UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

Improving Radar Sensing Capabilities and Data Quality Through Machine Learning

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

Anas Amaireh
Norman, Oklahoma
2024

Improving Radar Sensing Capabilities

and

Data Quality Through Machine Learning

A DISSERTATION APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Yan Zhang, Chair

Dr. Cameron Homeyer

Dr. David Schvartzman

Dr. Samuel Cheng

Dr. David Ebert

To my mother, whose boundless love and sacrifices have been the cornerstone of my journey - your unwavering belief in me has been my greatest strength.

To my father, whose wisdom and guidance have shaped the person I am today - your support has been my guiding star, leading me through every challenge.

To my sisters, Wlla and Farah, and my brother Mohammad, for being my strong support and for standing by me through thick and thin - your love means everything to me.

And to the quiet inspiration whose influence has fueled passion and illuminated my path with warmth and light - this impact has been truly invaluable and will never be forgotten.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Rockee Zhang. His unwavering support, guidance, and belief in my potential have been instrumental in reaching this milestone. Dr. Zhang not only provided me with the opportunity to pursue this research but also continually inspired me with his dedication and profound knowledge. He treated us like family, always showing patience, understanding, and flexibility. I am fortunate to have worked under his guidance, and I truly appreciate his trust in me to succeed. Thank you, Dr. Zhang, for everything.

I would also like to sincerely thank the members of my dissertation committee sincerely, Dr. David Schvartzman, Dr. Cameron Homeyer, Dr. Samuel Cheng, and Dr. David Ebert. Their valuable feedback, insightful comments, and encouragement throughout this journey have greatly enriched my work. Their expertise and dedication to excellence have been truly motivating.

I am deeply grateful to Dr. Pak-wai Chan, Director of the Hong Kong Observatory, for providing this research's essential data and resources. His invaluable advice and support have significantly contributed to the success of this work.

I want to thank my colleagues for their collaborative spirit, which made this journey both fulfilling and enriching. I also thank the Department of Electrical and Computer Engineering (ECE) and the Advanced Radar Research Center (ARRC) for providing an

innovative and supportive environment. The resources and opportunities offered by the department and ARRC were instrumental in completing this work.

Contents

List Of Tables

List Of Figures

Abstract

This dissertation integrates advanced machine learning (ML) techniques with radar technology to address significant challenges in atmospheric sciences, cloud profiling, and aviation safety. It aims to enhance the accuracy and reliability of radar-based measurements, improve the prediction of atmospheric relative humidity and Cloud Liquid Water Content (CLWC), and mitigate the impact of 5G interference on radar altimeters. These improvements are essential for advancing public safety, weather forecasting, and aviation technology.

Chapter 2 provides a comprehensive overview of ML, detailing its history, evolution, and significance in scientific research. It introduces supervised, unsupervised, and reinforcement learning, and discusses various ML models, such as regression and classification, establishing a foundation for integrating ML with radar technology.

Chapter 3 introduces a method for estimating atmospheric relative humidity using wind profiler radar and a cascaded ML algorithm. Unlike existing methods, this approach uses only moment data to generate an intermediate pressure profile, serving as training data for humidity estimations without requiring temperature as an input feature. The study evaluates various ML algorithms using radiosonde data from the Hong Kong Observatory, demonstrating the effectiveness of this simplified, feature-efficient model.

Chapter 4 uses ML techniques to enhance Cloud Liquid Water Content (CLWC) profiling. The study cross-validates ERA5 data with high-precision radiosonde observations from Hong Kong. It employs ML to interpolate radiosonde data to improve coverage and resolution, and uses a metaheuristic algorithm to cleanse data. This enhances the correlation between

input features and CLWC. The results show significant improvements in the accuracy and reliability of CLWC profile prediction.

Chapter 5 addresses the critical issue of 5G interference with radar altimeter signals, which is crucial for aviation safety. A new ML framework is developed to classify signals into pure or interfered categories and predict altitudes when interference is detected. The study employs real 5G signals from a base station in Norman, Oklahoma, and emulated radar signals to train and test the framework. This approach ensures the accuracy of radar altimeters despite the level of 5G interference.

This dissertation demonstrates the transformative potential of integrating ML techniques with radar technology. The proposed solutions enhance radar sensing capabilities and data quality, significantly improving public safety, weather forecasting, and aviation.

Chapter 1

Introduction

1.1 Radar Sensing

Radar technology, an acronym for "radio detection and ranging," has been a fundamental part of modern science and engineering since its inception in the early twentieth century. It was initially used as an instrument for navigation, surveillance, and enemy identification, notably during World War II (Skolnik 1980). The critical function of radar is to detect and estimate the presence, direction, distance, and speed of objects ranging from aircraft to atmospheric formations (Skolnik 1980). The detection and estimation are carried out by emitting and reflecting radio waves, with the return time providing crucial information about the target's distance and position. However, in practical applications, the radar must address complex scientific concepts such as signal degradation, noise interference, and the varying reflective properties of different objects (Levanon and Mozeson 2004).

Due to digital signal processing and machine learning advancements, radar technology has seamlessly integrated into various fields, including meteorology, aviation, defense, and autonomous vehicles (Bishop and Nasrabadi 2006). While initially developed for military use, radar's potential for weather prediction was quickly recognized, leading to the development of weather radar systems in the mid-20th century. These systems have significantly advanced weather prediction by offering real-time tracking and analysis of atmospheric phenomena,

detecting precipitation, and identifying storm patterns (Meischner 2005). For instance, radar systems generate abundant data for weather forecasts and early warnings of potential threats such as hurricanes or severe storms (Meischner 2005). Additionally, radar technology is integral to aviation safety through radar altimeters, which measure altitudes above ground levels, crucial for aviation and navigation (Skolnik 1980).

1.2 Machine Learning and Deep Learning: A Brief Overview

Machine learning (ML) is a method of data analysis that automates the building of analytical models (Alpaydin 2020). Unlike traditional programming, where computers are directly given rules to follow, ML uses statistical techniques to learn from provided examples and apply those patterns to new data. ML has two primary forms: supervised learning and unsupervised learning. Supervised learning uses a labeled dataset with known desired outcomes, while unsupervised learning identifies inherent patterns and structures within unlabeled data (Goodfellow et al. 2016).

Deep learning (DL), a subtype of ML, employs connected layers of nodes or "neurons" inspired by the structure and function of the human brain. Unlike typical ML models that require manual feature extraction, DL can learn from raw, unstructured data, such as pixel values in photos (Goodfellow et al. 2016). Since the 1980s, ML has evolved significantly, driven by greater data availability and more powerful computational resources. This evolution opened the way for more advanced statistical procedures (Alpaydin 2020). The 1990s saw the creation of significant ML algorithms, such as support vector machines (SVM) and random forests, which remain widely used today. DL began to gain prominence in the 2000s, accelerated by the increased availability of large labeled datasets, efficient training techniques, and the rise of graphics processing units (GPUs) (Goodfellow et al. 2016).

Applications of ML and DL are widespread in modern technology, enhancing various fields. ML drives technologies ranging from recommendation algorithms on online shopping platforms to fraud detection in financial services. DL, in particular, has revolutionized fields requiring complex pattern recognition, such as image and speech recognition in virtual assistants and facial identification systems on smartphones (LeCun et al. 2015).

1.3 ML/DL and Radar Technology Intersection

Integrating machine learning (ML) and deep learning (DL) with radar technology has revolutionized various fields. Radar data, with its high volume and complexity, provides an excellent resource for ML and DL applications, enhancing efficiency and accuracy in weather prediction, object recognition, and interference reduction (Gini and Rangaswamy 2008).

Traditional weather prediction models rely on physics-based methodologies, which have limitations due to assumptions and the inability to fully capture atmospheric dynamics (Senthil Kumar 2019). ML addresses these gaps by identifying predictive patterns from large datasets. Early applications in radar meteorology included classifying precipitation types using models like decision trees, k-nearest neighbors, and support vector machines, which outperform traditional methods in distinguishing between rain, snow, and hail (Besic et al. 2016; McGovern et al. 2023).

ML has significantly improved the integration of radar data into weather prediction models by learning detailed correlations between radar measurements and atmospheric conditions (McGovern et al. 2017). ML techniques, such as decision trees and ensemble methods, have enhanced nowcasting and short-term weather predictions by capturing the complexity of atmospheric dynamics, resulting in more accurate forecasts (Zhou et al. 2020). Additionally, ML has improved radar data correction and precipitation estimation, aiding in flood forecasting and water resource management (Hassan et al. 2022; Li et al. 2023).

While DL is not the primary focus of this dissertation, it has contributed to advancements in radar technology where complex pattern recognition is needed. DL-based strategies have been used in target detection and precipitation nowcasting, demonstrating effectiveness in managing spatial correlations and improving data analysis reliability (Gagne II et al. 2019; Jiang et al. 2021; Shi et al. 2015).

Beyond weather radar technology, ML enhances aviation safety by improving radar altimeter performance through noise filtering and accurate ground detection, resulting in safer aircraft operations (Amaireh and Zhang 2023). In marine applications, radar measures wave heights and detects obstacles in navigation, with ML refining these processes by differentiating types of sea clutter, leading to safer maritime navigation (Vicen-Bueno et al. 2009). In surveillance and defense, ML improves target recognition using radar return signals and enhances synthetic aperture radar (SAR) imaging quality, improving picture reliability (Fracastoro et al. 2021).

The potential benefits of ML and DL in radar technology are vast. As extreme weather events increase due to climate change, advancements in weather prediction and radar technology can significantly impact public safety, agriculture, and socioeconomic factors (Easterling et al. 2017). Ongoing advancements in ML and DL promise continued progress across various sectors, opening the way for technological innovation and scientific discovery.

1.4 Outlines and Research Objectives

This dissertation explores integrating machine learning (ML) techniques with radar technology to address various challenges in atmospheric sciences and communication systems. Each chapter tackles specific research questions to enhance atmospheric measurements, improve cloud profiling, and mitigate 5G interference in radar altimeters.

Chapter 2 provides a comprehensive overview of ML and its evolution, highlighting its significance in scientific research and its integration with radar technology. This chapter discusses various types of ML, including supervised, unsupervised, and reinforcement learning, and examines common ML models used in this dissertation, such as regression and classification models. By laying the foundation for understanding the integration of ML with radar technology, this chapter sets the stage for their application in subsequent chapters.

Chapter 3 aims to develop and validate a novel ML approach for estimating atmospheric relative humidity (RH) profiles using wind profiler radar data. The study introduces a cascaded ML framework that uses only moment data, avoiding reliance on temperature. An intermediate pressure estimation stage is integrated to enhance RH prediction accuracy, with performance assessed across different seasons for robustness and reliability. This chapter begins with an introduction to the importance of accurate humidity estimation and the limitations of existing models, followed by a literature review identifying gaps in current methods. It details instrumentation, data collection, and preprocessing steps. The methodology explains the cascaded ML approach and algorithm selection. Different performance evaluation metrics are used to compare the ML algorithms. The results and discussion analyze the models' performance using six-month and month-by-month datasets of 2021 in the Hong Kong area. The chapter concludes by highlighting the strengths of the novel ML framework while addressing potential limitations and suggesting future research directions.

Chapter 4 focuses on enhancing Cloud Liquid Water Content (CLWC) profiling using ML techniques, with the main goal of predicting CLWC profiles based on input features such as pressure, temperature, relative humidity, and specific humidity. The primary objectives include cross-validating ECMWF's ERA5 data with high-precision radiosonde observations from Hong Kong, improving data coverage and resolution through ML interpolation, and applying a metaheuristic algorithm for data cleansing to strengthen the correlation between input features and CLWC. Various regression ML algorithms are evaluated for performance and robustness. This chapter begins by highlighting the importance of accurate CLWC

prediction for weather forecasting, especially in microclimates like Hong Kong. It reviews advancements and limitations in traditional and ML-based CLWC estimation methods. The data description section details ERA5 and radiosonde datasets and their preprocessing. The methodology section explains how to use a metaheuristic algorithm for data cleaning and applying ML models. Results show that decision tree and ensemble models outperform the accuracy of neural networks and linear regression. The chapter concludes by discussing the approach's strengths, limitations, and future research directions.

Chapter 5 studies the significant issue of 5G interference with radar altimeter signals, which is essential for aviation safety. The chapter's main objective is to develop an ML framework to identify and mitigate the effects of 5G interference on radar altimeters. This framework categorizes signals as either pure or interfered and employs regression ML models to predict altitudes when interference is present, effectively addressing and reducing the impact of such interference. The study uses real 5G signals from a base station in Norman, Oklahoma, and emulates FMCW radar signals using MATLAB to test the ML framework in real-world conditions. The chapter starts with an introduction to the vital role of radar altimeters in aviation safety and the interference challenges posed by 5G networks. The data collection section explains how 5G signals were gathered and how radar altimeter signals were simulated. The methodology section outlines the ML framework, detailing feature extraction, preprocessing steps, and model training. The models' effectiveness is evaluated using many regression and classification metrics. The results and discussion section presents findings from classification and altitude prediction, showcasing the performance of different ML models. The chapter concludes by discussing the proposed approach's strengths and limitations and suggesting future research directions.

