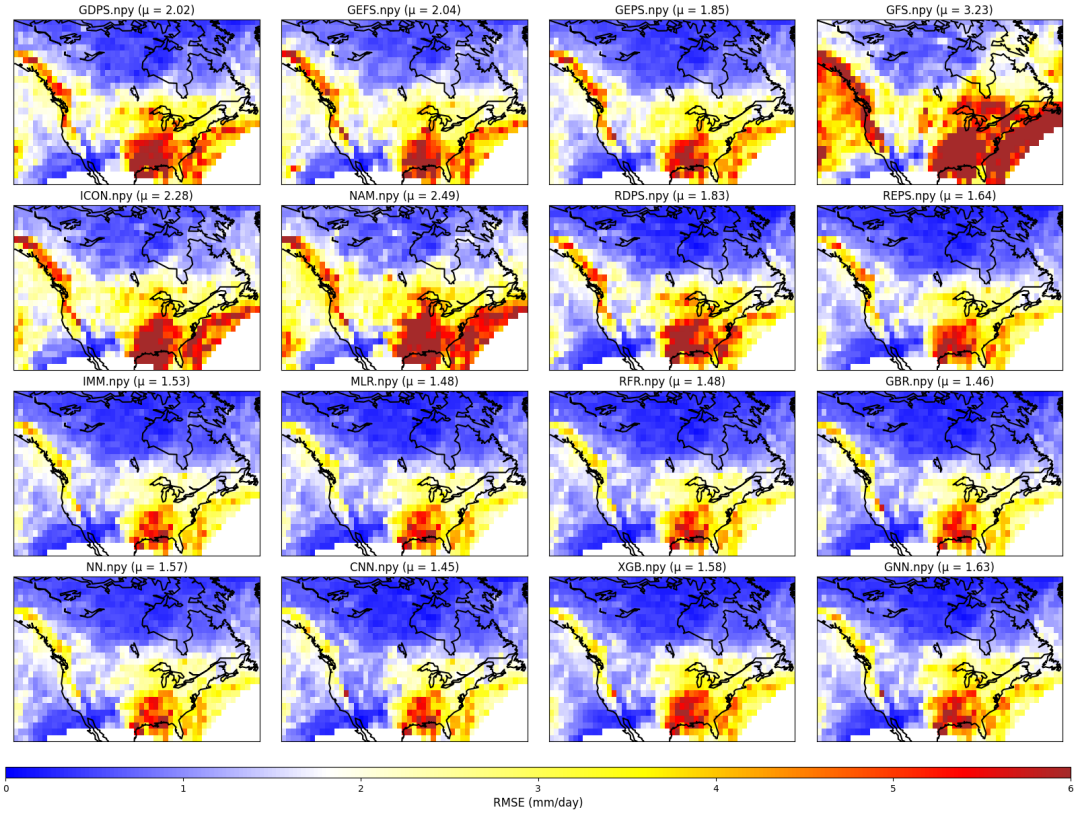

Figure 5.4: Root mean squared error (spatial). Lower is better.

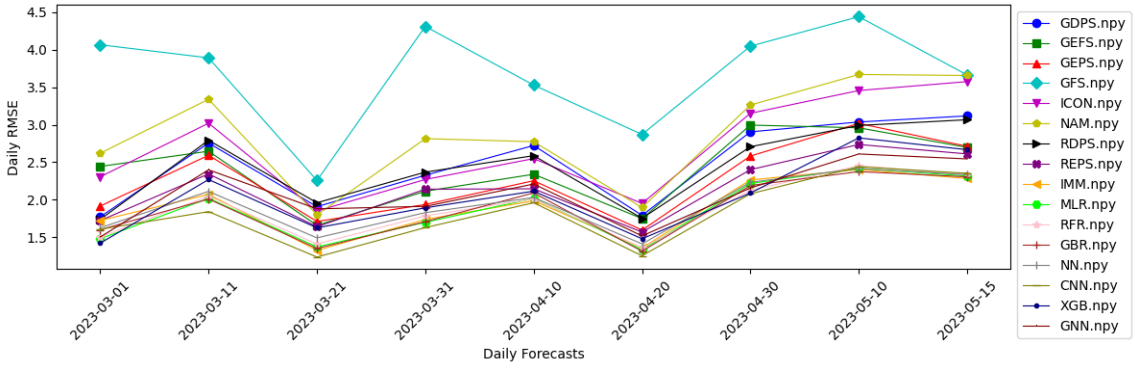

Figure 5.5: Root mean squared error (temporal). Lower is better.

with MDAE values of 0.17 mm/day. In contrast, some input WMs such as GEFS, GEPS, GFS, ICON, and NAM exhibit relatively higher MDAE values ranging from 0.24 to 0.32 mm/day, indicating room for improvement in their predictions.

Comparing the ML models to the baseline IMM with an MDAE of 0.20 mm/day, we can see in Fig 5.6 that several machine learning models have managed to

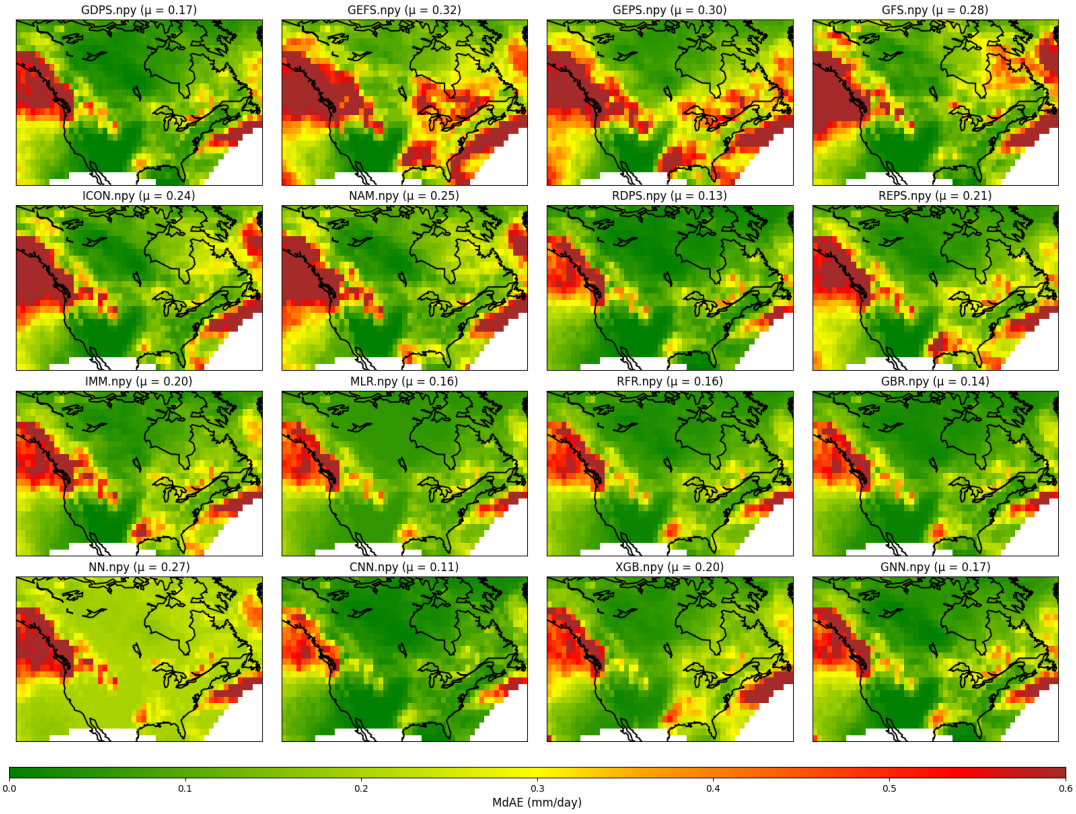

Figure 5.6: Median absolute error (spatial). Lower is better.

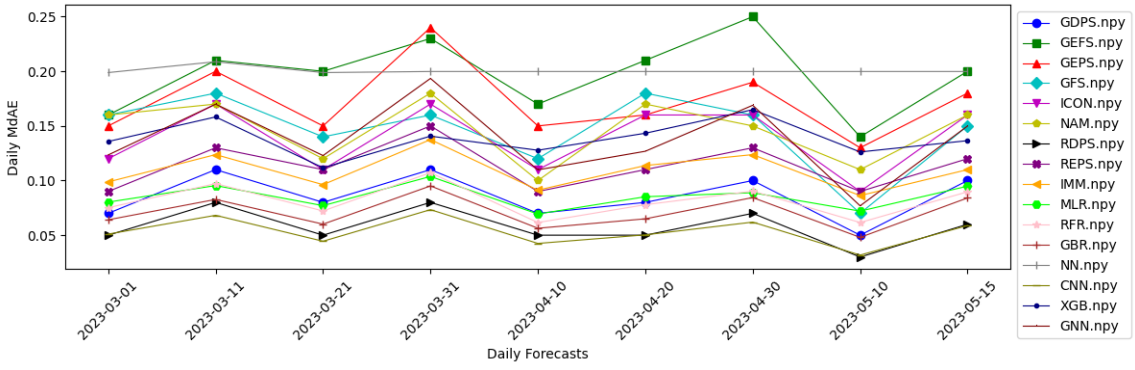

Figure 5.7: Median absolute error (temporal). Lower is better.

outperform IMM. Notably, CNN stands out as the top-performing machine learning model with an impressively low MDAE of 0.11 mm/day, highlighting its capability to predict weather accurately. On the other hand, NN exhibits a higher MDAE of 0.27 mm/day, indicating a relatively weaker performance compared to other machine learning models. Spatially high errors are concentrated in the north west and south

east. Some input WM have high errors in the lower middle, but these errors are comfortably reduced by CNN and GNN models.

Temporal results shown in Fig 5.7, align with the spatial results. CNN being the best machine learning performer and RDPS being the best WM. NN is the worst overall performing machine learning model followed by GEFS WM. Our previous work shows that CNN and NN had a impressive performance both spatially and temporarily, but NN fails to show much improvement this time.

5.4 Maximum Error (MaxE)

MaxE, which measures the maximum absolute difference between predicted and observed values, provides insights into the models performance under extreme conditions. Evaluating the performance of both machine learning models and input WMs using the MaxE metric reveals notable findings. Among the input weather models, REPS demonstrated the lowest averaged MaxE with a value of 8.62 mm/day, followed closely by GEPS at 9.50 mm/day. On the other hand, GFS exhibited the highest averaged MaxE at 17.47 mm/day, indicating the model's challenges in capturing extreme conditions.

We do note that there is a visible correlation of MaxE with the observed precipitation, showing that high errors come from wet zones of lower east coast and the low errors come from dry zones. As shown in Fig 5.8 the baseline IMM, had a MaxE of 8.22 mm/day, demonstrating reasonable performance under extreme conditions. When considering machine learning models, the GBR and XGBoost stood out as the top-performing ML models, achieving MaxE values of 8.03 and 8.08 mm/day, respectively. RFR, MLR and NN also exhibited promising performance with MaxE values ranging from 8.10 to 8.18 mm/day. Interestingly CNN and GNN had the highest maximum errors of 8.22 and 8.63 mm/day respectively. In our preliminary study maximum errors of GBR, RFR and CNN were among the highest among the trained ML models.

Temporally, it can be observed in Fig 5.9 that RDPS and GFS show a higher error value on most days. The trained ML models provide the low error across the daily forecast dates, with XGboost and GNN performing slightly better on a few days with higher overall precipitation.