1.5 Research Methodology

The research methodology employed in this dissertation covers various machine learning (ML) techniques and data sources to address distinct challenges in atmospheric sciences and radar technology. Each chapter leverages specific methodologies tailored to the research objectives and data characteristics.

For the study on atmospheric humidity estimation, data collection involved wind profilers and radiosondes in Hong Kong, providing comprehensive atmospheric measurements. Data preprocessing included outlier removal and data smoothing to ensure quality. A cascading ML solution was implemented, utilizing regression decision trees, ensemble trees, and neural networks. The bagged learning tree was selected for its effectiveness and computational efficiency balance. Performance was evaluated using standard regression metrics and a 5-fold cross-validation approach to ensure robustness and accuracy.

In the cloud liquid water content (CLWC) profiling study, data from the ERA5 climate dataset were validated against high-precision radiosonde observations. Data cleaning and preprocessing involved advanced metaheuristic algorithms to improve feature correlation. Various ML models, including linear regression, ensemble trees, and neural networks, were used, with the Bagged Ensemble Tree (BET) noted for its robust performance. Standard regression metrics and 5-fold cross-validation ensured comprehensive model evaluation.

Addressing the issue of 5G interference with radar altimeters, the study involved collecting 5G signals using Software-Defined Radio (SDR) and simulating radar signals to create a comprehensive dataset. Preprocessing included segmenting and rescaling 5G signals to emulate different interference levels, followed by feature extraction from both time-domain and frequency-domain characteristics. ML models, including classification and regression algorithms, were employed to identify and mitigate 5G interference. Performance evaluation used standard classification and regression metrics, with a 5-fold cross-validation approach to enhance model reliability and generalizability. The framework ensured accurate and consistent altitude measurements despite 5G interference, enhancing aviation safety and efficiency.

Chapter 2

Theoretical Foundations of Machine Learning Methods

2.1 Overview of Machine Learning

Machine Learning (ML) is a subfield of artificial intelligence (AI) that focuses on developing algorithms to enable computers to learn from data and make predictions, classifications, or decisions based on new, unseen data. Unlike traditional programming, which requires detailed instructions for each task, ML algorithms can recognize relationships and patterns within data. This capability allows systems to improve performance over time through experience (Mitchell 1997).

2.1.1 History and Evolution

The origins of machine learning can be traced back to the early days of AI research in the 1950s. Arthur Samuel created the phrase "machine learning" in 1959, defining it as the field of research that enables computers to learn without being explicitly programmed (Samuel 1959). Early research concentrated on symbolic AI and rule-based systems, which were limited by their inability to deal with data uncertainty and variability (Nilsson 1998). The introduction of statistical learning theory in the 1970s and 1980s marked a dramatic change, resulting in the development of probabilistic models and data-driven techniques (Bishop and Nasrabadi 2006). During this time, fundamental algorithms like decision trees

(Quinlan 1986), support vector machines (SVM) (Cortes and Vapnik 1995), and neural networks were developed (Mitchell 1997). The backpropagation method, introduced in 1986, motivated interest in neural networks and provided the foundation for current deep learning (DL) (Rumelhart et al. 1986).

Significant improvements in computational power and data availability occurred during the 1990s and 2000s, resulting in the development of more sophisticated algorithms and the application of ML across a wide range of domains (Domingos 2015). The rise of big data and the growing number of large-scale datasets have accelerated the evolution of machine learning, allowing models to learn from massive volumes of data while achieving extraordinary levels of accuracy and performance (Hastie et al. 2009). In recent years, deep learning, a subset of ML characterized by multi-layered neural networks, has revolutionized the field by achieving state-of-the-art results in areas such as image and speech recognition, natural language processing, and autonomous systems (Goodfellow et al. 2016). This period has also seen the rise of ML frameworks and libraries like MATLAB, TensorFlow, PyTorch, and Scikit-learn, allowing access to powerful ML tools and facilitating rapid experimentation and deployment (Goodfellow et al. 2016).

2.1.2 Significance in Scientific Research

Machine learning (ML) has become integral to modern scientific research, providing revolutionary data analysis, pattern recognition, and predictive modeling capabilities. Its ability to handle complex, high-dimensional datasets allows it to uncover insights beyond the reach of traditional statistical methods (Bishop and Nasrabadi 2006). ML analyzes vast datasets in fields such as genomics, neuroscience, and materials science to identify novel patterns and causal relationships, driving new discoveries (Jordan and Mitchell 2015). ML greatly improves predictive accuracy in various scientific areas, including climate modeling, disease outbreak prediction, and financial forecasting (Russell and Norvig 2016).

ML also automates complex, time-consuming processes such as image and signal processing, allowing researchers to concentrate on higher-level analysis. This is particularly valuable in medical imaging, where ML aids in diagnosing conditions from radiographic data (Krizhevsky et al. 2012). The adaptability of machine learning methods allows them to be applied across a wide range of scientific areas, enabling interconnected research and integrating varied data sources. This leads to more comprehensive approaches to solving complex scientific problems. Additionally, ML models can process and analyze data in real time, making them invaluable for rapid decision-making in autonomous vehicles, robotics, and real-time monitoring systems in environmental and industrial contexts (Chen and Guestrin 2016).

2.2 Types of Machine Learning

Machine learning (ML) can be broadly categorized into three main types: supervised learning, unsupervised learning, and reinforcement learning. Each type has distinct methodologies and applications, addressing different kinds of problems and data structures.

2.2.1 Supervised Learning

Supervised learning is the most common type of machine learning. This approach trains the algorithm on a labeled dataset, meaning that each training example is paired with an output label. The objective is for the model to learn a mapping from inputs to outputs to accurately predict the output for new, unseen inputs. Typical applications of supervised learning include classification and regression tasks. In classification, the output variable is categorical, such as spam detection in emails, where the input is the email text and the output is either 'spam' or 'not spam' (Bishop and Nasrabadi 2006). Regression tasks involve predicting a continuous output variable, like predicting house prices based on features such as size, location, and number of bedrooms (Mitchell 1997). Supervised learning algorithms

are widely used in image and speech recognition, where models learn from labeled images or audio clips to identify objects, people, or spoken words in new data (LeCun et al. 2015). Supervised learning is particularly effective when a large amount of labeled data is available, and the relationship between the input and output variables is relatively straightforward.

2.2.2 Unsupervised Learning

Unsupervised learning, on the other hand, works with unlabeled data and attempts to determine the fundamental structure existing in a set of data points. Since the data is not labeled, the algorithm tries to learn patterns and the underlying structure from the data itself. Typical applications of unsupervised learning include clustering, dimensionality reduction, and anomaly detection. Clustering algorithms group data points into clusters based on their similarity, which is useful in applications like customer segmentation based on purchase behavior without predefined group definitions (Hastie et al. 2009). Dimensionality reduction techniques, such as Principal Component Analysis (PCA), reduce the number of variables in a dataset while keeping as much information as possible, assisting in processing high-dimensional data or preprocessing before applying supervised learning (Bishop and Nasrabadi 2006). Unsupervised learning is also employed in anomaly detection to identify rare items, events, or observations that don't match the rest of the data, making it valuable in fraud detection, network security, and industrial fault detection (Breiman 2001). Unsupervised learning is valuable when the goal is to explore the data and find hidden patterns without having predefined labels.

2.2.3 Reinforcement Learning

Reinforcement learning (RL) represents a different paradigm in machine learning, where an agent learns to make decisions by performing actions in an environment to maximize some concept of cumulative reward (Sutton and Barto 2018). Unlike supervised learning, the agent is not provided with correct input-output pairs but must discover them through trial

and error. Typical applications of reinforcement learning include game playing (Mnih et al. 2015), robotics (Lillicrap et al. 2015), and self-driving cars (Arulkumaran et al. 2017). RL has been applied in game playing, where agents learn strategies to win games like Chess, Go, and video games, exemplified by DeepMind's AlphaGo program, which defeated world champions in Go (Silver et al. 2016, 2017). In robotics, RL helps in training robots to perform tasks such as navigation, manipulation, and interaction with the environment, where robots learn to improve their performance over time through rewards and penalties (Russell and Norvig, 2016). Similarly, RL is crucial in the development of autonomous vehicles, where the car learns to drive by receiving rewards for safe driving behaviors and penalties for unsafe actions (Chen, Guestrin, 2016). Reinforcement learning is particularly suited to problems where an agent must make a sequence of decisions and where the outcome of one decision influences the future state and rewards.

Understanding the different types of machine learning is crucial for selecting the appropriate algorithms for a given problem. Supervised learning is highly effective when labeled data is available and the goal is to predict outcomes (James et al. 2013). Unsupervised learning is useful for discovering hidden structures within data (Bishop and Nasrabadi 2006), and reinforcement learning excels in scenarios requiring sequential decision-making and dynamic interaction with an environment (Kaelbling et al. 1996). Every type of machine learning contributes distinct qualities to different applications, showing the flexibility and power of ML methods in addressing a wide range of challenges across different fields (LeCun et al. 2015).

2.3 Common Machine Learning Models Used

Machine learning (ML) models play a critical role in addressing regression and classification problems in this dissertation. The nature of the problem guides the choice of models, the data's characteristics, and the application's specific requirements. This section provides an

overview of the ML models used, divided into two main categories: regression and classification models.

2.3.1 Regression Models

Regression models are used to predict continuous outcomes based on input features. In this dissertation, several regression models were employed to address various problems in atmospheric sciences and communication technologies.

2.3.1.1 Linear Regression

Linear regression is one of the simplest and most widely used models for regression analysis. It assumes a linear relationship between the input features and the output variable (Montgomery et al. 2021). The model is represented by the equation:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n + \varepsilon \tag{2.1}$$

where y is the predicted output, β_0 is the intercept, β_i (for $i = 1, 2, \ldots, n$) are the coefficients or weights of the model, x_n are the input features, and ε is the error term.

The primary objective of linear regression is to estimate the coefficients (β_i) such that the sum of the squared differences between the predicted values (y) and the actual values is minimized. This method, known as Ordinary Least Squares (OLS), ensures that the model provides the best fit to the data (Montgomery et al. 2021).

Linear regression is particularly powerful for its simplicity and clarity. It provides clear insights into the relationship between the dependent and independent variables. For instance, in atmospheric sciences, it can be used to model the relationship between temperature and humidity levels, providing coefficients that indicate the strength and direction of the relationship (Kutner et al. 2005). However, linear regression also has limitations. It assumes a linear relationship between the variables, which may not always be true in real-world scenarios. Additionally, it is sensitive to outliers, which can negatively influence the model's

parameters. Despite these limitations, linear regression remains a valuable tool, especially when the relationship between variables is approximately linear, and the data is free from significant outliers (Montgomery et al. 2021).

2.3.1.2 Regression Decision Trees

A regression decision tree is a multivariable method used to model and predict continuous outcomes by splitting the data into subsets based on feature values. Unlike some models that might exclude events if certain conditions are unmet, a decision tree evaluates multiple conditions to accurately classify events. Each branch in the tree represents a decision rule based on feature values, and each leaf node represents a predicted continuous value (Loh 2011). The regression tree algorithm typically involves several key steps: setting the accuracy criterion for predictions, choosing splits, deciding when to stop splitting, and determining the optimal tree size (Hastie et al. 2009). The first step involves establishing the criterion for prediction accuracy. Common criteria include cross-validation, re-substitution error, and test sample error. The re-substitution error, calculated using the same data used to generate the prediction p, is defined as follows (Loh 2011):

$$E\left(p\right) = \frac{1}{N}\sum_{i=1}^{N}\left(u_i - p\left(v_i\right)\right)^2 \tag{2.2}$$

where N is the number of samples, u_i is the actual value of the target for the i_{th} data point, and $p(v_i)$ represent the predicted value from the model for the i_{th} data point, where v_i represents the features of the i_{th} data point.

For cross-validation error, the dataset is divided into k smaller, nearly equal-sized samples. One of these samples is used to build the predictor p, and the cross-validation error is computed as follows:

$$E^{CV}\left(p\right) = \frac{1}{N_k}\sum_{k}\sum_{\left(u_i,v_i\right)\in X_k}\left(v_i - p^{(k)}\left(u_i\right)\right)^2 \tag{2.3}$$

Here, $p(k)$ is derived from the small sample X_k. The test sample error is determined by splitting the dataset into two subsamples: X_1 with size N_1 and X_2 with size N_2. The error for the test sample is calculated as:

$$E^{ts}(p) = \frac{1}{N_2} \sum_{(u_i,v_i)\in X_k}^{N} \left(v_i - p\left(u_i\right)\right)^2 \tag{2.4}$$

where X_2 is the subsample not used for building the predictor.