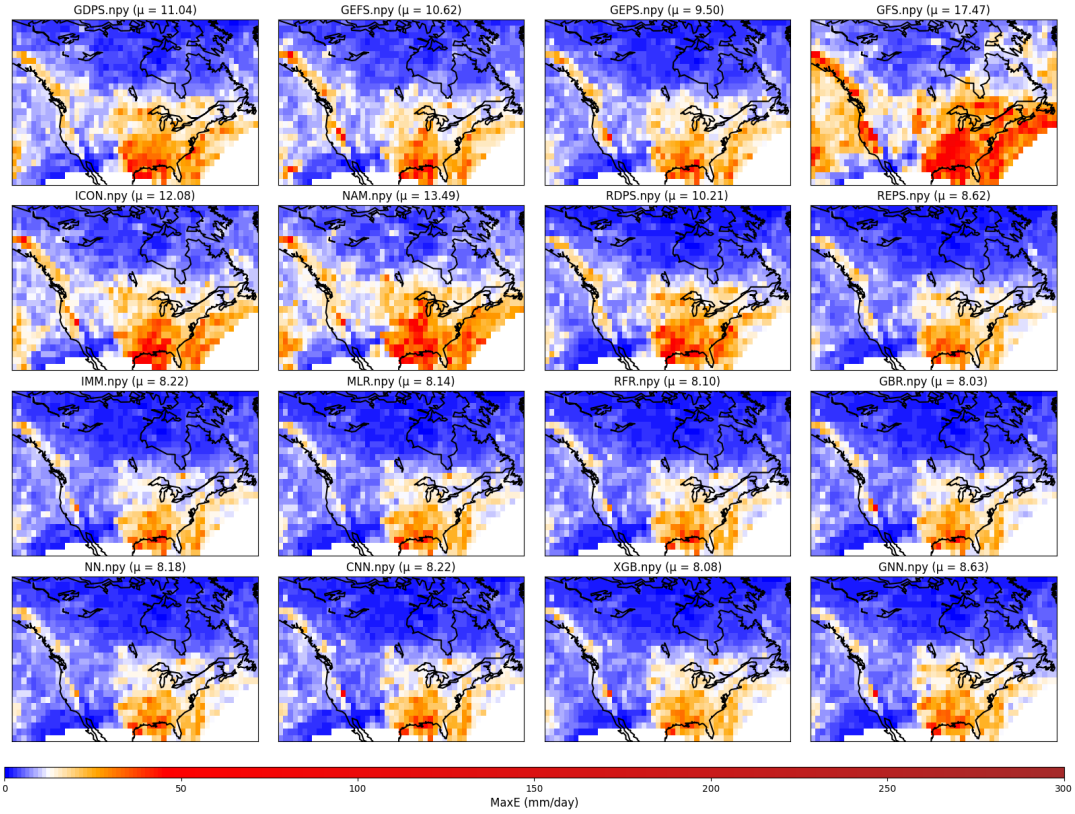

Figure 5.8: Maximum error (spatial). Lower is better.

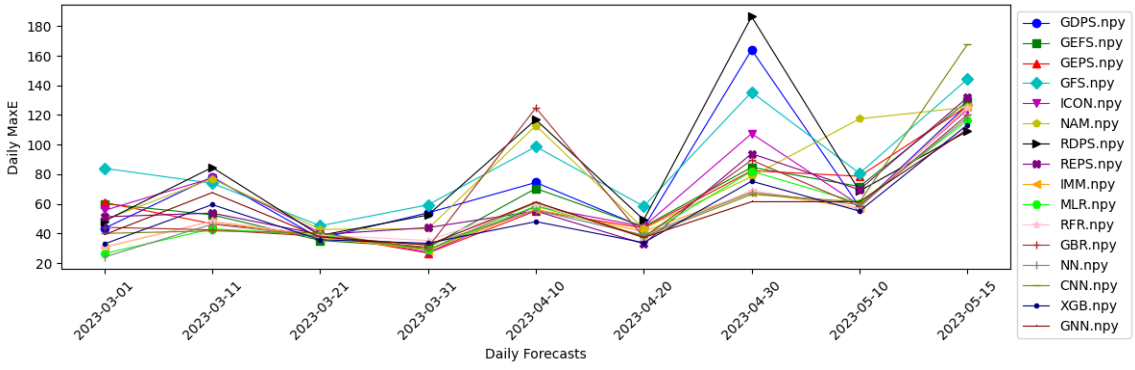

Figure 5.9: Maximum error (temporal). Lower is better.

5.5 Correlation Coefficient (CC)

The Correlation Coefficient is a vital metric for assessing the degree of linear relationship between predicted and observed values. In this evaluation of machine learning models and input WM, we present the results of this metric, shedding light

on the models' ability to capture linear dependencies. Among the input WM, REPS exhibited the highest CC with a value of 0.91, closely followed by RDPS at 0.90. On the other hand, GFS and NAM exhibited comparatively lower CC values at 0.50 and 0.78, respectively.

The baseline IMM showed a CC of 0.91, indicating a reasonable degree of linear relationship. Interestingly, among the trained machine learning models, GBR and CNN emerged as the top performers, achieving CC of 0.92 each. Notably, the majority of machine learning models, including MLR, RFR, XGBoost, NN, and GNN, maintained Correlation Coefficients of 0.91, demonstrating consistent performance similar to our previous work.

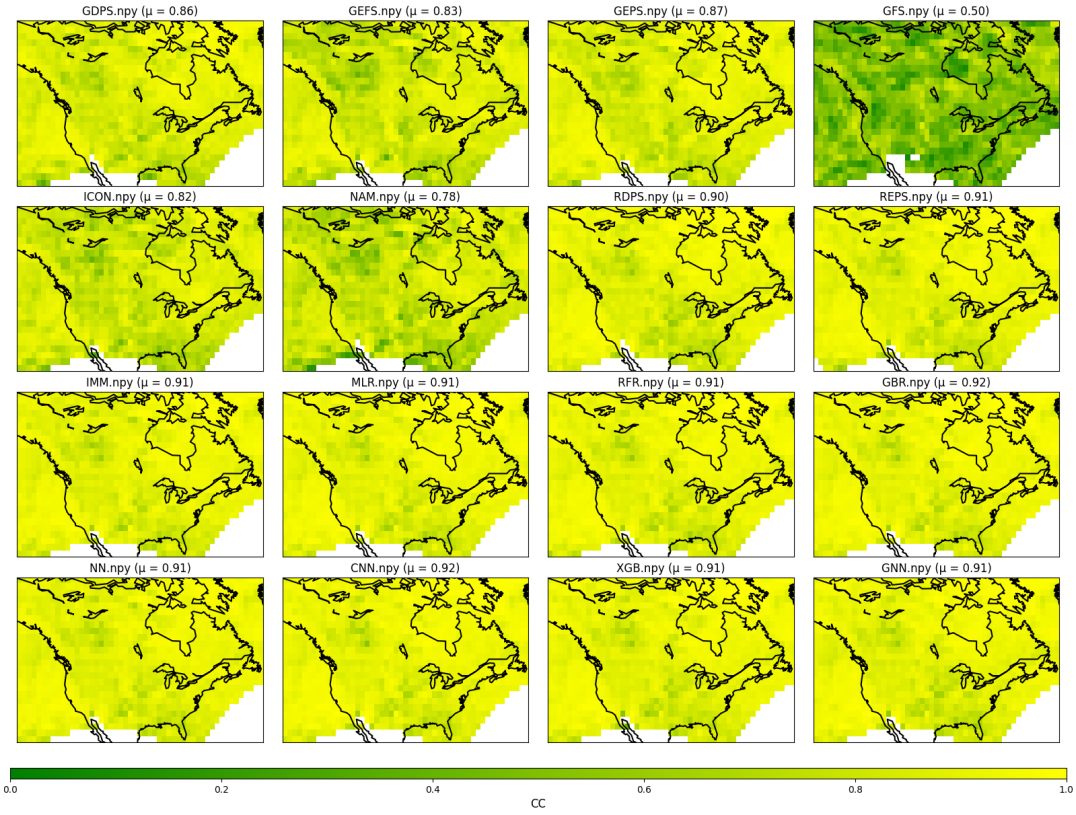

Figure 5.10: Correlation coefficient (spatial). Higher is better.

In spatial terms we can see in Fig 5.10, machine learning algorithms introduced improvements, particularly in regions where correlation coefficients were initially lower, as indicated by the shifting to higher values on the spatial precipitation distribution map. It appears the lowest correlation is observed at the upper and lower center of the map. Temporarily their is relatively low cross-model variation

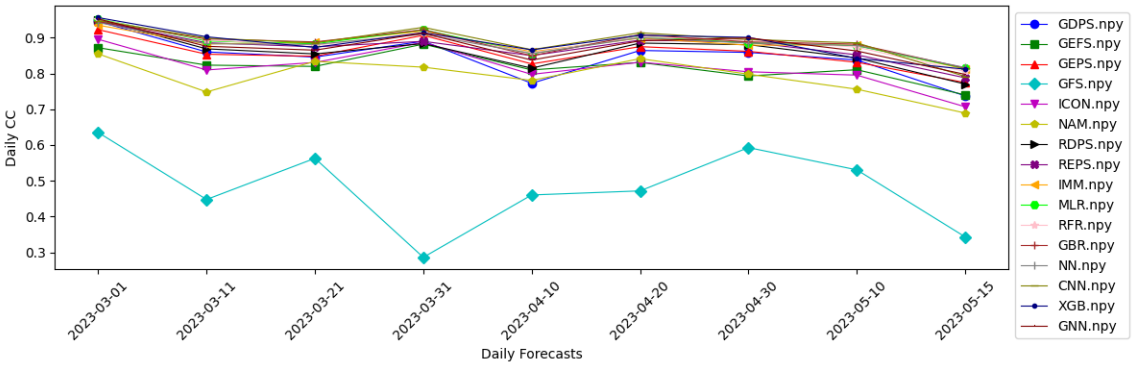

Figure 5.11: Correlation coefficient (spatial). Higher is better.

and all ML models perform better than the input WM as depicted in Fig 5.11.

5.6 Relative Bias (RB)

Relative Bias is a crucial metric for assessing the models' ability to predict the direction of the observed values. It measures the average relative difference between predicted and observed values and offers valuable insights into bias tendencies. Among the input weather models, GFS exhibited the most noticeable negative bias with a Relative Bias value of -0.64. ICON, GDPS, RDPS, and REPS had relatively modest positive biases, with values ranging from 0.18 to 0.27.

By observing Fig 5.12, IMM displayed a relatively low positive bias of 0.14, suggesting a slight overestimation of precipitation. Among the trained machine learning models, GBR and CNN showed the lowest biases, with values of 0.11 and -0.07, respectively. It is worth noting that CNN displayed a negative bias, indicating an inclination to under predict precipitation. Spatially, it can be seen that most of the input WM over predict the lower west coast denoted by blue color. Interestingly, XGBoost has both high and low bias extreme in that region. GNN similarly to CNN is also able to reduce high bias in the same region.

Temporal investigation reveals a similar result as shown in Fig 5.13. Again, trained machine learning models (GBR, NN and CNN) perform significantly better than the rest of the models.

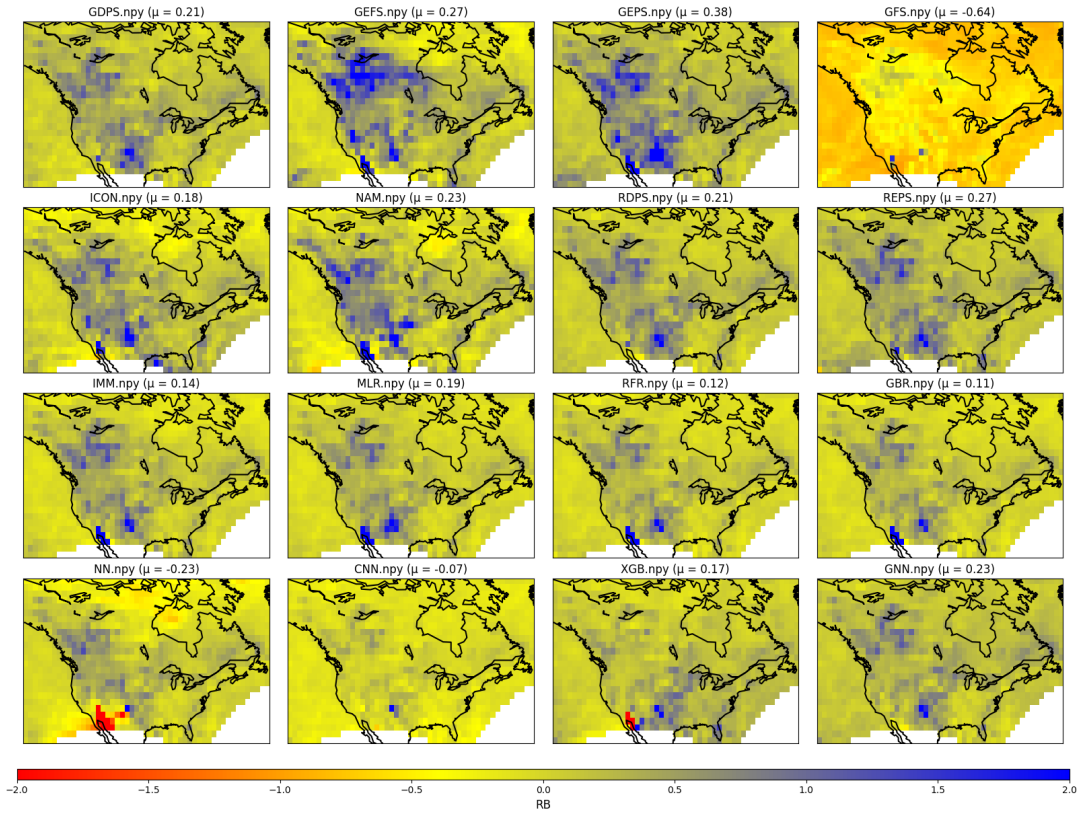

Figure 5.12: Relative bias (spatial). The closer to zero, the better.

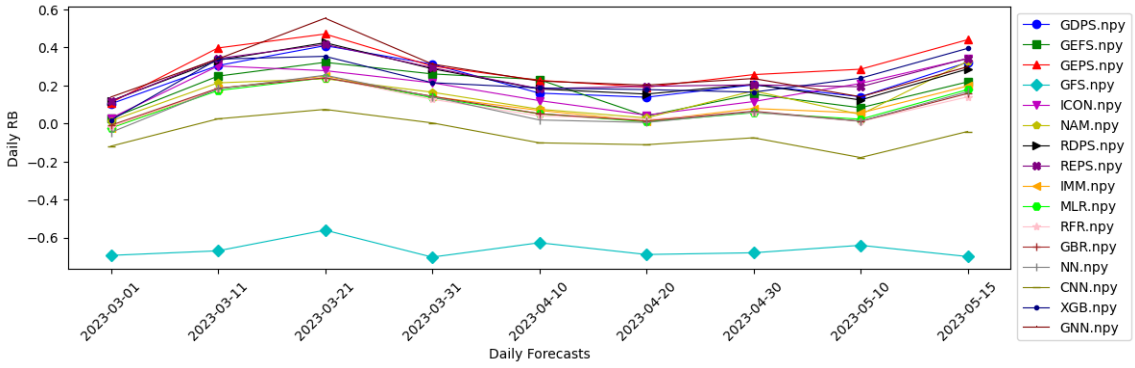

Figure 5.13: Relative bias (spatial). The closer to zero, the better.

5.7 Probability Of Detection (POD)

Probability of Detection (POD) measures the ability of models to correctly predict the occurrence of an event. In Fig 5.14 it can be seen that among the input WMs, REPS demonstrated the highest POD with a score of 0.81, closely followed by GEPS

and RDPS at 0.78. On the other hand, GFS had a lower POD at 0.33, indicating challenges in correctly predicting certain events.

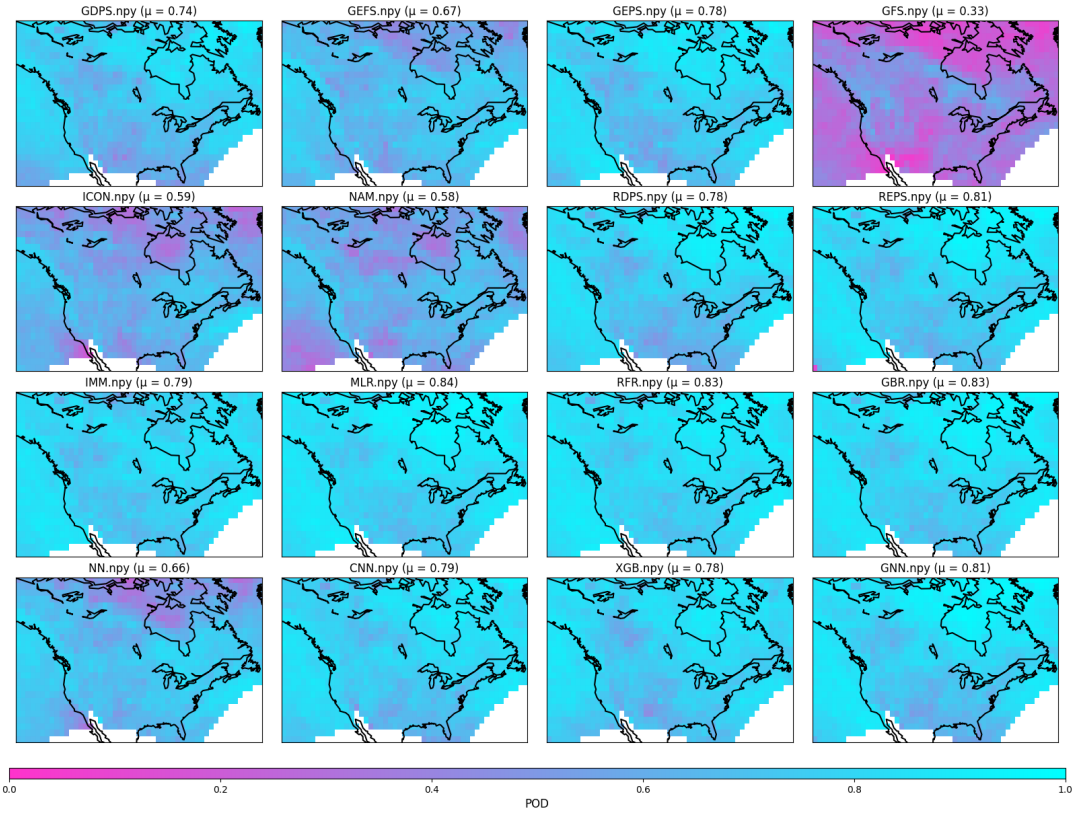

Figure 5.14: Probability of detection (spatial). Higher is better.

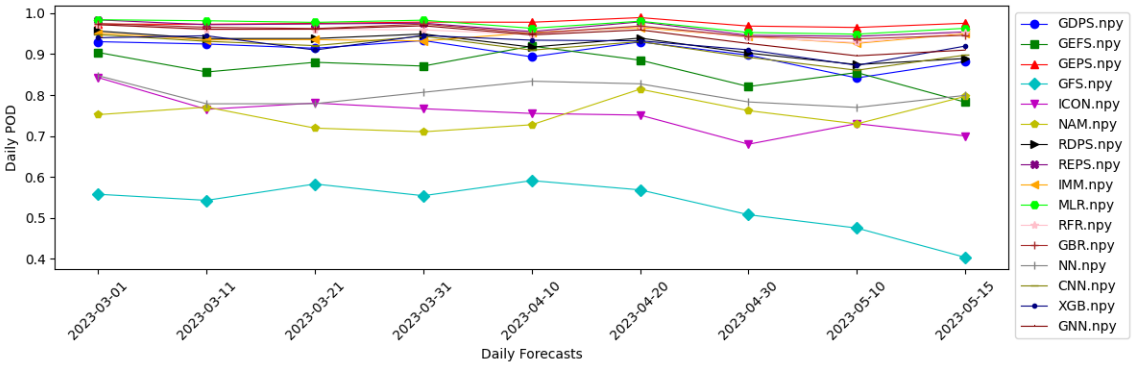

Figure 5.15: Probability of detection (spatial). Higher is better.

Comparing these trained ML models to the baseline IMM with a POD of 0.79, we observe variations in performance. Interestingly, not all the trained machine learning

models outperformed IMM in terms of POD. Notable standouts include MLR, RFR, GBR and GNN, all achieving a POD above of 0.81, showcasing their capability in event detection. Convolutional neural network (CNN) also displayed strong performance, with a POD of 0.79. Meanwhile, neural networks (NN) demonstrated a lower score, which was also the case in our previous set of experiments.

Fig 5.15 shows the temporal results. In temporal terms, GEPS once again achieved the highest result, with REPS as the second best. These WMs are followed by the remaining trained ML models. The POD for most of the ML models look similar but their is a noteworthy difference between NN and the rest of the models.

5.8 False Alarm Ratio (FAR)

The False Alarm Rate (FAR) metric is instrumental in evaluating the accuracy of identifying false positive predictions in the machine learning models and the input WM. In the spatial context as depicted in Fig 5.16, the FAR metric reveals interesting insights into the models' performance. Among the input WM, RDPS exhibited the lowest FAR with a value of 0.22, closely followed by GDPS at 0.26. Other WMs such as ICON, REPS, and NAM also showed competitive FAR values ranging from 0.31 to 0.34. On the other hand, GFS demonstrated a higher FAR of 0.47, indicating a greater tendency for false alarms.

Comparing input WMs to the baseline IMM, it is notable that IMM had a FAR of 0.30, suggesting a moderate ability to identify false alarms. Among the machine learning models, CNN stands out as the top performer with a remarkably low FAR of 0.19, while GNN and GBR also delivered impressive results with a FAR of 0.24. GBR and CNN are able to reduce a cluster of high false alarm regions observed around the Canadian Prairies in all input WM's. The remaining ML models showed FAR values in the range of 0.25 to 0.35, demonstrating varying degrees of accuracy in detecting false alarms.

Fig 5.17 shows the temporal performance. We see CNN is the lowest by a comfortable margin, followed by RDPS and NN. The temporal graph shows that NN does a better job than GBR reducing false alarms, and MLR and XGboost are the worst performers.

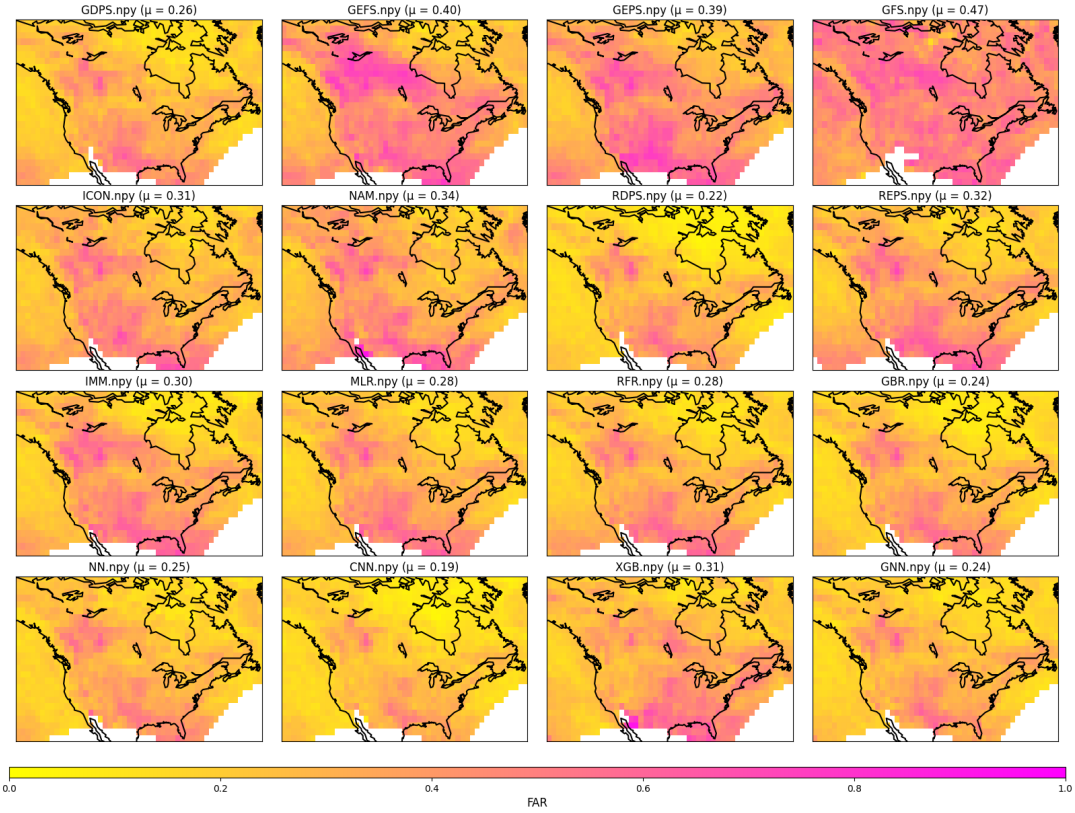

Figure 5.16: False alarm ratio (temporal). Lower is better.