The regression tree technique involves determining the optimal splits to estimate continuous variable values. Splits are quantified using a node impurity measure, which indicates the relative homogeneity within terminal nodes. A common measure for regression trees is the least-squared deviation, which minimizes the variance within each node. The node impurity $R(t)$ is defined as:

$$R(t) = \frac{1}{N_w(t)} \sum_{i\in t} w_i f_i \left(u_i - \bar{v}(t)\right)^2 \tag{2.5}$$

where $N_w(t)$ is the weighted number of instances in node t, w_i is the weight of the $i-th$ instance, f_i is the frequency variable, u_i is the response variable, and $\bar{v}(t)$ is the weighted mean of node t.

The decision to stop splitting is typically based on a predefined criterion, such as the minimum number of samples in a node, the maximum depth of the tree, or the minimum decrease in node impurity required to continue splitting. These criteria prevent the tree from overfitting the training data by growing too complex and capturing noise. Once the tree is grown, the next step is determining its optimal size, often called pruning. Pruning involves cutting back the tree to remove nodes that provide little to no additional predictive power, thereby simplifying the model. The optimal tree size is usually achieved through cost-complexity pruning, which balances the tree's complexity against its predictive accuracy. The aim is to find the smallest tree with the least error (Loh 2011).

Regression decision trees provide multiple advantages, including interpretability, the ability to handle non-linear relationships, and the flexibility to manage different data types without expensive data preprocessing. Compared to other regression methods, such as linear

regression and neural networks, regression decision trees provide a balance between simplicity and flexibility. However, they have drawbacks, such as the potential to overfit if not adequately pruned, and their sensitivity to small variations in the data, which might result in various splits and hence different trees (Costa and Pedreira 2023).

2.3.1.3 Ensemble Trees

Ensemble learning is a powerful technique in machine learning that combines multiple models to improve overall performance, robustness, and accuracy. Among the various ensemble methods, bagged (Bootstrap Aggregating) trees and boosted trees are particularly effective for regression and classification tasks. These approaches exploit the advantages of individual decision trees while reducing their drawbacks, such as overfitting due to high variance (Breiman 1996; Hastie et al. 2009).

Bagged Trees (Bootstrap Aggregating): Bagging, or Bootstrap Aggregating, is an ensemble technique designed to reduce the variance of a model by averaging the predictions of multiple decision trees. The process involves generating several bootstrap samples from the original dataset, training a decision tree on each sample, and then aggregating the predictions from all the trees (Breiman 1996).

The procedure for bagging can be summarized as follows: First, bootstrap sampling involves creating multiple subsets of the original dataset by random sampling with replacement. Each subset, or bootstrap sample, is the same size as the original dataset but contains duplicate instances, ensuring that each tree in the ensemble is trained on a slightly different dataset and promoting diversity among the trees. Next, a decision tree is trained on each bootstrap sample in the training phase. These trees are typically grown to their full depth without pruning, allowing them to capture the nuances of the data. By fully growing the trees, they become highly specialized to the bootstrap samples, capturing detailed patterns that may be missed in a single tree (Hastie et al. 2009). Finally, in the aggregation phase, the

final prediction is obtained for regression tasks by averaging the predictions of all individual trees. For classification tasks, the final prediction is determined by majority voting, where each tree votes for a class, and the class with the most votes is chosen as the final prediction.

This approach leads to more robust and accurate models, especially when decision trees overfit the training data (Breiman 1996).

Boosted Trees: Boosting is an effective ensemble strategy that enhances the performance of a model by training a number of decision trees sequentially. Each succeeding tree is designed to repair the faults made by the previous trees. Unlike bagging, which builds trees independently, boosting builds trees in a dependent manner, with each tree learning from the mistakes of the previous ones (Freund and Schapire 1997; Hastie et al. 2009).

The boosting process involves the following steps: Initially, start with an initial prediction, which is often the mean of the target variable in regression tasks, serving as the baseline model. In the next step, a decision tree will be trained on the residual errors from the previous model. These residuals differ between the actual values and the current ensemble model's predictions. Each new tree learns to predict these residuals, thereby addressing the errors made by the model up to that point. Finally, the new tree is added to the ensemble model, updating the predictions by integrating the previous predictions with the new tree's output. This process continues for several iterations or until the model reaches a desired accuracy level. A learning rate parameter can adjust the influence of each new tree on the overall prediction, controlling the contribution of each tree to the model (Friedman 2001).

Boosting can significantly improve decision tree accuracy by reducing bias and variance. However, careful tuning of hyperparameters, such as the learning rate and the number of iterations, is required to prevent overfitting. Boosting is particularly effective in handling complex, non-linear relationships and has been successfully applied in various domains, including finance, metrology, healthcare, and natural language processing. The ability to boost

focus on difficult cases and iteratively improve the model's performance makes it a powerful tool for predictive modeling (Freund and Schapire 1997; Hastie et al. 2009).

Comparison and Applications: Both bagging and boosting are powerful ensemble methods that enhance the performance of decision trees. Bagging primarily reduces variance by averaging multiple independent trees, making it suitable for high-variance, low-bias scenarios. In contrast, Boosting reduces bias and variance by sequentially improving the model's accuracy, making it ideal for scenarios where the initial model has high bias.

2.3.1.4 Neural Networks for Regression

Artificial neural networks (ANNs) are powerful prediction, classification, pattern recognition, and clustering models. They are competitive with traditional statistical and regression models, offering high-speed processing and considerable promise for parallel applications (Goodfellow et al. 2016; Mitchell 1997). ANNs have been widely used in natural language processing and image recognition tasks due to their self-learning, fault tolerance, non-linearity, adaptivity, and advanced input-to-output mapping capabilities (Bishop and Nasrabadi 2006).

ANNs mimic the human brain's information processing system. They consist of interconnected neurons organized into an input layer, one or more hidden layers, and an output layer. The input layer receives raw data, with each neuron representing a feature of the input data. Hidden layers perform computations by applying weighted sums to inputs, adding bias terms, and using activation functions such as sigmoid, tanh, or ReLU (Rectified Linear Unit). The non-linearity introduced by these activation functions enables the network to capture complex relationships between inputs and outputs (Bishop and Nasrabadi 2006).

The output layer produces the final prediction, a continuous value in regression tasks. Training a neural network involves forward propagation, where input data generates predictions, and backward propagation, where errors are propagated back to update weights and

biases. This process uses the backpropagation algorithm and gradient descent optimization to minimize a loss function, leading to accurate predictions (LeCun et al. 2015).

Figure 2.1 depicts a basic neural network design consisting of input, hidden, and output layers.

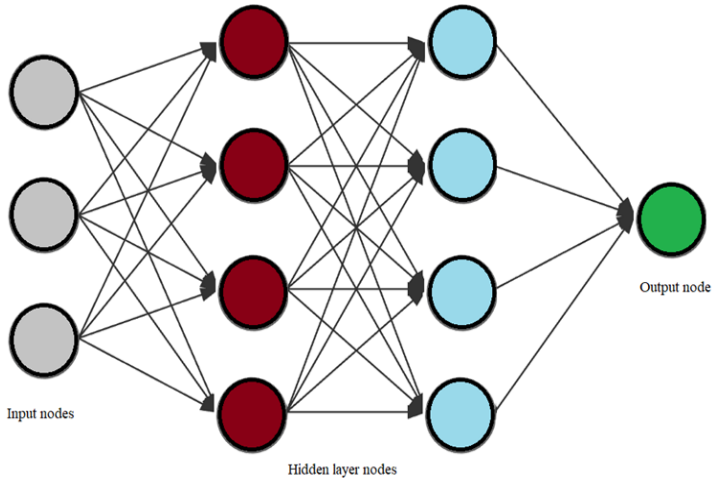


Figure 2.1: An example of a basic neural network design.

Despite the advantages, ANNs have some drawbacks. They can be very demanding regarding computational resources and susceptible to overfitting. Training them often requires a large amount of labeled data. Additionally, they lack transparency and face problems like vanishing and exploding gradients, affecting their performance (Goodfellow et al. 2016).

This dissertation employs the ReLU function for all neural networks due to its effectiveness in backpropagation and gradient descent, preventing gradient disappearance and explosion.

2.3.2 Classification Models

Classification is a core task in machine learning that involves predicting categorical labels for given inputs based on their features. Unlike regression, which predicts continuous values, classification assigns inputs to discrete classes or categories. This task is widely used in

various domains, such as healthcare, finance, marketing, and natural language processing. In classification, a model learns from a dataset where each example is described by features and a corresponding class label to accurately assign labels to new, unseen examples. Classification problems can be binary, where there are two possible classes (e.g., spam or not spam), or multi-class, where there are three or more classes (e.g., handwritten digit recognition from 0 to 9) (Hastie et al. 2009; Murphy 2012).

2.3.2.1 k-Nearest Neighbors

The k-Nearest Neighbors (k-NN) algorithm is a simple and powerful classification method in machine learning. It is non-parametric and lazy, making no assumptions about the data distribution, and defers computation until classification. k-NN is popular for its simplicity, ease of implementation, and flexibility. It classifies an input data point based on the majority class of its k closest neighbors in the feature space. The "k" represents the number of neighbors considered during classification (Murphy 2012).

The steps involved in k-NN classification are: First, the user must select an appropriate value for k. A small value of k (e.g., k=1) can make the model sensitive to noise, while a large value can smooth out decision boundaries but might miss local patterns. For a given test data point, the algorithm computes the distance between this point and all points in the training dataset. Common distance metrics include Euclidean distance, Manhattan distance, and Minkowski distance. Euclidean distance, defined as $d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$, is the most commonly used metric, where p and q are data points in the feature space. The algorithm then sorts all the distances and selects the k data points in the training set that are closest to the test point. Finally, it assigns the test point to the class that appears most frequently among the k nearest neighbors. In a tie, various tie-breaking strategies can be applied, such as selecting the class with the smallest average distance (Hastie et al. 2009).

One of the key benefits of k-NN is its simplicity. It is easy to understand and implement, making it a popular choice for beginners in machine learning. Additionally, k-NN can

handle multi-class classification problems effectively and works well with non-linear decision boundaries, as it makes decisions based on the local neighborhood of the data points. However, k-NN also has some limitations. Its computational cost can be high, especially with large datasets, as it requires calculating the distance between the test point and all training points. Another limitation is that k-NN is sensitive to the scale of the data; therefore, it is important to standardize or normalize the features before applying the algorithm to ensure that all features contribute equally to the distance calculations (Murphy 2012).

The choice of k is crucial for the performance of k-NN. A small k value can lead to overfitting, while a large k value can lead to underfitting. Typically, the optimal value of k is determined through cross-validation. In practical applications, k-NN has been used in various domains, such as image recognition, which classifies images based on pixel intensity patterns, and recommendation systems, which suggest items to users based on similar preferences. It is also widely used in medical diagnosis, classifying patients based on similarities in symptoms and test results (Bishop and Nasrabadi 2006).

2.3.2.2 Logistic Regression

Logistic regression is a fundamental classification algorithm widely used in statistics and machine learning. It is particularly effective for binary classification problems, where the goal is to classify input data into one of two possible categories (Mitchell 1997). Logistic regression estimates the likelihood that a given input belongs to a particular class. It operates by fitting a logistic function, also known as the sigmoid function, to the input data. The sigmoid function, defined as $\sigma(z) = \frac{1}{1+e^{-z}}$, maps any real-valued numbers into the interval (0, 1). This property makes it ideal for modeling probabilities (Hosmer Jr et al. 2013). The logistic regression model starts with a linear combination of the input features, similar to linear regression:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n \tag{2.6}$$

where β_0 is the intercept, β_i are the coefficients and x_i represents the input features. This linear combination is then passed through the sigmoid function to produce a probability:

$$P\left(y=1|x\right)=\sigma\left(z\right)=\frac{1}{1+e^{-(\beta_0+\beta_1x_1+\beta_2x_2+...+\beta_nx_n)}} \tag{2.7}$$

The model predicts the class label based on this probability. A common decision rule for binary classification is to classify the input as class 1 if the probability is greater than or equal to 0.5 and as class 0 otherwise (Bishop and Nasrabadi 2006). One of the key strengths of logistic regression is its interpretability, allowing users to understand the influence of each feature on the outcome (Hosmer Jr et al. 2013).

Logistic regression is typically trained using maximum likelihood estimation. The likelihood function measures how likely it is that the model generated the observed data. The coefficients are estimated by maximizing this likelihood function, which is equivalent to minimizing the negative log-likelihood. This optimization is usually performed using algorithms such as gradient descent (Goodfellow et al. 2016). In addition to its simplicity and interpretability, logistic regression has several other advantages: it is computationally efficient, can be trained quickly even on large datasets, and is less prone to overfitting than more complex models. Regularization helps to prevent overfitting by adding a penalty to the magnitude of the coefficients, encouraging simpler models (Hosmer Jr et al. 2013).