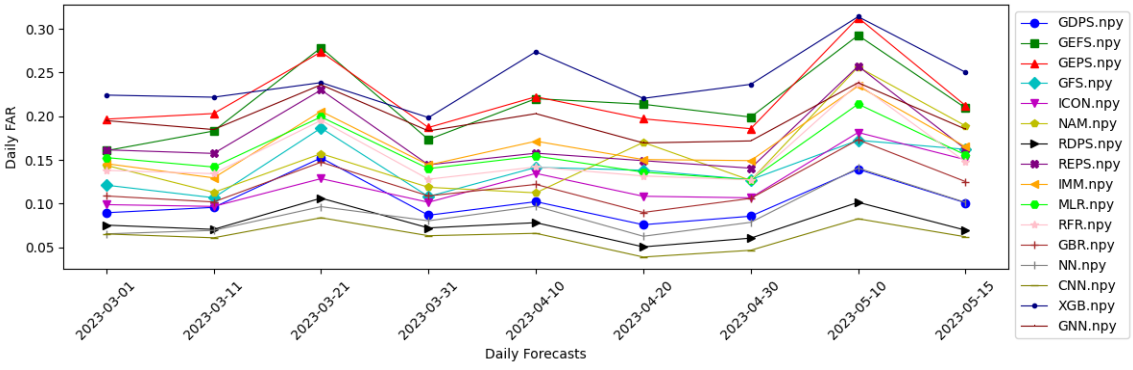

Figure 5.17: False alarm ratio (temporal). Lower is better.

5.9 Critical Success Index (CSI)

The Critical Success Index (CSI) metric assesses the accuracy of model predictions by considering both hits (correct positive predictions) and misses (observed positive events not predicted). Spatially, among the input WM, RDPS exhibited the highest

CSI score, with a value of 0.66, indicating its proficiency in capturing critical weather events. It was closely followed by REPS at 0.60 and GDPS at 0.59, showcasing their reliability in predicting events accurately.

As shown in Fig 5.18 the baseline IMM showed a CSI score of 0.60, signifying reasonably accurate predictions. Comparatively, the machine learning models also showed competitive performance, with CNN achieving the highest CSI at 0.67 followed by GNN and GBR with CSI 0.66, indicating a significant ability to predict critical weather events successfully. NN and XGBoost scored 0.54 and 0.59, respectively, indicating their moderate success in capturing critical events.

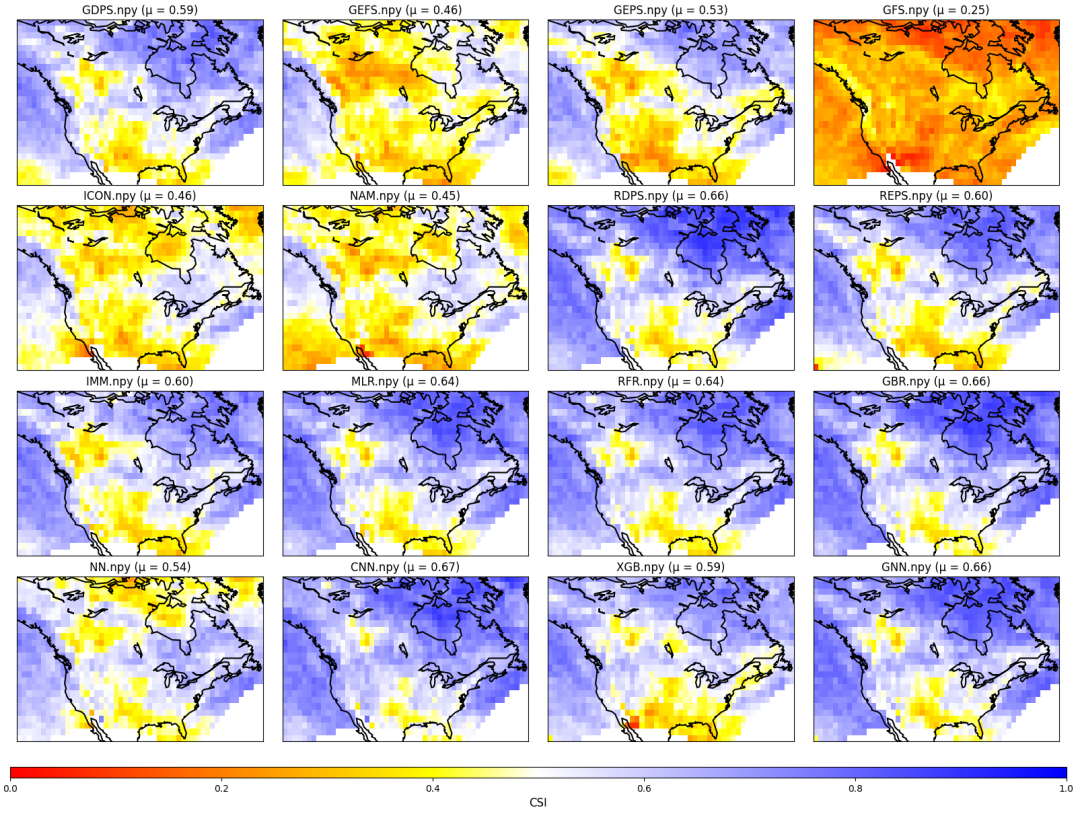

Figure 5.18: Critical score index (temporal). Higher is better.

In temporal analysis shown in Fig 5.19, similar patterns were observed, CNN, RDPS and GBR are the top performers. GFS, MLR, NAM and ICON can be categorized as poor performers. Baseline IMM and the remaining models achieved decent critical score index values.

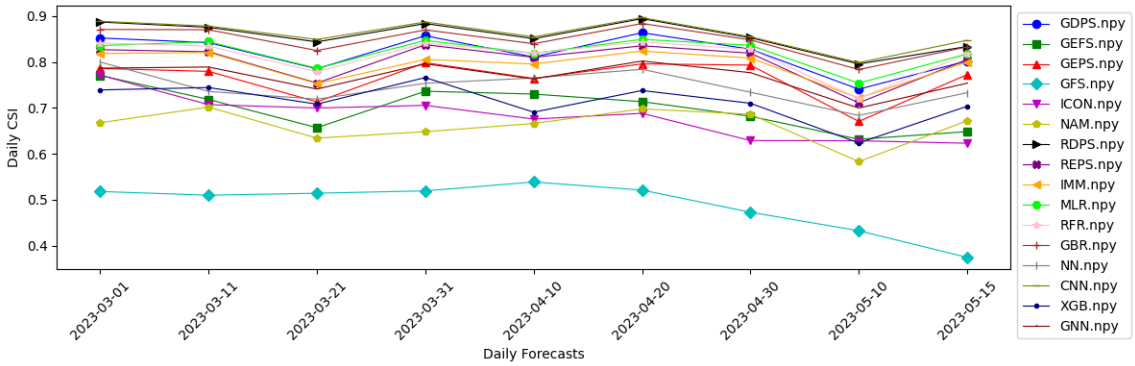

Figure 5.19: Critical score index (temporal). Higher is better.

5.10 Confusion Matrix (CM)

A confusion matrix is a tabular representation indicating the number of true positive, true negative, false positive, and false negative predictions to assess the model's accuracy and error rates. Inspired by [68] and information from domain experts, we *bin* the predicted and observation data into 4 precipitation (PR) levels. These bins are labelled as following: *Nil*: $PR < 0.20mm/day$, *Light*: $0.20mm/day <= PR < 5mm/day$, *Moderate*: $5mm/day <= PR < 20mm/day$ and *Heavy*: $20mm/day <= PR$.

In the Fig 5.20 we can see how the top performing machine learning models along with our baseline IMM classify the precipitation into 4 categories. In the appendix in A.3 we share confusion matrix of MLR and RFR models. For the *Nil* category, CNN achieved the highest score, correctly identifying days with no precipitation, while NN and GBR also demonstrated strong performance. In the *Light* precipitation category, MLR, GBR and GNN outperformed the other models, displaying sensitivity to light precipitation events. For the *Moderate* category, NN exhibited superior precision closely followed by XGBoost, minimizing false alarms, which is crucial in scenarios requiring accurate moderate precipitation forecasts. Lastly, the most sensitive and difficult to predict *Heavy* category, XGBoost and GNN showcased a robust ability to predict heavy precipitation events, suggesting its utility in applications demanding early detection of extreme weather conditions.

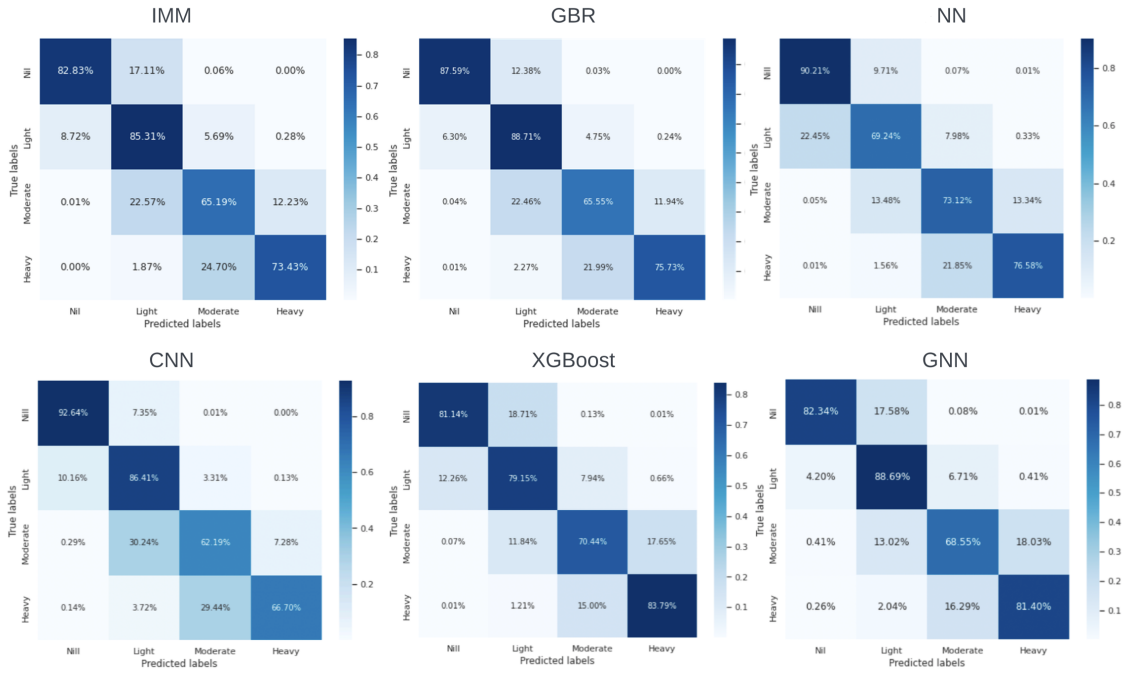

Figure 5.20: Confusion matrix of various ML models

5.11 Discussion

The results of our experiments provide valuable insights into the strengths and weaknesses of various machine learning models. The results indicate that CNN and GBR exhibit robust performance across various evaluation metrics, showcasing their versatility and reliability in post-processing NWP models. Interestingly, the performance of GBR has notably improved from our previous study, showcasing its adaptability to the larger dataset. However, it is worth noting that each model has its own set of strengths and weaknesses, making them suitable for different applications based on specific requirements.

In terms of Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), we observed that the Gradient Boosting Regression (GBR) and Convolutional Neural Network (CNN) emerged as top performers, displaying lower MAE and RMSE values. This indicates their proficiency in accurately predicting daily precipitation.

When assessing the models' ability to predict extreme conditions (MaxE), CNN excelled again, outperforming other models. In terms Correlation Coefficient, GBR and CNN again stood out, showcasing their capability to establish strong relationships between predicted and observed values.

Table 5.1: Averaged results for the entire testing dataset. Best values are highlighted. The top 8 rows are input WMs. IMM is the baseline model.

	MAE	RMSE	MdAE	MaxE	CC	RB	POD	FAR	CSI
GDPS	0.85	2.02	0.17	11.04	0.86	0.21	0.74	0.26	0.59
GEFS	0.97	2.04	0.32	10.62	0.83	0.27	0.67	0.40	0.46
GEPS	0.90	1.85	0.30	9.5	0.87	0.38	0.78	0.39	0.53
GFS	1.35	3.23	0.28	17.47	0.50	-0.64	0.33	0.47	0.25
ICON	1.01	2.28	0.24	12.08	0.82	0.18	0.59	0.31	0.46
NAM	1.07	2.49	0.25	13.49	0.78	0.23	0.58	0.34	0.45
RDPS	0.75	1.83	0.13	10.21	0.90	0.21	0.78	0.22	0.66
REPS	0.76	1.64	0.21	8.62	0.91	0.27	0.81	0.32	0.60
IMM	0.70	1.53	0.20	8.22	0.91	0.14	0.79	0.30	0.60
MLR	0.65	1.48	0.16	8.14	0.91	0.19	**0.84**	0.28	0.64
RFR	0.64	1.48	0.16	8.10	0.91	0.12	0.83	0.28	0.64
GBR	0.63	1.46	0.14	**8.03**	**0.92**	0.11	0.83	0.24	0.66
NN	0.78	1.57	0.27	8.18	0.91	-0.23	0.66	0.25	0.54
CNN	**0.59**	**1.45**	**0.11**	8.22	**0.92**	**-0.07**	0.79	**0.19**	**0.67**
XGB	0.73	1.58	0.20	8.08	0.91	0.17	0.78	0.31	0.59
GNN	0.72	1.63	0.17	8.63	0.91	0.23	0.81	0.24	0.66

The analysis of Relative Bias (RB) yielded intriguing results, showcasing distinct bias tendencies within the CNN and GBR models. CNN exhibited a slight negative bias, while GBR demonstrated a modest positive bias, implying a tendency for underprediction and overprediction of precipitation, respectively. MLR showcased a noteworthy Probability of Detection (POD), reflecting its aptitude for accurately forecasting precipitation events. Additionally, False Alarm Ration (FAR) and Critical Success Index (CSI) demonstrated that CNN, GNN and GBR excelled in accurately predicting critical weather events with low false alarms.

The confusion matrix analysis provides further insights into the abilities of the machine learning models to correctly predict different precipitation levels. While CNN and GBR demonstrated a overall performance, they were unable to repeat the same performance with higher precipitation levels. On the other hand, XGBoost and GNN perfromed exceedingly well in predicting heavy and moderate precipitation even though they were not as effective as CNN in terms of most of the discussed metrics.

The exceptional overall performance of CNN and GBR, can be attributed to their incredible ability to predict lower levels of precipitation. This proficiency along with the substantial prevalence of days with low precipitation values in the dataset contributes to reduced metrics like MAE and RMSE. On the other hand NN, XGBoost

and GNN display a balance in their predictive capabilities. In case of XGBoost and GNN this might be due to the presence of secondary meteorological features, which other models do not have access to. We can analyze this theory by studying the shapley values, these values help to understand how the model's predictions are affected by the presence or absence of specific features for a given instance.

5.11.1 Shapley Analysis

The concept of Shapley values, a pivotal development in the field of cooperative game theory, can be traced back to the pioneering work of Lloyd Shapley in 1953 [13]. Shapley values, are often used to explain the contribution of individual features or elements within a dataset to the output of a machine learning model. They provide insights into the importance of each feature in the decision-making processes. Below we have discussed Shapley values for GBR and NN with Dataset 1, as well as XGBoost with Dataset 2.

For a comprehensive understanding, we randomly sampled 500 data points from each of the four previously mentioned precipitation levels in both datasets. We then assessed the performance of GBR, NN, and XGBoost models using the shap python library. Shap library provides multiple visualization of shap values out which we selected the shap summary plot. The results of shap summary plot for different levels of precipitation are shown below, and are easy to interpret. The horizontal position of points tells us about SHAP values, showing how they affect the model's predictions. The color of these points indicates whether a feature value is high (in red) or low (in blue). For instance, if we take a look at Figure 5.21, when the rdps variable in an GBR model has low values (colored in blue), it tends to push the prediction lower. Conversely, when the rdps variable has high values (colored in red), it tends to push the prediction higher.

For *Nil* and *Light* precipitation values shown in Fig 5.21 and Fig 5.22, we can observe that GBR and NN place significant reliance on a small number of WM's when there are substantial deviations in their prediction values. This indicates that GBR and NN are highly dependent on the performance of specific input weather variables. In contrast, XGBoost appears to distribute its reliance more evenly across various secondary meteorological features when making decisions.

For *Moderate* and *Heavy* precipitation values shown in Fig 5.23 and Fig 5.24, the NAM weather model takes on growing importance for both the GBR and NN.

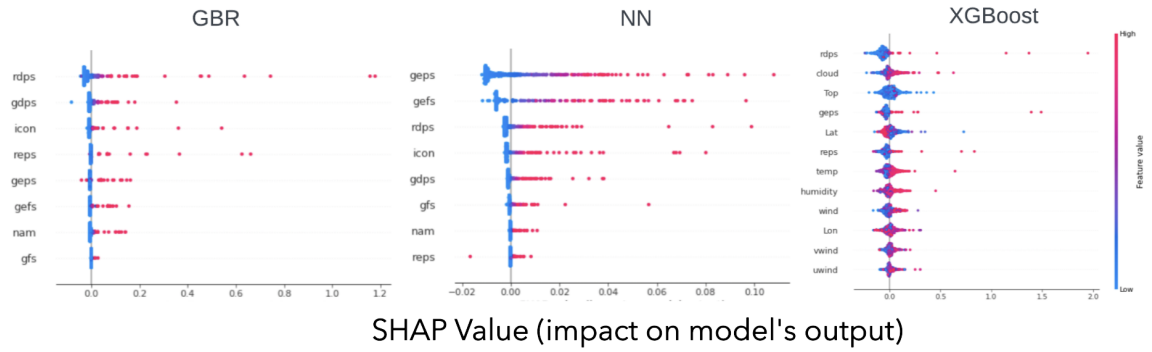

Figure 5.21: Shapley values for randomly sampled Nil precipitation level

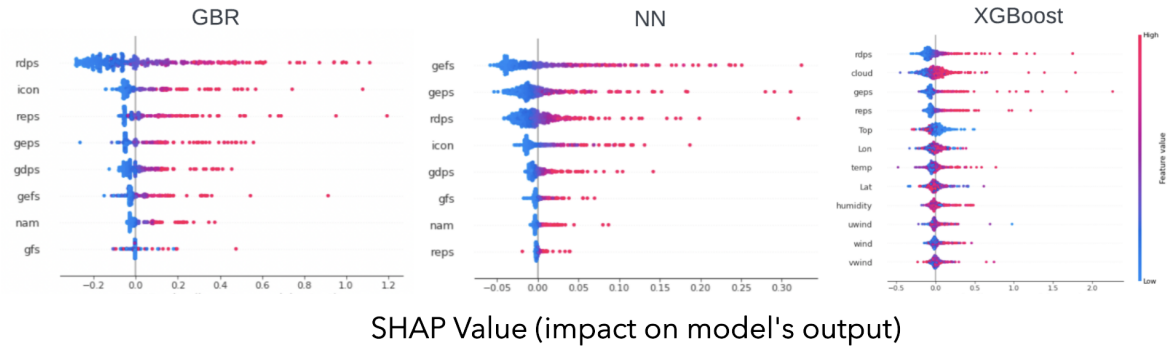

Figure 5.22: Shapley values for randomly sampled Light precipitation level

Additionally, XGBoost exhibits an increasing dependence on the three input weather models for significant deviations in its predictions.

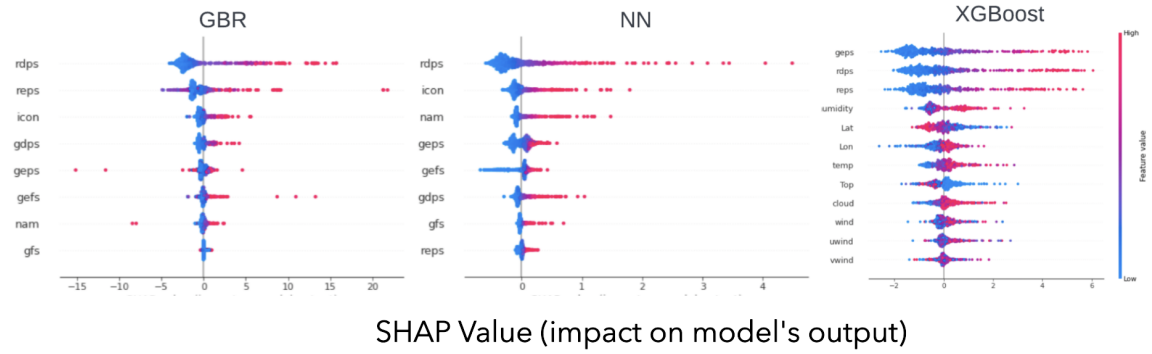

Figure 5.23: Shapley values for randomly sampled Moderate precipitation level

In general, the RDPS weather model emerges as the most influential factor in

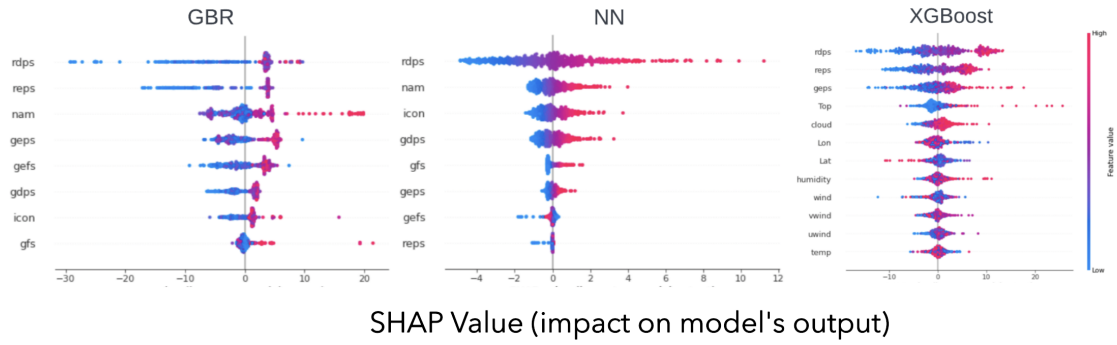

Figure 5.24: Shapley values for randomly sampled Heavy precipitation level

shaping the decisions of both the GBR and NN models, regardless of the precipitation levels. Interestingly, GBR does not seem to give much consideration to the GFS weather model, while NN tends to disregard REPS. This observation could potentially provide an explanation for NN's sub-par performance. Unlike GBR and NN, XGBoost considers most of its features to take a decision. XGBoost's balanced performance in all precipitation levels can be attributed to the presence of information from secondary features. All features collectively contribute significantly in the decision making process, especially when precipitation levels are high and shows that it is not dependent/heavily biased on a single feature.

Table 5.2: Model Training and Testing Times

Model	**IMM**	**MLR**	**RFR**	**GBR**	**XGB**	**NN**	**CNN**	**GNN**
Train	10 sec	65 sec	18 mins	5 mins	14 mins	30 mins	25 mins	15 mins
Test	10 sec	5 sec	5 sec	20 sec	2 mins	10 sec	10 sec	10 sec

The Shapley values have provided valuable insights into the feature importance and their contributions to model predictions. However, it is essential to consider not only the decision-making aspects but also the computational efficiency of the above mentioned models. The time performance of each model is summarized in the Table 5.2. Training times for all the models are within the 30-minute range, and the inference times for most of the models are under 20 seconds, with the exception of XGBoost. This is a particularly important factor in scenarios where real-time or rapid model deployment is essential.

In developing the precipitation prediction model for 24-hour accumulated forecasts, a deliberate decision was made to exclude recurrent neural network (RNNs) and LSTMs. The nature of the task did not inherently demand a temporal model,

as the focus was on capturing granular patterns rather than sequential dependencies over time. This choice was driven by a preference for model interpretability and considerations regarding available data size. In scenarios requiring more fine-grained predictions, such as 3-hourly intervals, the use of RNNs might have been more justified due to their effectiveness in capturing temporal dependencies at shorter intervals.