The probabilistic output of logistic regression provides a measure of confidence in the predictions, which can be useful in many applications, such as medical diagnosis and fraud detection. However, logistic regression also has limitations. It assumes a linear relationship between the input features and the log odds of the outcome, which may not hold true for more complex datasets. Additionally, it can struggle with highly imbalanced datasets, where one class is much more frequent than the other. Techniques such as resampling, adjusting class weights, or using more complex algorithms may be necessary in such cases (Hosmer Jr et al. 2013).

2.3.2.3 Classification Decision Trees

Decision trees in classification are known for their simplicity and effectiveness in dealing with both linear and non-linear data relationships. A classification decision tree works by iteratively splitting data into subsets depending on feature values and then assigning a class label to each terminal node or leaf (Quinlan 1986).

A classification decision tree structure involves a root node, internal nodes, branches, and leaf nodes. The root node represents the entire dataset, and each internal node corresponds to a feature. Branches represent the decision rules, and leaf nodes represent the final class labels. Building a classification decision tree involves selecting the optimal feature and threshold for each split to maximize the separation between classes (Loh 2011).

Classification decision trees use impurity measures such as Gini impurity and entropy to determine the best splits. Gini impurity measures the likelihood of an incorrect classification of a randomly chosen element if it was randomly labeled according to the distribution of labels in the node (Loh 2011). It is defined as:

$$Gini(t) = 1 - \sum_{i=1}^{C} p_i^2 \tag{2.8}$$

where p_i is the proportion of instances of class i in the node, and C is the number of classes. A node with a lower Gini impurity is preferred as it indicates a purer node with a higher concentration of a single class. Entropy is another measure of node impurity that quantifies the amount of disorder or uncertainty (Quinlan 1986). It is defined as:

$$Entropy(t) = -\sum_{i=1}^{C} p_i log_2\left(p_i\right) \tag{2.9}$$

where p_i is the proportion of instances of class i in the node. The goal is to achieve a high information gain, which is the reduction in entropy after the dataset is split on an attribute. Information gain is calculated as the difference between the entropy of the parent node and the weighted sum of the entropies of the child nodes:

$$Information\ Gain = Entropy(parent) - \sum_{k=1}^{K} \frac{|t_k|}{|t|}\ Entropy\left(t_k\right) \tag{2.10}$$

where t is the parent node, t_k are the child nodes, and K is the number of child nodes. Once a tree is built, it can categorize new instances by going through it from root to leaf node and applying the decision rules at each node. The new instance's class label is the leaf node's label (Quinlan 1986).

Decision trees offer several advantages for classification tasks. They are highly understandable, allowing users to understand each feature's decision-making process and importance. This makes decision trees particularly valuable in fields where transparency and explainability are crucial, such as healthcare and finance (Loh 2011). However, decision trees also have some limitations. They are susceptible to overfitting, especially when the tree can grow to its full depth without any constraints. Overfitting can lead to poor generalization of new data (Loh 2011). Moreover, decision trees can be sensitive to small changes in the data, resulting in significantly different trees being generated (Loh 2011).

In practical applications, decision trees are used in various domains, including medical diagnosis, where they help classify patients based on symptoms and test results, and customer segmentation, where they help businesses identify and target different customer groups (Quinlan 1986).

2.3.2.4 Neural Networks for Classification

Neural networks (NNs) are powerful and flexible machine learning models that have proven to be highly effective for classification tasks. Unlike regression neural networks, classification neural networks assign discrete class labels to input data. This makes them ideal for various applications, from image and speech recognition to medical diagnosis and spam detection (LeCun et al. 2015).

The core of neural networks for classification is the same foundational structure as regression neural networks, which are interconnected neurons organized into layers. However,

for classification tasks, the output layer and activation functions are explicitly designed to handle discrete class labels (Goodfellow et al. 2016).

The output layer of a classification neural network usually consists of one neuron for each class. For binary classification, a single output neuron with a sigmoid activation function is commonly used to produce a probability score between 0 and 1, representing the likelihood of the input belonging to the positive class. For multi-class classification, the output layer typically uses a softmax activation function, which converts the output scores into a probability distribution over the classes (Bishop and Nasrabadi 2006). The softmax function is defined as:

$$softmax\left(z_i\right) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \tag{2.11}$$

where z_i is the output score for class i, and C is the total number of classes. The class with the highest probability is then selected as the predicted class.

Training a neural network for classification involves using a loss function appropriate for classification tasks. For binary classification, the binary cross-entropy loss function is commonly used:

$$Binary\ CrossEntropy\ Loss = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i log\left(p_i\right) + \left(1-y_i\right)\ log\left(1-p_i\right)\right] \tag{2.12}$$

where y_i is the true label, p_i is the predicted probability, and N is the number of samples. For multi-class classification, the categorical cross-entropy loss function is used:

$$Categorical\ CrossEntropy\ Loss == -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} log\left(p_{ij}\right) \tag{2.13}$$

where y_{ij} is a binary indicator (0 or 1) if class label j is the correct classification for sample i, p_{ij} is the predicted probability for class j for sample i, and C is the total number of classes (Murphy 2012). These loss functions measure the divergence between the true labels and the predicted probabilities, guiding the optimization process during training. The training process uses algorithms like stochastic gradient descent (SGD) and its variants, like

Adam, to minimize the loss function. The backpropagation algorithm computes the gradients of the loss function with respect to the network's weights, allowing the optimization algorithm to update the weights iteratively and improve the model's performance (LeCun et al. 2015).

Neural networks are effective for classification because they can model complex, non-linear decision boundaries (Goodfellow et al. 2016). Neural networks' strengths include scalability and adaptability. They handle large datasets with many features, learning from vast amounts of data for accurate predictions. Neural networks have transformed fields such as healthcare, where they diagnose diseases from medical images, and finance, where they detect fraudulent transactions. Their ability to extract features from raw data makes them invaluable for developing intelligent systems (LeCun et al. 2015).

2.4 Metaheuristic Optimization Algorithms for Data Quality Control

Optimization algorithms are computational methods designed to find the best possible solution, or an approximate solution, to complex problems by maximizing or minimizing an objective function. These algorithms are essential in various fields, including engineering, economics, logistics, and machine learning, to find optimum solutions under specified constraints (Amaireh et al. 2020a,b, 2022; SS and HS 2022).

Metaheuristic algorithms form a class of optimization techniques that are particularly effective for solving high-dimensional, non-linear, and multi-modal problems where traditional methods may struggle. They are termed "metaheuristics" because they provide a higher-level strategy to guide other heuristics, ensuring that the search process is both explorative and exploitative. Nature-inspired algorithms draw their principles from natural processes and phenomena, such as biological evolution, animal behavior, and physical laws. This inspiration helps design strategies that mimic these natural processes to solve optimization problems. Examples include Genetic Algorithms (GA) inspired by natural selection, Particle

Swarm Optimization (PSO) inspired by bird flocking behavior, and Ant Colony Optimization (ACO) inspired by the searching behavior of ants (Deb 2011; Mirjalili 2015).

Metaheuristic algorithms are considered a subset of artificial intelligence (AI) methods because they involve intelligent search strategies and adaptive mechanisms to solve complex problems. These algorithms can learn from the environment and improve their performance over time, aligning with AI's core principles. They are used in various AI applications such as machine learning for hyperparameter tuning, feature selection, and optimizing neural network architectures; robotics for path planning and control optimization; engineering for design optimization and resource allocation; healthcare for optimizing treatment plans and medical diagnoses; and finance for portfolio optimization and fraud detection (SS and HS 2022).

This dissertation employed a hybrid optimization technique combining Antlion Optimization (ALO) and Grasshopper Optimization Algorithm (GOA) for data cleaning. This hybrid approach leverages the strengths of both ALO and GOA to create a more robust and effective optimization process. Antlion Optimization mimics the hunting mechanism of antlions. The Antlion Optimization (ALO) algorithm mimics the way antlions dig traps in the sand to catch ants. This algorithm steers candidate solutions towards the most promising areas within the search space. ALO focuses on the exploitation process and developing solutions to improve its quality. On the other hand, the Grasshopper Optimization Algorithm (GOA) takes inspiration from the swarming behavior of grasshoppers, particularly their instinct to move towards food sources. GOA emphasizes exploration, allowing the algorithm to cover a broad search space and avoid getting stuck in local optima too early (Amaireh et al. 2023c).

The hybrid algorithm integrates ALO's exploitation capabilities and GOA's exploration strengths, resulting in a comprehensive optimization technique. The process begins with initializing a population of candidate solutions represented by ants and grasshoppers. These solutions undergo movements influenced by both ALO and GOA mechanisms. The antlions' traps guide the solutions towards optimal regions (exploitation), while the grasshoppers'

swarming behavior ensures diverse coverage of the search space (exploration) (Amaireh et al. 2023c). Applying this hybrid metaheuristic optimization algorithm to data cleaning significantly improves input data quality. Cleaner data enhances the accuracy and reliability of machine learning models, boosting predictive capabilities and operational efficiency.

2.5 Evaluation Metrics for ML Models

Evaluating the performance of machine learning models is crucial to ensure their accuracy, reliability, and effectiveness. Various metrics are used depending on the type of problem—regression or classification. This section provides an overview of the key metrics used to evaluate the machine learning models in this dissertation. A combination of these evaluation metrics is used to comprehensively assess the machine learning models' performance. For regression tasks, metrics like MAE, MSE, RMSE, and R^2 are employed to measure the accuracy of continuous predictions. For classification tasks, accuracy, precision, and recall are utilized to evaluate the models' ability to correctly classify instances and handle imbalanced data (Bishop and Nasrabadi 2006; Hastie et al. 2009).

2.5.1 Evaluation Metrics for Regression

2.5.1.1 Correlation Coefficient (ρ)

The correlation coefficient (ρ) measures the strength and direction of the linear relationship between predicted and actual values. It ranges from -1 to 1, where 1 indicates a perfect positive linear relationship, -1 indicates a perfect negative linear relationship, and 0 indicates no linear relationship. It is calculated as:

$$\rho\left(X_{\text{true}}, X_{\text{predicted}}\right) = \frac{1}{N-1} \sum_{i=1}^{N} \left(\frac{X_{\text{true}}(i)-\mu_{\text{true}}}{\sigma_{\text{true}}}\right) \left(\frac{X_{\text{predicted}}(i)-\mu_{\text{predicted}}}{\sigma_{\text{predicted}}}\right) \quad (2.14)$$

In this equation, N represents the number of data points. μ_{true} and $\mu_{\text{predicted}}$ represent the mean values of the true and predicted data sets, respectively, while σ_{true} and $\sigma_{\text{predicted}}$ are their corresponding standard deviations (Murphy 2012).

2.5.1.2 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) measures the average magnitude of errors in a set of predictions without considering their direction. It provides a straightforward measure of how far predictions are from actual outcomes on average. MAE is calculated as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|X_{true}\ (i) -\ X_{predicted}\ (i)\right| \tag{2.15}$$

where N is the total sample count, X_{true} the observed values, and $X_{predicted}$ the predicted values, respectively. MAE is simple to understand and clearly indicates the average prediction error. It is particularly useful when the cost of errors does not depend on their direction or magnitude (Hastie et al. 2009).

2.5.1.3 Mean Squared Error (MSE)

Mean Squared Error (MSE) measures the average squared difference between the estimated and actual values. By squaring the errors, MSE gives more weight to larger errors, making it sensitive to outliers. This can be advantageous when large errors are particularly undesirable (Hastie et al. 2009). MSE is defined as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(X_{true}\ (i) -\ X_{predicted}\ (i)\right)^2 \tag{2.16}$$

2.5.1.4 Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) is the square root of the MSE. RMSE provides an error metric on the same scale as the original data, making it easier to interpret. RMSE is calculated as:

$$RMSE = \sqrt{\left[\left(\sum_{i=1}^{N}\left(X_{true}\,(i) - \ X_{predicted}\,(i)\right)^2\right)/N\right]} \tag{2.17}$$

RMSE is helpful for measuring the magnitude of prediction errors and is particularly sensitive to significant errors, similar to MSE, but is more interpretable due to its same-scale nature (Hastie et al. 2009).

2.5.1.5 Coefficient of Determination (R^2)

The coefficient of determination, or R-squared (R^2), measures the proportion of the variance in the dependent variable that is predictable from the independent variables. It indicates the goodness-of-fit of the model and is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(X_{true}\ \ (i) - \ X_{predicted}\ \ (i)\right)^2}{\sum_{i=1}^{N}\left(X_{true}\ \ (i) - \ \bar{X_{true}}\right)^2} \tag{2.18}$$

where $X_{true}\ \ (i)$ is the actual value, $X_{predicted}\ \ (i)$ is the predicted value, X_{true} is the mean of the actual values, and N is the number of samples. (R^2) values range from negative infinity to 1. An (R^2) value of 1 indicates that the model perfectly fits the data, explaining 100% of the variance in the dependent variable. An (R^2) value of 0 means that the model does not explain any of the variance in the dependent variable, effectively performing no better than a horizontal line at the mean of the dependent variable. Negative values of (R^2) indicate that the model performs worse than a horizontal line at the mean, suggesting that the model is not appropriate for the data (Hastie et al. 2009).

2.5.2 Evaluation Metrics for Classification

2.5.2.1 Accuracy

Accuracy is one of the most straightforward and commonly used metrics for evaluating classification models. It represents the proportion of correct predictions among the cases examined. The formula for accuracy is defined as:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.19}$$

where TP (True Positives) are the cases where the model correctly predicts the positive class. TN (True Negatives) are the cases where the model correctly predicts the negative class. FP (False Positives) are the cases where the model incorrectly predicts the positive class. FN (False Negatives) are the cases where the model incorrectly predicts the negative class.

Accuracy is intuitive and easy to interpret, making it a popular choice for evaluating model performance. However, accuracy can be misleading, especially in class imbalance. In datasets where one class significantly outnumbers the other, a model that always predicts the majority class can achieve high accuracy while failing to capture the minority class. For example, in a dataset with 95% of instances belonging to one class and 5% to another, a model that always predicts the majority class would achieve 95% accuracy but fail to identify any instances of the minority class (Bishop and Nasrabadi 2006).

2.5.2.2 Additional Evaluation Metrics

Additional metrics such as Total Cost, Positive Predictive Value (PPV), False Discovery Rate (FDR), True Positive Rate (TPR), and False Negative Rate (FNR) are essential to evaluate the classification model performance (Bishop and Nasrabadi 2006). The Total Cost, a metric frequently unnoticed, measures the comprehensive expenses linked to misclassifications. This is crucial in operational contexts where prediction inaccuracies may lead to financial losses or

safety risks. Positive Predictive Value (PPV), also known as precision, measures the model's ability to categorize positive instances properly (Bishop and Nasrabadi 2006):

$$PPV = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2.20}$$

The False Discovery Rate (FDR) is the complement of PPV and displays the proportion of false positives within predicted positives (Bishop and Nasrabadi 2006):

$$FDR = \frac{\text{False Positives}}{\text{True Positives} + \text{False Positives}} \tag{2.21}$$

The True Positive Rate (TPR), or Sensitivity, evaluates the model's ability to detect all true positive instances (Bishop and Nasrabadi 2006):

$$TPR = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2.22}$$

In contrast, False Negative Rate (FNR) is the rate at which actual positives are wrongly predicted as negatives (Bishop and Nasrabadi 2006).

$$FNR = \frac{\text{False Negatives}}{\text{True Positives} + \text{False Negatives}} \tag{2.23}$$

Chapter 3

Atmospheric Humidity Estimation from Wind Profiler Radar

3.1 Introduction

Estimating the atmospheric humidity profile is essential for various fields, including weather prediction, climate studies, aviation safety, agriculture, hydrology, and environmental monitoring (Chowdhury et al. 2021; Guinn and Barry 2016; Harel et al. 2014; Hartmann 2015; Jacob 2001; O'Connor et al. 2021; Sherwood 2010; Trenberth and Guillemot 1998). Accurate estimates of the humidity profile can help improve decision-making and planning in these fields and ultimately lead to a better understanding of the earth's atmosphere and its role in the global climate system (Barreca 2012; Davis et al. 2016). The challenge of retrieving low-medium (up to 10 km) atmosphere humidity using Doppler wind profiler radar has been discussed in numerous previous studies (Adab et al. 2013; Bianco et al. 2005; Chen and Bradley 2006; Eccel 2012; Imura et al. 2007; Klaus et al. 2006; Kuo et al. 1994; Saïd et al. 2018; Stankov et al. 2003; Tsuda et al. 2001).

Most of the existing methods depend on the physical models, which are based on a set of assumptions about the atmosphere's behavior, and they rely on precise input data to make accurate predictions. On the other hand, the physical models are limited in their accuracy and adaptation capability due to the interactions of uncertainties in the physical

parameters, the limited data availability, and the model structure limitations (Brajard et al. 2021). For example, these models require input values for physical parameters such as temperature, pressure, and wind speed, but the measurement of these parameters is often uncertain, leading to small input faults and significant model prediction errors (Olafsson and Bao 2020). Additionally, the models require large amounts of data and can be biased by data gaps and errors, which may not accurately reflect the behavior of the real-world system. These limitations could result in physical models providing unreliable or inaccurate predictions of atmospheric humidity profiles. In such cases, alternative methods, such as machine learning (ML) algorithms, may be desirable to offer humidity profile estimations based on less dependence on physical modeling.

Machine learning (ML) algorithms capable of constructing complex, non-linear relationships between input variables from vast data sets had diverse applications across energy, environmental engineering, and atmospheric predictions (Amar et al. 2022a, 2021, 2022b, 2019; Hassanien et al. 2014; Ng et al. 2022; Talebkeikhah et al. 2020; Wuest et al. 2016). The unique characteristics of ML algorithms are the flexibility and reducing necessity for accurate physical parameters (Hassanien et al. 2014). Therefore, ML algorithms supplement traditional physical models through their adaptability to unknown and dynamic environments.

Exploring machine learning methods for atmospheric parameter prediction is an active field of study. For example, Artificial Neural Networks (ANN) have been utilized for refractivity prediction in several studies (Javeed et al. 2018). However, these studies do not address the specific needs of profiler radar sensing. Other studies have demonstrated the application of machine learning to weather forecasting using various types of meteorological data. For instance, linear and functional regression models have been employed to forecast maximum and minimum temperatures for seven days using two days' worth of weather data (Holmstrom et al. 2016). Similarly, an hourly rainfall forecast model based on a Support Vector Machine (SVM) was developed to predict rainfall with high temporal resolution

and accuracy (Zhao et al. 2021). For satellite remote sensing, ML has been integrated to improve data interpretation. A standalone cloud detection algorithm was designed for the Microwave Humidity Sounder-2 (MWHS-2) satellite sensors, utilizing a gradient-boosting decision tree. This algorithm was trained on observations from China's new-generation weather radar (CINRAD) (Liu et al. 2020). Looking at the incorporation of Global Navigation Satellite System-Radio Occultation (GNSS-RO) data, machine learning has been leveraged to forecast wind fields in the Beijing-Tianjin-Hebei region of China (Chu et al. 2022). Initial processing established relationships between thermodynamic and kinetic parameters through historical monitoring data. This step was followed by predictions using ML models, such as Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and Deep Neural Networks (DNN). ML has also been applied to estimate relative humidity with specific resolutions. A study used Random Forest and XGBoost to estimate daily near-surface relative humidity at a 1-km resolution over Japan and South Korea, with separate schemes for clear and cloudy sky conditions, has been reported in (Park et al. 2021).

This chapter explores a novel approach using ML for improved humidity estimation. Initially, the clutter return is separated from useful atmospheric echoes through spectrum processing to obtain basic wind estimation and clutter spectrum properties. Subsequently, the profiler estimation outputs generate a set of "features." These higher-level features, along with some low-level moment estimations from spectrum processing, are combined as training features for machine learning (ML). The "bagged tree," selected through tuning and comparison with other algorithms, is applied as the optimal ML algorithm solution. Based on the evaluations, it proves to be the most effective and computationally efficient algorithm. A key novelty of this research is the demonstration that radar profiler moment data alone is sufficient to achieve reasonably accurate humidity estimations without relying on additional input parameters such as temperature. This approach simplifies the estimation process while maintaining accuracy. Furthermore, this study introduces the "cascading ML" approach, which was applied for the first time in humidity estimation. This technique includes an

intermediate training stage that utilizes "synthetic" pressure data, further enhancing the prediction accuracy of the final model. Notably, this work estimates relative humidity (RH) without using temperature as an intermediate input, marking a significant departure from traditional methods. This innovation reduces the model's complexity and potential sources of error, making it a robust solution for atmospheric humidity estimation.

This chapter is organized as follows: Section 3.2 summarizes the instrumentation, data collection, and existing methods based on physical models. In Section 3.3, 'Approach and Methodology,' the proposed approach is described in detail. This section includes the following subsections: Processing Flow, General Climatology of Hong Kong, Selection of the Machine Learning Solutions, Data Quality Control and Preprocessing, Description of the Dataset, and Prevention of Overfitting. Results and discussion based on the available datasets are presented in Section 3.4, while Section 3.5 provides conclusions and suggestions for future work.

3.2 Instrumentation and Physical Modeling Methods

The Hong Kong Observatory's (HKO) wind profilers operate at 1299 MHz and locate at multiple locations in Hong Kong. Two of the three profilers are installed at the airport, while the third is installed at the city center. A profile has three beams, one vertical and two 50-degree tilted from the vertical. These three beams can support retrieval of the three-dimensional vector wind (u, v, w). Raw I/Q data can be collected, and it is useful for estimating the vector wind profiles. On the other hand, spectral moment data directly from the profiler outputs can be used to study atmospheric turbulence parameters, which are related to the RH profiles. For the radiosonde truth data, HKO uses Vaisala RS41-SG radiosondes to perform daily routine soundings for both automatic (AUTOSONDE) and manual launches. These radiosondes are used to measure relative humidity, temperature,

and pressure with 2% to 4% range of accuracies. The sounding data is collected every 6 minutes. In addition to information on relative humidity, temperature, and pressure, it also provides information such as wind speed, wind direction, and dew points. The data covered up to an altitude of around 10 km (Amaireh et al. 2023a). The HKO used the measurement data collected by the city center's profiler for training and testing. The combination of the wind profiler data and the radiosonde data provide a comprehensive view of the atmospheric conditions, which improves weather parameters forecasting.

The classical method of estimating the humidity profile is based on the following equations 3.1 and 3.2 (Saïd et al. 2018). In these equations, q_0 is the humidity at level z_0. T represents the temperature, P represents the atmospheric pressure, and $\theta = (1000/P)^{\frac{2}{7}}$. M is the vertical (z-direction) gradient of the atmosphere refractivity. C_n^2 is the turbulence structure parameter, which can be estimated from the zeroth radar moment (or total power) of radar return signals. α^2 is a scalar constant dependent on specific regions. $\frac{dV}{dz}$ is the vertical shear of the horizontal wind vector, which can be estimated from the second radar moment (wind velocity estimations). Turbulent kinetic energy ϵ is directly related to the third radar moment or spectrum width.

$$q(z) = \theta^2 \int_{z_0}^{z} (1.67 \times 10^{-6} \frac{MT^2}{P} + \frac{1}{7750} \frac{d\theta}{dz}) \theta^{-2} dz + q_0 \tag{3.1}$$

$$C_n^2 = \alpha^2 \frac{\epsilon^{2/3} M^2}{(\frac{dV}{dz})^2} \tag{3.2}$$

Although Equations 3.1 and 3.2 are used as the initial guidance for RH retrieval algorithms, they highlight the importance of accurate estimations of the spectrum moments. They also inspire the application of machine learning algorithms in the way that temperature, pressure, or radar spectrum moments might be directly used as feature vectors for humidity retrieval. However, many factors, such as clutter, noise, and equipment quality,

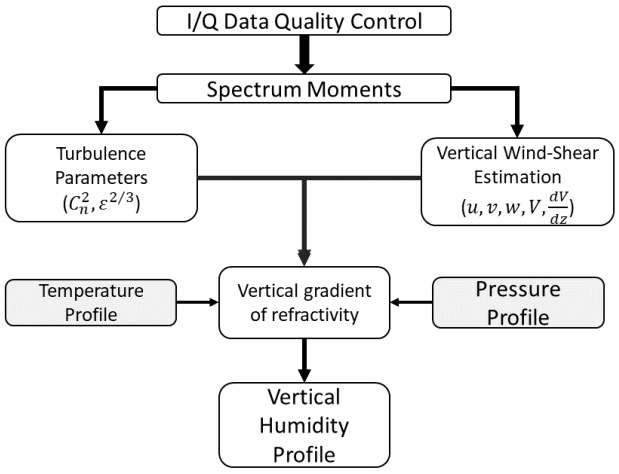
affect the precise estimation of radar moments. Even with preprocessing to enhance estimation performance, physical models still present many challenges. Consequently, this approach focuses on a machine-learning solution that is inspired by the parameters of physical models.