In summary, our study highlights the diverse strengths and trade-offs of different machine learning models for precipitation prediction. Understanding their performance metrics, bias tendencies, and feature contributions, alongside their computational efficiency, allows for informed model selection based on specific application needs and constraints. Further research and model refinement can build upon these findings to enhance the accuracy and efficiency of precipitation forecasting.

Chapter 6

Conclusions

Numerical Weather Prediction models have made remarkable strides in meteorology, allowing us to forecast atmospheric conditions with increasing accuracy. Despite these advancements, the forecasting of precipitation remains one of the most enduring challenges in the field. The complex, multi-scale nature of precipitation phenomena, the limitations of NWP models, and inherent uncertainties in atmospheric dynamics have necessitated a search for innovative approaches to improve precipitation predictions. In this thesis, we embarked on a comprehensive exploration of the fusion of 24-h period NWP precipitation forecast using an ML and DL covering most of the continental USA and Canada with dataset provided by WeatherLogics Inc. Our preliminary work reported in [1], provided a foundation for this research, demonstrating that machine learning and deep learning algorithms can indeed enhance the accuracy and reliability of NWP precipitation predictions.

In our previous work, we focused on a six-month dataset of 8 NWP models and employed 5 machine learning techniques, including Multiple Linear Regression, Random Forest Regression, Gradient Boosting Regression, Neural Networks and Convolutional Neural Networks, demonstrating that machine learning can significantly enhance the accuracy of NWP precipitation predictions. The larger dataset in our thesis, spanning 18 months, allowed us to dive deeper into the impacts of machine learning on precipitation forecasting. We expanded our dataset to include 9 additional secondary features like temperature, wind, location, etc, and trained additional machine learning models: XGBoost and Graph Neural Networks. Our analysis encompassed evaluation metrics such as Mean Absolute Error, Root Mean Squared Error, Median Absolute Error, Maximum Error, Correlation Coefficient, Relative Bias, Probability of Detection, False Alarm Rate, Critical Success Index,

and Confusion Matrix, providing a detailed understanding of model performance.

Our extensive evaluation of different machine learning models and input weather models for daily precipitation prediction has yielded valuable insights into their performance. We have assessed these models using various metrics, shedding light on their capabilities in predicting different aspects of precipitation. Convolutional Neural Networks and Gradient Boosting Regression emerged as top performers, displaying optimum performance on most metrics compared to the baseline methods currently deployed. With 15% improvement in Mean Absolute Error, 5% in Root Mean Squared Error, 45% in Median Absolute Error and 50% in Relative Bias, signifying proficiency of Convolutional Neural Networks in accurately predicting daily precipitation. Gradient Boosting Regression and XGBoost were able to approximately reduce extreme errors by 2%. Furthermore, all trained machine learning models exhibit high correlation compared to the input weather models and the IMM baseline. Out of the input weather models RDPS and REPS, show the lowest of errors and high capabilities to predict accurate precipitation in the regions of US and Canada. The confusion matrix show that, although Convolutional Neural Networks and Gradient Boosting Regression displayed strong overall performance, XGBoost and Graph Neural Networks demonstrated balanced performance across different precipitation levels with an above 80% accuracy in detecting heavy precipitation.

Furthermore, the Shapley values analysis revealed insights into feature importance and their contributions to model predictions. RDPS weather model appeared as the most influential factor in shaping model decisions for Gradient Boosting Regression and Neural Networks, while XGBoost distributed its reliance more evenly across various secondary meteorological features. Given the 500 randomly selected datapoints, XGBoost's balanced performance might be due to its consideration of multiple features. This suggests that the secondary features do hold valuable information and should be considered while training machine learning models in future research.

By incorporating a more extensive dataset, we observed a substantial enhancement in the performance of both Gradient Boosting Regression and Convolutional Neural Networks compared to our prior research. With an even longer time frame, it would be interesting to see the performance improvements with XGBoost and graph based neural networks, especially considering the integration of more meteorological features. There remains a vast landscape of potential future work in this field. One promising avenue for further research is the exploration of even more advanced

machine learning algorithms and techniques. This could involve the utilization of cutting-edge deep learning models, such as Transformers and Reinforcement Learning, to improve the modeling of complex meteorological processes. Additionally, the incorporation of high-resolution satellite and remote sensing data, along with more extensive ground-based observations, can further enhance the input features for machine learning models, potentially leading to more precise precipitation forecasts.

In the realm of climate change and extreme weather events, future work can delve into how machine learning can be harnessed to improve long-term climate modeling. This study highlights the strengths and weaknesses of different machine learning models and input weather models for daily precipitation prediction. The choice of the model should depend on the specific requirements of the application, such as the need for real-time predictions, accuracy in extreme conditions, or balanced performance across different precipitation levels. Understanding the changing nature of precipitation patterns in a warming world and adapting NWP models to these shifts is an area with high societal relevance. Our research contributes to the evolving field of meteorology by offering a comprehensive exploration of machine learning methods with an expanded dataset. The integration of secondary features and the introduction of new machine learning models present an opportunity to achieve even greater precision and reliability in precipitation predictions. The results from this research have the potential to revolutionize the way we approach next day precipitation forecasting, with far-reaching implications for various sectors influenced by weather forecasts.

Bibliography

[1] Sengoz C, Ramanna S, Kehler S, Goomer R, Pries P (2023) Machine learning approaches to improve north american precipitation forecasts. IEEE Access 11:97664–97681, DOI 10.1109/ACCESS.2023.3309054

[2] Bauer P, Thorpe A, Brunet G (2015) The quiet revolution of numerical weather prediction. Nature 525(7567):47–55

[3] Pu Z, Kalnay E (2019) Numerical Weather Prediction Basics: Models, Numerical Methods, and Data Assimilation, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 67–97

[4] Hou A, Kakar RK, Neeck SP, Azarbarzin AA, Kummerow CD, Kojima M, Oki R, Nakamura K, Iguchi T (2014) The global precipitation measurement mission. Bulletin of the American Meteorological Society 95:701–722

[5] Charney JG, Fjörtoft R, Neumann Jv (1950) Numerical integration of the barotropic vorticity equation. Tellus 2(4):237–254

[6] Lynch P (2008) The origins of computer weather prediction and climate modeling. Journal of computational physics 227(7):3431–3444

[7] Williams PD (2005) Modelling climate change: the role of unresolved processes. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 363(1837):2931–2946

[8] Grenier H, Bretherton CS (2001) A moist pbl parameterization for large-scale models and its application to subtropical cloud-topped marine boundary layers. Monthly weather review 129(3):357–377

[9] Richardson LF, Lynch P (2007) Weather Prediction by Numerical Process, 2nd edn. Cambridge Mathematical Library, Cambridge University Press, DOI 10.1017/CBO9780511618291

[10] Rodwell M, Palmer T (2007) Using numerical weather prediction to assess climate models. Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography 133(622):129–146

[11] Schultz MG, Betancourt C, Gong B, Kleinert F, Langguth M, Leufen LH, Mozaffari A, Stadtler S (2021) Can deep learning beat numerical weather prediction? Philosophical Transactions of the Royal Society A 379(2194):20200097

[12] Bochenek B, Ustrnul Z (2022) Machine learning in weather prediction and climate analyses—applications and perspectives. Atmosphere 13(2):180

[13] Shapley LS (1953) Stochastic games. Proceedings of the national academy of sciences 39(10):1095–1100

[14] Deo RC, Şahin M (2015) Application of the artificial neural network model for prediction of monthly standardized precipitation and evapotranspiration index using hydrometeorological parameters and climate indices in eastern australia. Atmospheric research 161:65–81

[15] Le J, El-Askary H, Allali M, Struppa DC (2017) Application of recurrent neural networks for drought projections in california. Atmospheric research 188:100–106

[16] Haidar A, Verma B (2018) Monthly rainfall forecasting using one-dimensional deep convolutional neural network. IEEE Access 6:69053–69063, DOI 10.1109/ACCESS.2018.2880044

[17] Wu Y, Tang Y, Yang X, Zhang W, Zhang G (2020) Graph convolutional regression networks for quantitative precipitation estimation. IEEE Geoscience and Remote Sensing Letters 18(7):1124–1128

[18] Xu J, Chen L, Lv M, Zhan C, Chen S, Chang J (2021) Highair: A hierarchical graph neural network-based air quality forecasting method. URL `https://doi.org/10.48550/arXiv.2101.04264`, `2101.04264`

[19] Ko CM, Jeong YY, Lee YM, Kim BS (2020) The development of a quantitative precipitation forecast correction technique based on machine learning for hydrological applications. Atmosphere 11(1), DOI 10.3390/atmos11010111, URL https://www.mdpi.com/2073-4433/11/1/111

[20] Weyn JA, Durran DR, Caruana R, Cresswell-Clay N (2021) Sub-seasonal forecasting with a large ensemble of deep-learning weather prediction models. Journal of Advances in Modeling Earth Systems 13(7):e2021MS002502

[21] Joshi P, Shekhar M, Kumar A, Quamara J (2022) Artificial neural network model for precipitation forecast over western himalaya using satellite images. MAUSAM 73(1):83–90

[22] Barrera-Animas AY, Oyedele LO, Bilal M, Akinosho TD, Delgado JMD, Akanbi LA (2022) Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting. Machine Learning with Applications 7:100204, DOI https://doi.org/10.1016/j.mlwa.2021.100204, URL https://www.sciencedirect.com/science/article/pii/S266682702100102X

[23] Hamill TM, Engle E, Myrick D, Peroutka M, Finan C, Scheuerer M (2017) The u.s. national blend of models for statistical postprocessing of probability of precipitation and deterministic precipitation amount. Monthly Weather Review 145(9):3441 – 3463, DOI https://doi.org/10.1175/MWR-D-16-0331.1, URL https://journals.ametsoc.org/view/journals/mwre/145/9/mwr-d-16-0331.1.xml

[24] Hamill TM, Stovern DR, Smith LL (2023) Improving national blend of models probabilistic precipitation forecasts using long time series of reforecasts and precipitation reanalyses. part i: Methods. Monthly Weather Review 151(6):1521 – 1534, DOI https://doi.org/10.1175/MWR-D-22-0308.1, URL https://journals.ametsoc.org/view/journals/mwre/151/6/MWR-D-22-0308.1.xml

[25] Stovern DR, Hamill TM, Smith LL (2023) Improving national blend of models probabilistic precipitation forecasts using long time series of reforecasts and precipitation reanalyses. part ii: Results. Monthly Weather Review 151(6):1535 – 1550, DOI https://doi.org/10.1175/MWR-D-22-0310.1, URL https://journals.ametsoc.org/view/journals/mwre/151/6/MWR-D-22-0310.1.xml

[26] Craven J, Rudack D, Shafer P (2020) National blend of models: A statistically post-processed multi-model ensemble. Journal of Operational Meteorology pp 1–14, DOI 10.15191/nwajom.2020.0801

[27] Buizza R, Palmer TN (1998) Impact of ensemble size on ensemble prediction. Monthly Weather Review 126(9):2503–2518

[28] Ebert EE (2001) Ability of a poor man's ensemble to predict the probability and distribution of precipitation. Monthly Weather Review 129(10):2461–2480

[29] Krasnopolsky V, Lin Y (2012) A neural network nonlinear multimodel ensemble to improve precipitation forecasts over continental US. Advances in Meteorology 2012, DOI 10.1155/2012/649450

[30] Gagne D, Mcgovern A, Xue M (2014) Machine learning enhancement of storm-scale ensemble probabilistic quantitative precipitation forecasts. Weather and Forecasting 29:1024–1043, DOI 10.1175/WAF-D-13-00108.1

[31] Scheuerer M, Switanek MB, Worsnop RP, Hamill TM (2020) Using artificial neural networks for generating probabilistic subseasonal precipitation forecasts over california. Monthly Weather Review 148(8):3489–3506

[32] Sønderby CK, Espeholt L, Heek J, Dehghani M, Oliver A, Salimans T, Agrawal S, Hickey J, Kalchbrenner N (2020) Metnet: A neural weather model for precipitation forecasting. URL `https://doi.org/10.48550/arXiv.2003.12140`, `2003.12140`