3.3 Approach and Methodology

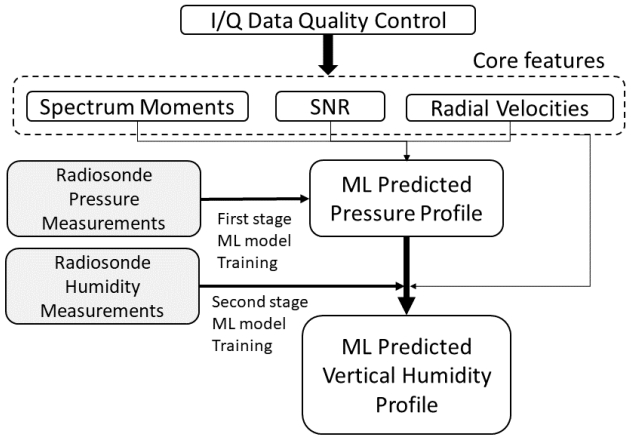
3.3.1 Processing Flow

The overall technical solution is depicted in Figure 3.1. Initially, classic Doppler radar signal processing is used to accurately estimate the Doppler spectrum and wind estimations for different altitudes (up to 10 km). Spectral data from the three beams of the wind profiler are utilized to estimate three moments: power (zeroth moment), velocity (first moment), and spectrum width (second moment). Subsequently, clutter return is separated from useful atmospheric echoes through spectrum processing, providing both basic wind estimation and clutter spectrum properties. Finally, the profiler estimation outputs and other in-situ probe information, such as pressure, are combined to generate a set of training features for machine learning (ML).

In the context of the complex physical model outlined in Equations 3.1 and 3.2, the conventional method described in the literature involves first estimating wind vectors and then calculating turbulence parameters as shown in Figure 3.1. However, the absence of access to vendor-specific and proprietary information on profiler processing algorithms poses a challenge to obtaining validated wind vector estimations. To address this, a new approach was explored that shifts the "entry-point" of machine learning (ML) closer to the raw spectrum data. Instead of performing precise wind estimation, this method directly uses the moment data from the raw spectrum as ML feature variables. The moments calculated from the three beams of the wind profiler serve as feature variables for ML models to predict relative humidity directly. This method avoids reliance on precise wind estimation and parameters such as temperature and pressure from radiosonde data, which may not always be available.



(a)



(b)

Figure 3.1: Comparison of (a) traditional physical model-based method, and (b) the cascaded ML method for humidity profile estimation.

In a two-step cascading process, the moment data is first used to predict pressure. Then, the predicted pressure and moment data are combined to predict relative humidity (RH) without the need for temperature data. This cascaded ML solution simplifies the model and reduces computational demands for RH estimation by omitting temperature, typically a critical element in humidity prediction.

The moments from the level-II radar profiler are used as inputs. To ensure the integrity of the processing pipeline, the processing procedure includes a range of data cleaning steps, such as outlier removal and smoothing, which are detailed in the subsequent subsections. The feature engineering phase was exploratory and iterative. It involved exploring potential features and assessing their correlation with the target variables of pressure and humidity. A set of features was selected based on their optimal correlation with the target variables and their wide variability. An innovative aspect of the study is the creation and application of the "synthetic" pressure data. Following a rigorous data-cleaning process, the algorithm begins with selecting suitable moment data and other features for pressure prediction. Next, various ML algorithms are trained to achieve an optimal pressure estimation. This optimal pressure estimation is an additional input feature for humidity estimation. The moment data and the predicted pressure are then cleaned again to ensure optimal correlation with relative humidity. These high-quality input features are then used to train different ML algorithms to achieve the most accurate possible humidity prediction.

3.3.2 General Climatology of Hong Kong

Understanding the climatology trend of the region will help us better understand data and algorithm verification. The first half of 2021, from January to June, was unusually warm, primarily due to the four months' worth of temperatures that were much above average. The average maximum, the average minimum, and the entire time average temperature were 26.3 C^O, 23.3 C^O, and 21.3 C^O, respectively. For January 2021, the early half experienced cooler temperatures than the second half, which was comparatively warmer. The mean temperature for the month was 16.2 C^O, 0.3 degrees cooler than the average of 16.5 C^O. It was drier and sunnier in January 2021 than typical. Total sunlight hours for the month were 217.3, which is 49 percent more than the average of 145.8 hours. In the month, there was very little rain. February 2021 was significantly sunnier and warmer in Hong Kong than usual due to the northeast monsoon across southern China being less than typical for the majority

of the month. The average maximum and minimum temperatures were 23.5 and 17.5 C^O, respectively, which are 4.1$C^O$and 2.2 C^O higher than the corresponding normal. More than double the average of 101.7 hours, the total number of hours of bright sunlight in February was 205.1. As a result, the winter in Hong Kong during December 2020, January 2021, and February 2021 was hotter than typical, mainly due to the unusually sunny and warm weather. March 2021 in Hong Kong continued to be unusually warm despite fewer outbreaks of cold air from the north. The monthly average minimum and maximum temperatures were 22.0 $C^O$and 24.8 C^O, respectively. These three temperatures were 2.5 C^O and 2.9 C^O, above their respective normals, and were the maximum monthly average values for March on record. The total rainfall for the month was only 3.5 millimeters, or around 5% of the average amount of 75.3 millimeters, making it significantly drier than typical. April 2021 stayed substantially warmer than usual. The average monthly minimum temperature was 22.4 C^O, and the average monthly maximum temperature was 27.0 C^O. These values were 1.3 degrees and 1.4 degrees higher than normal. With a total rainfall of just 32.5 mm, or around 21 percent of the average of 153.0 millimeters, April 2021 was also significantly drier than usual due to the dominance of upper-air anticyclones across southern China for the majority of the month. May 2021 was the warmest May in Hong Kong, primarily due to the stronger-than-normal subtropical ridge across southern China. The monthly average low temperature of 27.0 $C^O$was recorded for May and was 2.5 degrees beyond its respective normal. The average high temperature, 32.1 C^O, was 3.3 degrees higher than average. Hong Kong saw the hottest spring in history from March to May 2021, along with unusually warm temperatures in March and April 2021. With rainfall data of only 65.0 mm, May was also significantly drier than typical. June 2021 had rainfall totals of 628.0 mm, roughly 28% more than the average of 491.5 mm. The mean temperature was 28.8 C^O, which is 0.5 C^O higher than the average of 28.3 C^O, making it hotter than typical. In this month, three tropical cyclones passed across the South China Sea and the western North Pacific (Hong Kong Observatory 2021). During these months, Hong Kong experienced varying mean relative humidity levels. The

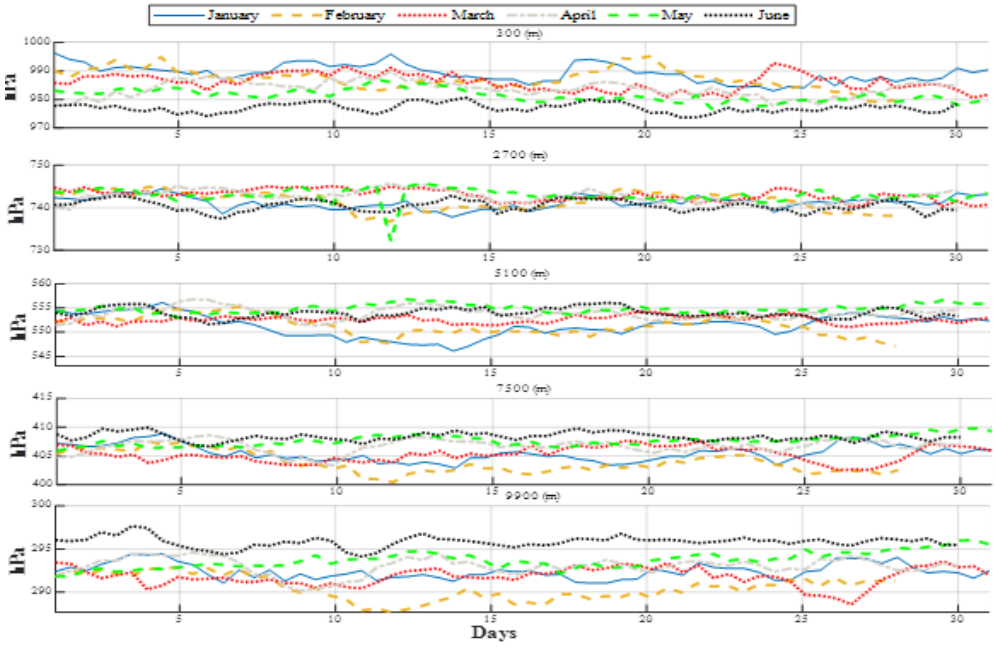
mean relative humidity in January was 62%, increasing to 75% in February. March and April recorded a mean relative humidity of 79%, while May slightly decreased to 78%. The highest mean relative humidity was observed in June, reaching 82% (Hong Kong Observatory 2021). Figure 3.2 shows the pressure and humidity values from radiosonde data for the study region during the first half of 2021 at different altitudes (from 0 to around 10 km).

3.3.3 Selection of the Machine Learning Solutions

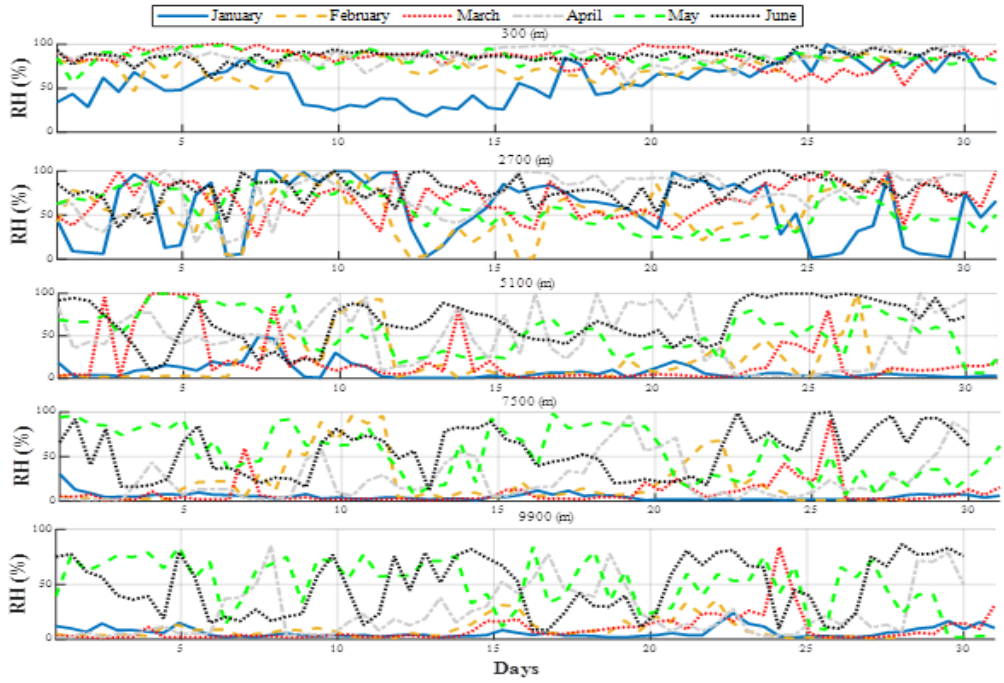
The selection of ML solutions included model complexity, ease of interpretation, computational efficiency, and prediction accuracy. The Bagged Ensemble Tree (BET), Coarse Tree, and Wide Neural Network (WNN) emerged as the primary algorithms for this work. However, several alternative methods were also examined and compared to validate these choices and gain a broader perspective.

The Bagged Ensemble Tree (BET) was ultimately chosen as the optimal solution for this study due to its superior ability to manage high-dimensional data and its natural resistance to overfitting. As an ensemble method, the BET combines the predictions of multiple decision trees, thereby reducing variance and enhancing model accuracy. This approach proved particularly effective in handling the complex, non-linear relationships present in atmospheric humidity data. While Bagged Trees can demand substantial computational resources, especially when dealing with huge datasets, their accuracy and robustness in this context outweighed these concerns, making them the preferred choice for this study.

In contrast, the Coarse Tree, known for its simplicity and interpretability, facilitates easier visualization and understanding of data relationships. This straightforwardness reduces the risk of overfitting, which is crucial given the intricate nature of atmospheric data. However, this simplicity might also restrict its ability to understand complex, non-linear relationships between features, potentially compromising accuracy. The Wide Neural Network model was chosen for its proficiency in modeling non-linear relationships. With a single layer and numerous neurons, the WNN successfully captures the complicated interplay between



(a)



(b)

Figure 3.2: Comparison (a) pressure values (kPa), (b) relative humidity values (%) of the region with different times (month) and different altitudes.

atmospheric parameters. However, it might struggle to effectively capture more intricate hierarchical representations that deeper networks handle better. Also, careful tuning of neurons is needed to prevent overfitting.

The models and their chosen hyperparameters are shown in Table 3.1.