[33] Espeholt L, Agrawal S, Sønderby C, Kumar M, Heek J, Bromberg C, Gazen C, Hickey J, Bell A, Kalchbrenner N (2021) Skillful twelve hour precipitation forecasts using large context neural networks. URL `https://doi.org/10.48550/arXiv.2111.07470`, `2111.07470`

[34] Fan Y, Krasnopolsky V, van den Dool H, Wu CY, Gottschalck J (2021) Using artificial neural networks to improve CFS week 3-4 precipitation and 2-meter air temperature forecasts. Weather and Forecasting

[35] Fan Z, Li W, Jiang Q, Sun W, Wen J, Gao J (2021) A comparative study of four merging approaches for regional precipitation estimation. IEEE Access 9:33625–33637

[36] Cho D, Yoo C, Son B, Im J, Yoon D, Cha DH (2022) A novel ensemble learning for post-processing of NWP model's next-day maximum air temperature forecast in summer using deep learning and statistical approaches. Weather and Climate Extremes 35:100410

[37] Frnda J, Durica M, Rozhon J, Vojtekova M, Nedoma J, Martinek R (2022) ECMWF short-term prediction accuracy improvement by deep learning. Scientific Reports 12(1):1–11

[38] Li W, Duan Q, Wang QJ, Huang S, Liu S (2022) Evaluation and statistical post-processing of two precipitation reforecast products during summer in the mainland of China. Journal of Geophysical Research: Atmospheres 127(12):e2022JD036606

[39] Jha SK, Shrestha DL, Stadnyk TA, Coulibaly P (2018) Evaluation of ensemble precipitation forecasts generated through post-processing in a Canadian catchment. Hydrology and Earth System Sciences 22(3):1957–1969

[40] Robertson D, Shrestha D, Wang Q (2013) Post-processing rainfall forecasts from numerical weather prediction models for short-term streamflow forecasting. Hydrology and Earth System Sciences 17(9):3587–3603

[41] Zhou K, Sun J, Zheng Y, Zhang Y (2022) Quantitative precipitation forecast experiment based on basic NWP variables using deep learning. Advances in Atmospheric Sciences pp 1–15

[42] Bi K, Xie L, Zhang H, Chen X, Gu X, Tian Q (2022) Pangu-weather: A 3d high-resolution model for fast and accurate global weather forecast. URL `https://doi.org/10.48550/arXiv.2211.02556`, `2211.02556`

[43] Keisler R (2022) Forecasting global weather with graph neural networks. URL `https://doi.org/10.48550/arXiv.2202.07575`, `2202.07575`

[44] Lam R, Sanchez-Gonzalez A, Willson M, Wirnsberger P, Fortunato M, Alet F, Ravuri S, Ewalds T, Eaton-Rosen Z, Hu W, Merose A, Hoyer S, Holland G, Vinyals O, Stott J, Pritzel A, Mohamed S, Battaglia P (2023) Graphcast: Learning skillful medium-range global weather forecasting. URL `https://doi.org/10.48550/arXiv.2212.12794`, `2212.12794`

[45] Dong J, Zeng W, Wu L, Huang J, Gaiser T, Srivastava AK (2023) Enhancing short-term forecasting of daily precipitation using numerical weather prediction bias correcting with xgboost in different regions of china. Engineering Applications of Artificial Intelligence 117:105579

[46] Cunningham P, Cord M, Delany SJ (2008) Supervised learning. In: Machine learning techniques for multimedia: case studies on organization and retrieval, Springer, pp 21–49

[47] Quinlan JR (1986) Induction of decision trees. Machine learning 1:81–106, URL https://doi.org/10.1007/BF00116251

[48] Breiman L (2001) Random forests. Machine learning 45:5–32, URL https://doi.org/10.1023/A:1010933404324

[49] Friedman J (2000) Greedy function approximation: A gradient boosting machine. The Annals of Statistics 29, DOI 10.1214/aos/1013203451

[50] Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp 785–794

[51] Otter DW, Medina JR, Kalita JK (2020) A survey of the usages of deep learning for natural language processing. IEEE transactions on neural networks and learning systems 32(2):604–624

[52] Wu L, Chen Y, Shen K, Guo X, Gao H, Li S, Pei J, Long B, et al (2023) Graph neural networks for natural language processing: A survey. Foundations and Trends® in Machine Learning 16(2):119–328

[53] Abiodun OI, Jantan A, Omolara AE, Dada KV, Mohamed NA, Arshad H (2018) State-of-the-art in artificial neural network applications: A survey. Heliyon 4(11)

[54] Hewage P, Trovati M, Pereira E, Behera A (2021) Deep learning-based effective fine-grained weather forecasting model. Pattern Analysis and Applications 24(1):343–366

[55] Wong BK, Selvi Y (1998) Neural network applications in finance: A review and analysis of literature (1990–1996). Information & management 34(3):129–139

[56] Rosenblatt F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review 65(6):386

[57] Minsky M, Papert S (1969) An introduction to computational geometry. Cambridge tiass, HIT 479(480):104

[58] Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. nature 323(6088):533–536

[59] Osler TJ (1971) Fractional Derivatives and Leibniz Rule, vol 78. Mathematical Association of America, URL http://www.jstor.org/stable/2316573

[60] LeCun Y, Bengio Y, et al (1995) Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks 3361(10):1995

[61] Girshick R, Donahue J, Darrell T, Malik J (2015) Region-based convolutional networks for accurate object detection and segmentation. IEEE transactions on pattern analysis and machine intelligence 38(1):142–158

[62] Xiao Y, Tian Z, Yu J, Zhang Y, Liu S, Du S, Lan X (2020) A review of object detection based on deep learning. Multimedia Tools and Applications 79:23729–23791

[63] Diwan T, Anirudh G, Tembhurne JV (2023) Object detection using yolo: Challenges, architectural successors, datasets and applications. multimedia Tools and Applications 82(6):9243–9275

[64] Liu L, Ouyang W, Wang X, Fieguth P, Chen J, Liu X, Pietikäinen M (2020) Deep learning for generic object detection: A survey. International journal of computer vision 128:261–318

[65] Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. URL https://doi.org/10.48550/arXiv.1609.02907, 1609.02907

[66] Chai T, Draxler RR (2014) Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. Geoscientific model development 7(3):1247–1250

[67] Korn GA, Korn TM (2000) Appendix b: B9. plane and spherical trigonometry: Formulas expressed in terms of the haversine function. Mathematical handbook for scientists and engineers: Definitions, theorems, and formulas for reference and review, pp 892–893

[68] Kim T, Ho N, Kim D, Yun SY (2022) Benchmark dataset for precipitation forecasting by post-processing the numerical weather prediction. URL `https://doi.org/10.48550/arXiv.2206.15241`, `2206.15241`

Appendix A

Appendix

A.1 Precipitation Forecasts

This section includes sample visualizations of the 24 hour accumulated precipitation derived from RDPA (ground truth) and forecasts from trained ML algorithms: Machine learning Regression (MLR), Random Forest Regression (RFR), Gradient Boosting Regression (GBR), XGBoost (XGB), Simple Neural Networks (NN), Convolutional Neural Networks (CNN) and Graph Neural Networks (GNN). We also share visualization of the errors made by the trained ML models to better understand the strengths and weakness of each model. The trained ML models can be seen to perform better than the baseline Input Means Model (IMM), with most errors located in the wet regions and low errors are scene in drier areas.

A.1.1 Visualization for 1^{st} $March$, 2023

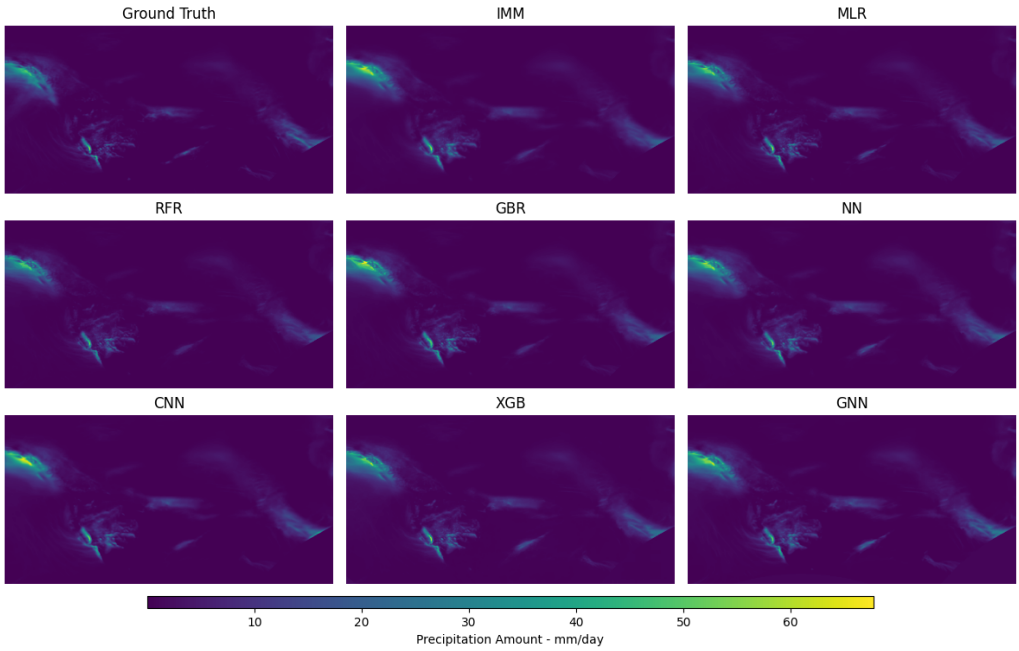

Figure A.1: Visualization of accumulated precipitation for March 1^{st} 2023.

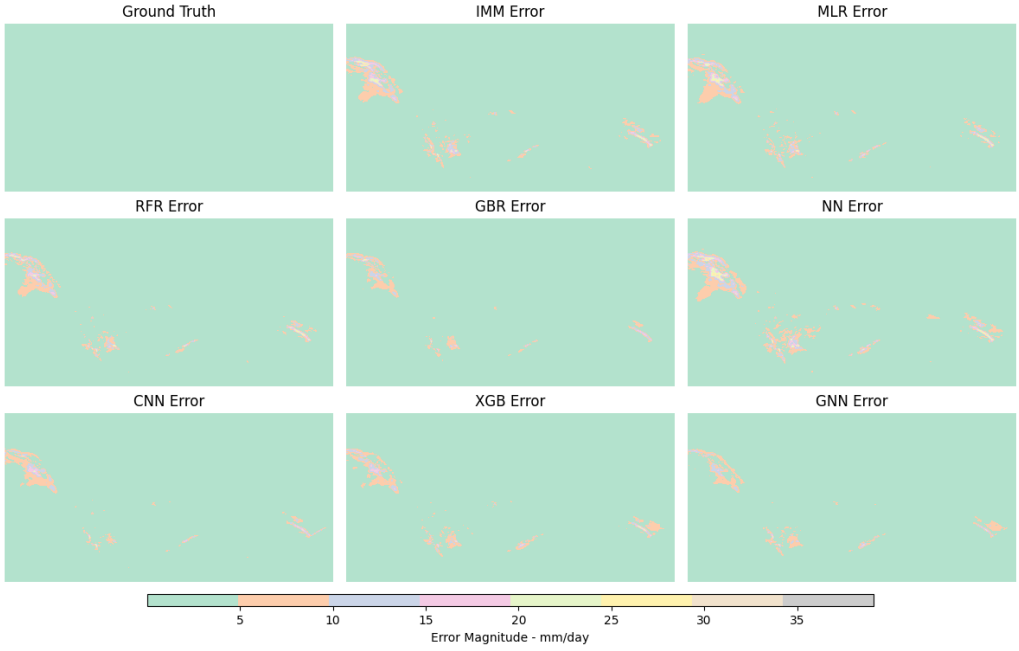

Figure A.2: Visualizing error in accumulated precipitation for March 1^{st} 2023.