Table 3.1: Details of the Hyperparameters of the proposed machine learning models.

Model Type	**Hyperparameters**
Interactions Linear Regression (ILR)	Terms: Interactions
Fine Tree (FT)	Minimum leaf size= 4
Medium Tree (MT)	Minimum leaf size= 12
Coarse Tree (CT)	Minimum leaf size= 36
Boosted Ensemble Tree (BoostedET)	Minimum leaf size= 8; Number of learners: 30
Bagged Ensemble Tree (BET)	Minimum leaf size: 8; Number of learners: 30
Narrow Neural Network (NNN)	1 layer; layer size= 10; Activation: ReLU; Iteration limit: 1000
Medium Neural Network (MNN)	1 layer; layer size= 25; Activation: ReLU; Iteration limit: 1000
Wide Neural Network (WNN)	1 layer; layer size= 100; Activation: ReLU; Iteration limit: 1000
Bilayered Neural Network (BiNN)	2 layers; layers' size= 10; Activation: ReLU; Iteration limit: 1000
Trilayered Neural Network (TriNN)	3 layers; layers' size= 10; Activation: ReLU; Iteration limit: 1000

3.3.4 Data Quality Control and Prepossessing

The outliers and missing numbers could impact the ML algorithm's reliability and validity, leading to inaccurate results. Detecting and handling missing variables and outliers to maintain the quality and validity of study results can be accomplished using various strategies, such as estimating missing values, eliminating outliers, or changing the data to a more appropriate scale (Kwak and Kim 2017). Missing data can either be discarded during preprocessing or substituted with values calculated using statistical algorithms. In the current study, all missing values have been removed to ensure accurate and reliable results. Outliers must either be modified once their sources have been found or replaced with replacement values. In this work, all the outliers have been removed (similar to (Kwak and Kim 2017)). After removing the outliers and missing data from the complete dataset, the data was further smoothed to enhance the correlation between the input features and the targets. The

moving median smoothing method, also known as the linear Gaussian filter(Arce 2005), was used in this study. It is remarkably robust against uncommon events, such as sudden shocks, which can be well-handled by the Laplace distribution (Simonoff 2012). Furthermore, this method can enhance the correlation between the input features and targets.

3.3.5 Description of the Dataset

A six-month dataset that includes both Doppler spectral moments and radiosonde data was utilized for verification. This dataset spans 49 different heights, from 300 m to 9900 m, at intervals of 200 m. In the Doppler spectral data, samples were recorded every one minute and forty-one seconds. Radiosonde data samples were collected at 12 PM and 12 AM daily, with each set containing samples taken every two seconds until the 47th minute. The data was carefully matched to the corresponding height, date, and time to ensure accuracy and reliability in the results. Over the six months, this dataset provides a comprehensive understanding of the pressure and relative humidity patterns. To effectively utilize this information, a cross-validation approach with K=5 was employed to assess the performance of the proposed machine learning algorithms. The cross-validation dataset comprised 85% of the 1,406,688 unique samples, while the remaining 15% served as the testing dataset to validate the accuracy of the algorithms. Thus, the complete dataset was divided into a cross-validation set with 1,195,688 samples and a testing set with 211,000 samples.

3.3.6 Prevention of Overfitting

To prevent overfitting, a strategy was adopted to apply the K-fold cross-validation technique with K set to 5. This technique divides the data into five distinct subsets, running a training cycle and validating the model five times, using a different subset as validation data. This comprehensive approach gives a reliable estimate of how the model might perform on unseen data. Moreover, out of the 1,406,688 unique samples, a majority, 85% or 1,195,688 samples, were used in the cross-validation dataset, with the remaining 15% (211,000 samples) set aside

as a testing dataset. This large set of previously unseen data, held throughout the model's training phase, helps correctly measure the model's performance and provides a safety net against overfitting. Furthermore, early stopping was used during the model's training process. This strategy continuously evaluates the model's performance on the validation set during training and stops the procedure if the validation error rises. This technique protects against overfitting caused by intrinsic noise in training data. As a result, these metrics balanced the trade-off between bias and variance, which leads to limiting models' overfitting.

3.4 Results and Discussion

The evaluation process begins by analyzing the entire six-month dataset, followed by an examination of data from each individual month. Both evaluations are compared and analyzed to provide a comprehensive understanding of the pressure and relative humidity patterns, as well as the performance of the proposed machine learning algorithms. Each evaluation includes a subsection where pressure values are estimated using regression techniques with Doppler spectral moments as input features. Subsequently, a cascading ML algorithm estimates relative humidity (RH) using the same features, augmented by the predicted pressure values from the first step. These experiments were conducted on a desktop computer equipped with an i7-2600K CPU @ 3.40 GHz and 24 GB of RAM.

3.4.1 Analysis of Results from Six-Months Data

3.4.1.1 Pressure Prediction

Tables 3.2 and 3.3 summarize comparisons of the performance of various machine learning algorithms, evaluated using six different metrics, namely root mean squared error (RMSE) (kPa), mean squared error (MSE) (kPa^2), correlation coefficient, mean absolute error (MAE) (kPa), R-squared, prediction speed (observations/second), and training time in seconds. Table 3.2 presents the results for the cross-validation dataset, while Table 3.3 provides the

results for the testing dataset. The best four RMSE and MSE values (in kPa and kPa^2, respectively) achieved by the ML methods for the predicted pressure for the cross-validation dataset are (87.09 kPa, 7585.12 kPa^2), (92.43 kPa, 8542.43 kPa^2), (92.55 kPa, 8564.87 kPa^2), and (94.54 kPa, 8938.46 kPa^2) for the Bagged Ensemble Tree (BET), Coarse Tree (CT), Medium Tree (MT), and Wide Neural Network (WNN), respectively. Therefore, these models demonstrate the highest accuracy in evaluating prediction errors for the cross-validation dataset and show the smallest discrepancies between the predicted values and actual values. Moreover, the achieved MAE values (in kPa) of the same ML algorithms are 61.42, 64.03, 64.07, and 64.67 for the Bagged Ensemble Tree (BET), Coarse Tree (CT), Fine Tree (FT), and Medium Tree (MT), respectively. Furthermore, based on cross-validation dataset analysis, the correlation coefficients for the best four machine learning models are 0.91, 0.9, 0.89, and 0.88 for FT, BET, MT, and CT, respectively. The tree models and the BET offer the best correlation outcomes. The best three R-squared values of the utilized models for the cross-validation dataset are 0.78, 0.75, and 0.75 for the BET, MT, and CT, respectively. This demonstrates that the BET model explains 78% of the variability found in the pressure variable. Linear Regression, on the other hand, performs poorly, implying that the selection criteria are not linear.

On the other hand, the best four RMSE and MSE values (in kPa and kPa^2, respectively) achieved for pressure estimation on the testing dataset over the entire 6-month period are (86.64 kPa, 7505.88 kPa^2), (92.26 kPa, 8511.95 kPa^2), (92.28 kPa, 8516.51 kPa^2), and (93.63 kPa, 8766.93 kPa^2) for the WNN, BET, Narrow Neural Network (NNN), and Boosted Ensemble Tree (BoostedET), respectively. Further, the MAE values (in kPa) of the same ML algorithms were 60.98, 63.19, 63.56, and 64.36 for the WNN, BoostedET, BET, and NNN, respectively. These results prove that the BET and WNN methods have better stability than the others in having low estimation errors for pressure parameters in cross-validation and testing datasets. According to the testing dataset evaluation, the correlation coefficients for the top four machine learning models were 0.91, 0.9, 0.89, and 0.88 for BoostedET, WNN,

BET, and NNN, respectively. As can be seen, the ensemble tree and neural network models produce the highest performance in terms of correlation. The top three R-squared values for the BET, WNN, and NNN for the testing dataset were 0.78, 0.78, and 0.76, respectively, indicating that the BET and WNN models can account for 78% of the variability in the pressure variable.

In terms of computational speed, all of the employed neural networks can predict up to 1445728 observations per second, the linear regression model can estimate roughly 652257 observations per second, the decision trees can estimate approximately 603824 observations per second, and the ensemble trees can handle 128835 observations per second. However, the neural network methods and the FT have longer training times than the other models, with the WNN, which has 100 neurons, being the slowest. In contrast, the linear regression and ensemble trees were the fastest models during training, taking around 37 seconds and 560 seconds, respectively.

In summary, the Bagged Ensemble Tree was found to have the optimal overall performance. So it is used to estimate pressure value, which are then used as an input feature to estimate the relative humidity (RH) in the next step.

Table 3.2: Comparison of performance of the proposed machine learning models in predicting pressure for the six months of cross-validation data.

Model Type	RMSE (Valid) (kPa)	MSE (Valid) (kPa^2)	ρ (Valid)	R^2 (Valid)	MAE (Valid) (kPa)	Prediction Speed (obs/sec)	Training Time (sec)
ILR	108.52	11776.85	0.81	0.66	80.03	652257.94	37.51
FT	94.57	8942.63	0.91	0.74	64.03	417087.25	12216.47
MT	92.55	8564.87	0.89	0.75	64.07	419819.87	4188.86
CT	92.43	8542.43	0.88	0.75	64.67	603824.96	561.64
BoostedET	103.59	10730.38	0.85	0.69	79.77	128835.49	551.78
BET	87.09	7585.12	0.9	0.78	61.42	23874.35	2615.18
NNN	102.05	10413.95	0.85	0.7	75.11	860477.63	5651.68
MNN	97.41	9487.97	0.85	0.73	70.59	1018433.15	10905.88
WNN	94.54	8938.46	0.86	0.74	67.86	1187889.67	18863.05
BiNN	98.15	9633.41	0.85	0.72	71.53	951444.23	9758
TriNN	97.39	9483.95	0.85	0.73	70.57	1445728.81	13432.38

Table 3.3: Comparison of performance of the proposed machine learning models in predicting the pressure for the six months of test data.

Model Type	RMSE (Test) (kPa)	MSE (Test) (kPa^2)	MAE (Test) (kPa)	ρ (Test)	R^2 (Test)
ILR	118.06	13937.82	87.64	0.78	0.6
FT	109.2	11923.67	80.42	0.81	0.66
MT	120.34	14481.19	85.79	0.77	0.58
CT	109.2	11923.67	80.42	0.81	0.66
BoostedET	93.63	8766.93	63.19	0.91	0.75
BET	92.26	8511.95	63.56	0.89	0.78
NNN	92.28	8516.51	64.36	0.88	0.76
MNN	104.01	10819.05	79.96	0.84	0.69
WNN	86.64	7505.88	60.98	0.9	0.78
BiNN	98.97	9794.77	71.87	0.85	0.72
TriNN	97.44	9493.95	70.71	0.85	0.73

Partial Dependency Relationship (PDR) plots are shown in Figure 3.3 to better illustrate the relationship between the input features and the target response. This plot demonstrates how the predicted target response varies with changes in the input features. For example, the expected pressure increases as the received power increases above -40 dBm. In the case of the WNN model, there is a linear relationship between the velocity and the pressure when the velocity is less than 0. On the other hand, the BET and MT models exhibit an exponential relationship between the SNR and the pressure, mainly when the SNR is around -25 dB. Conversely, all models demonstrate an inverse relationship between the spectrum width and the pressure values. These observations highlight the importance of considering the impact of individual input features on the predicted target response.