A.1.2 Visualization for 1^{st} *April*, 2023

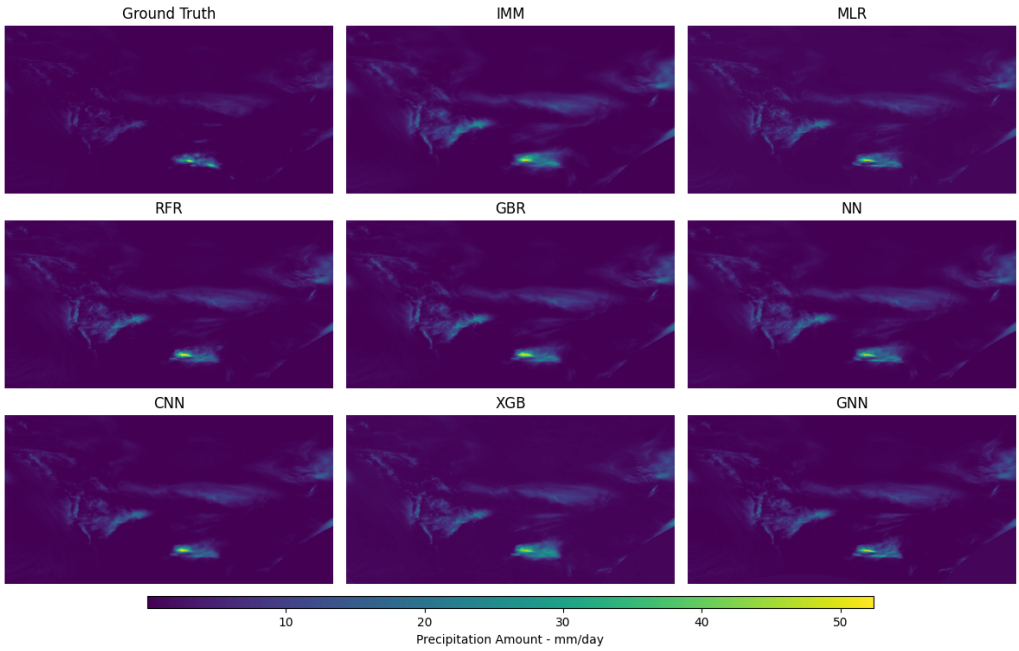

Figure A.3: Visualizing accumulated precipitation for April 1^{st} 2023

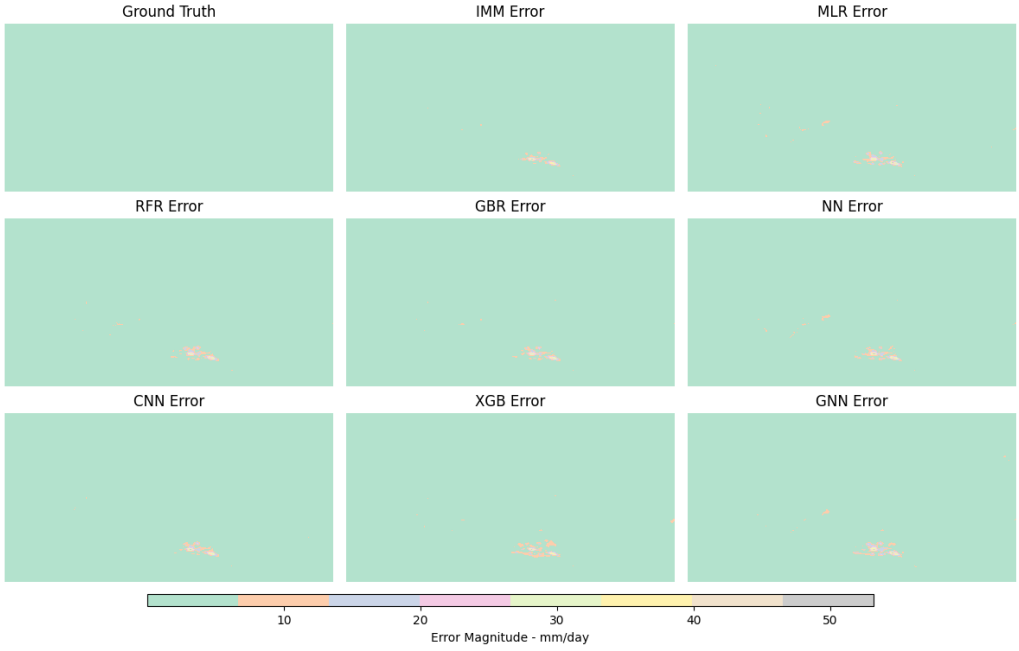

Figure A.4: Visualizing error in accumulated precipitation for April 1^{st} 2023

A.1.3 Visualization for 1^{st} May, 2023

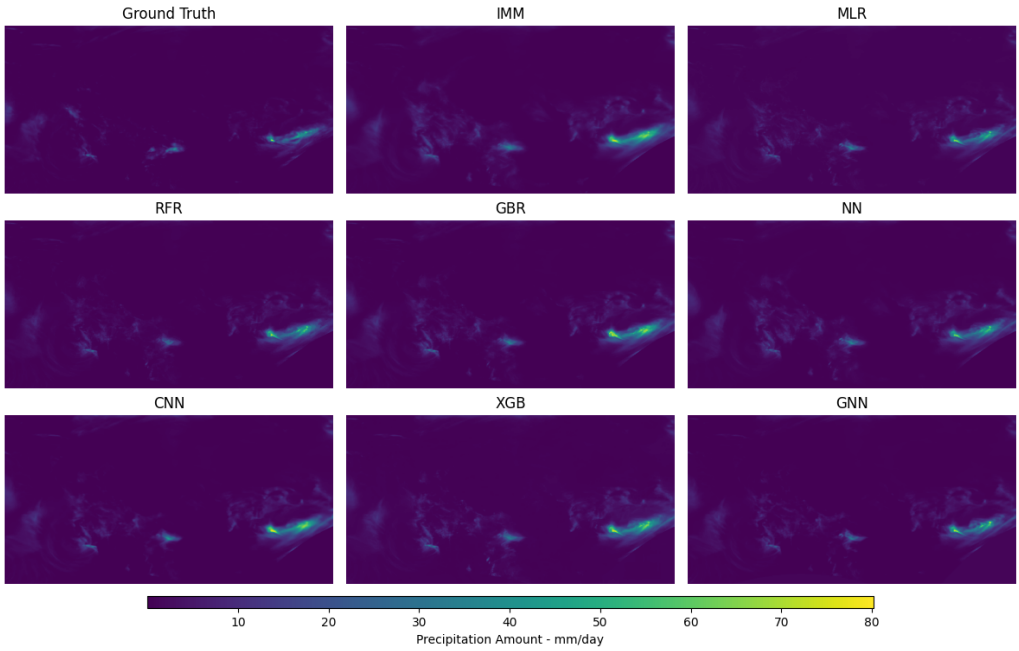

Figure A.5: Visualizing accumulated precipitation for May 1^{st} 2023

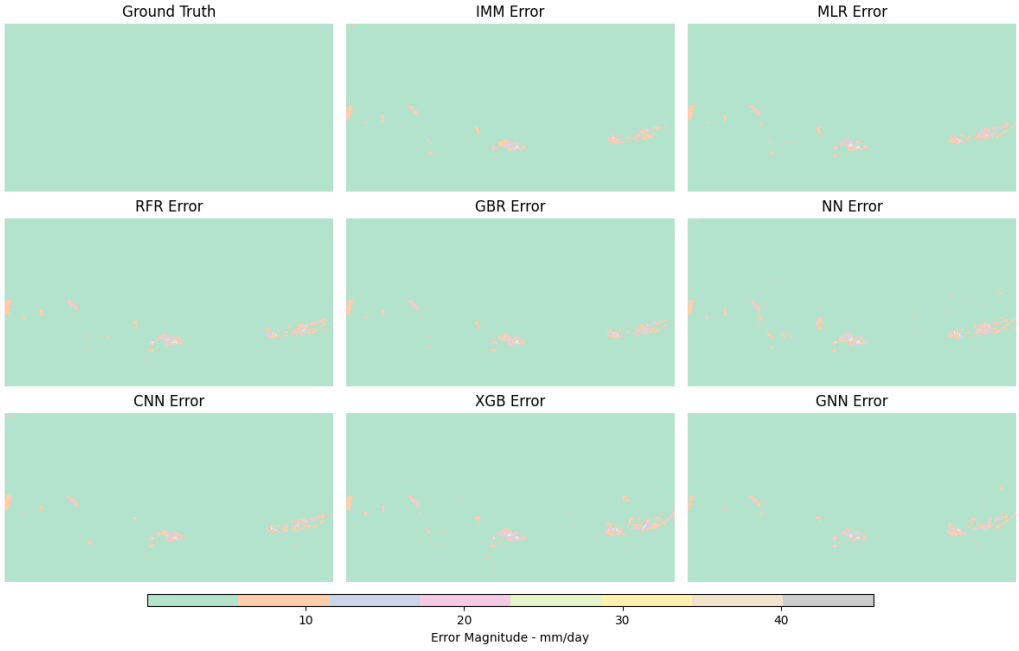

Figure A.6: Visualizing error in accumulated precipitation for May 1^{st} 2023

A.2 Model Parameters

Table A.1: Trained ML model hyperparameters yielding the most optimal results.

Model	Specifications
MLR	Loss: squared error
GBR	Number of trees:100 Tree depth: 7 Loss: squared error
RFR	Tree depth: 7 Number of trees: 10 Loss: squared error
NN	Number of trainable params: 109,201 Number of hidden layers : 2 Number of neurons in hidden layer : (1000, 100) Dropout: 0.1 Optimizer: RMSprop Batch size: 200,000 Loss: squared logarithmic error Normalized: Yes (Z-Score)
CNN	Number of trainable params: 79,105 Number of hidden layers: 3 Dropout: 0.1 Optimizer: RMSprop Kernel initialization: Random normal Kernel count and shape: $64 \times (3 \times 3)$ Batch size: 1 Loss: squared logarithmic error Normalized: Yes (Z-Score)
XGBoost	Number of trees: 500 Tree depth: 15 Sample Weights: Yes Gamma: 0.1 Subsample: 0.5 Alpha and Lambda: 0.3
GNN	Number of trainable params: 1,105 Number of hidden layers: 2 Batch size: 1 Kernel shape: 10×10 Dropout: 0.001 Optimizer: Adam Kernel initialization: Xavier

A.3 Confusion Matrix

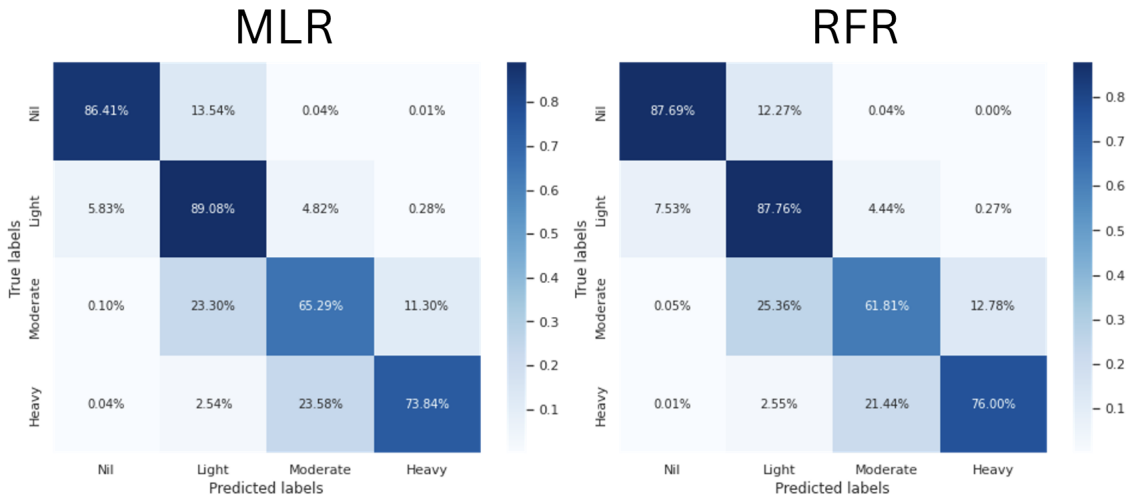

Figure A.7: Confusion matrix of Machine Learning Regression (MLR) and Random Forest Regression (RFR) for different precipitation levels. Both of the trained ML models show a good performance when predicting low precipitation levels, but perform poorly when predicting higher precipitation levels.