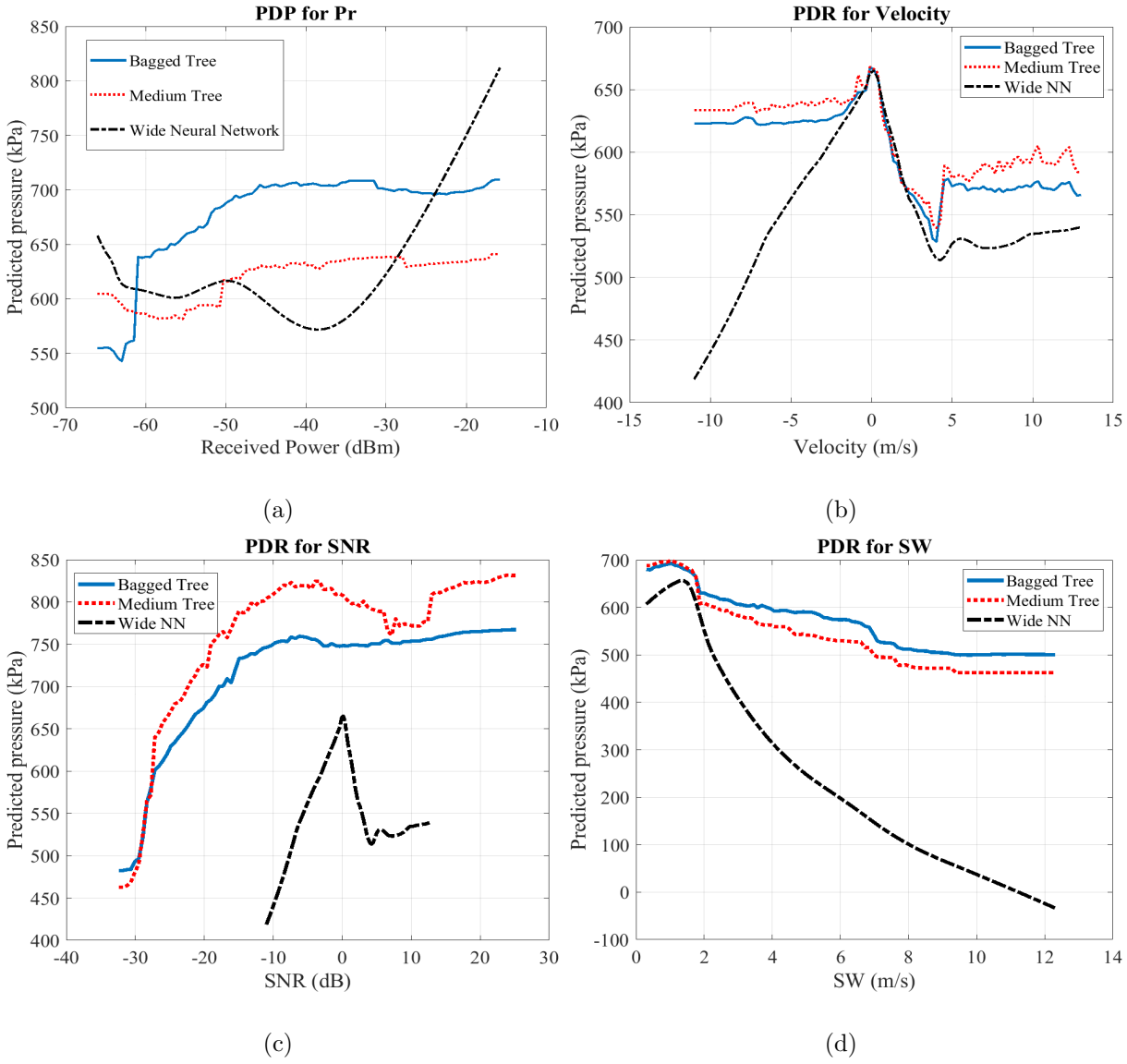


Figure 3.3: Partial Dependency Relationship (PDR) plots between predicted pressure and inputs features of the six-month dataset. PDR for (a) Pr and predicted pressure, (b) Velocity and predicted pressure, (c) SNR and predicted pressure, and (d) Spectrum Width (SW) and predicted pressure.

Scatter plots of truth and the estimated pressure values are displayed as density figures with color maps in two ways to address the issue of many sample points in the plots. Figure 3.4 shows typical scatter density plots with color maps of the estimated pressure by the multiple regression models versus the actual pressure data. Figure 3.5 presents 2D histogram

plots that include mini squares with color maps inside the plots, representing the density of predicted versus true response values categorized into ten classes. All these figures describe the use of cross-validation data over the six months. According to these plots, a slight improvement can be observed for the bagged tree model compared to the other two mentioned models: coarse tree and wide neural network. This is because the scatter points for the Bagged Tree are more concentrated around the center than those for the other models. Figures 3.6 and 3.7 show normal scatter density plots with a color map and 2D histogram plots of the predicted pressure by the three ML models vs. the true pressure values for the testing dataset. Similar to the cross-validation dataset, the bagged tree model performs better than CT and WNN because the points are clustered closer to the center.

As shown in Figure 3.8, three pressure profiles are computed sequentially from the predicted pressure values using the BET, CT, and WNN and then compared to simultaneous radiosonde profiles. This figure shows how well the proposed ML algorithms can predict the pressure at different heights from 0 to 10 km, which is evident that the Bagged tree could estimate the pressure values at various altitudes with high accuracy, significantly below 7 km. Similar observations can be seen in both WNN and coarse tree plots. Finally, it is worth mentioning that all the plotted profiles are random samples from the six-month dataset to prove the ability of these ML methods to predict the pressure values during different seasons and times.

3.4.1.2 Humidity Estimation

Next, the same dataset of six months, including the four input features along with the previously predicted pressure, is used to estimate the relative humidity (RH). The data is split into two datasets: cross-validation, with 1,195,688 samples, and testing, with 211,000 samples. The same eleven machine learning models with the same hyperparameters were used in the pressure prediction stage to train and test the dataset. Table 3.4 and Table 3.5 show

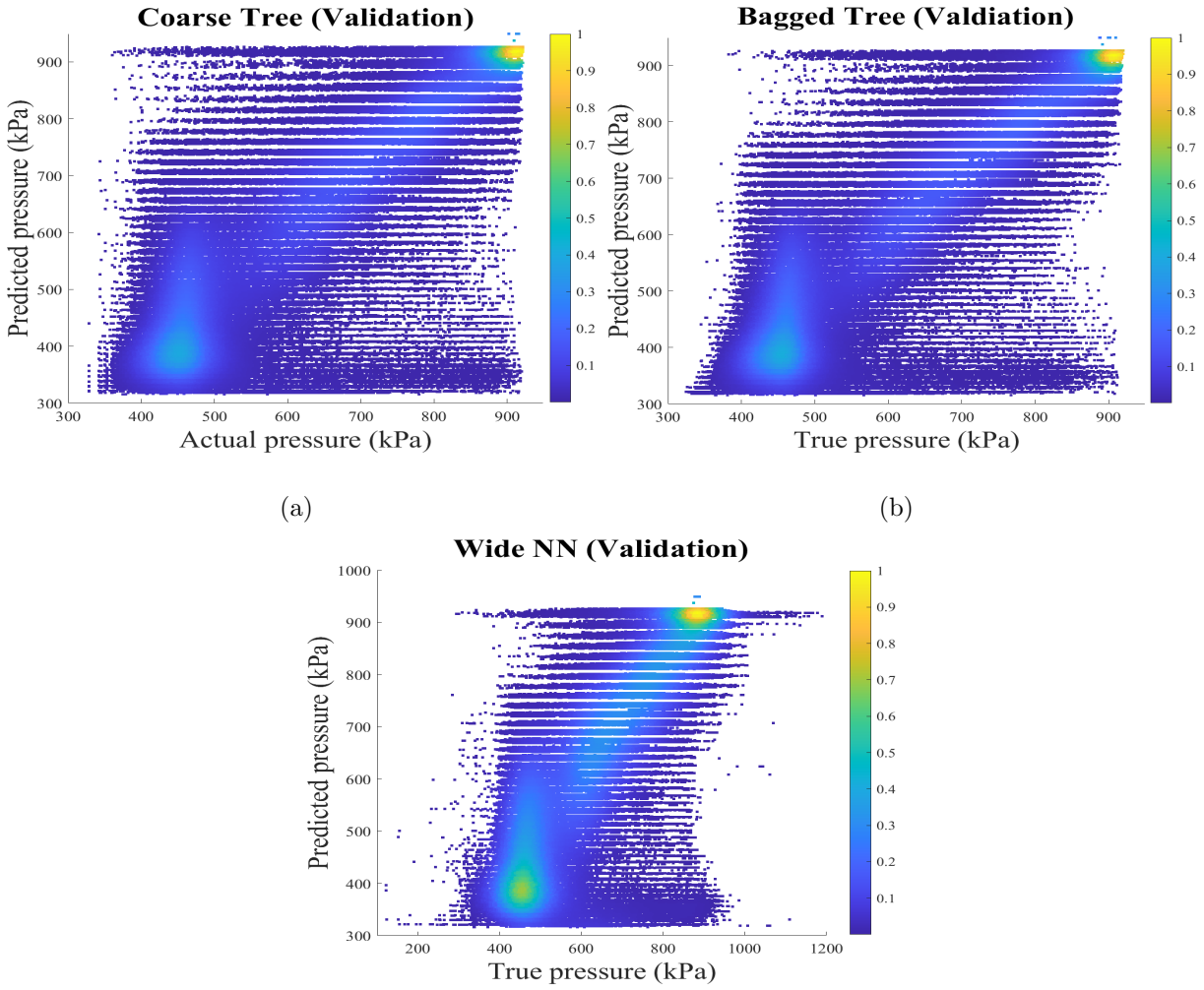


Figure 3.4: Scatter density graphs with color maps of predicted vs. actual pressure of the (all months) validation dataset using different ML methods; (a) Coarse tree, (b) Bagged tree, (c) Wide Neural Network (WNN).

the performance details of the machine learning algorithms in terms of the six performance metrics for the cross-validation and the testing datasets, respectively.

The top three RMSE and MSE results of the cross-validation dataset based on the relative humidity (RH) are (22.2%, 493.02%^{2}), (24.17%, 584.14%^{2}), and (24.47%, 598.7%^{2}), respectively, corresponding to the BET, CT, and WNN methods. On the other hand, the ML algorithms with the best MAE performances are BET, MT, and FT, with result values

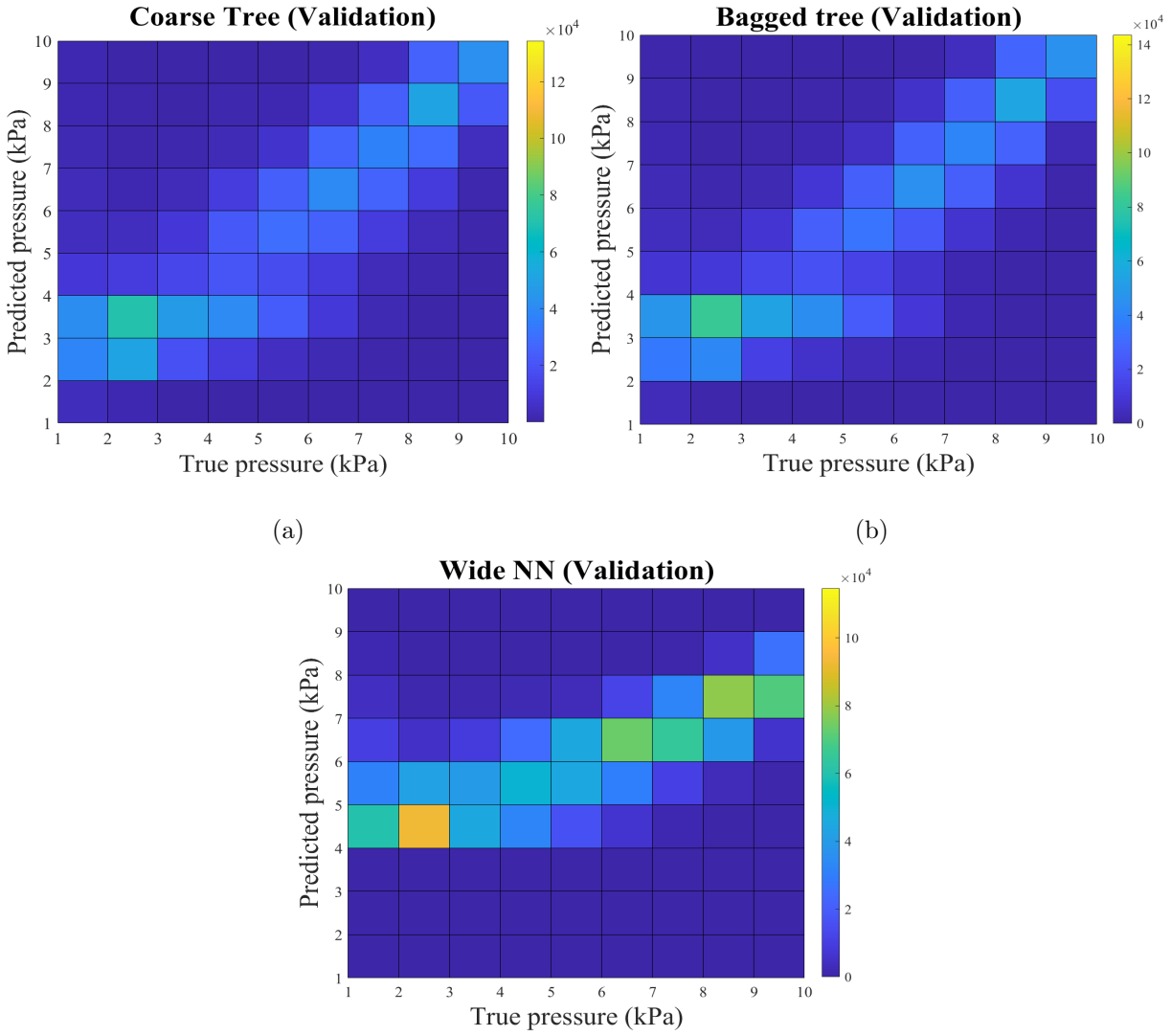
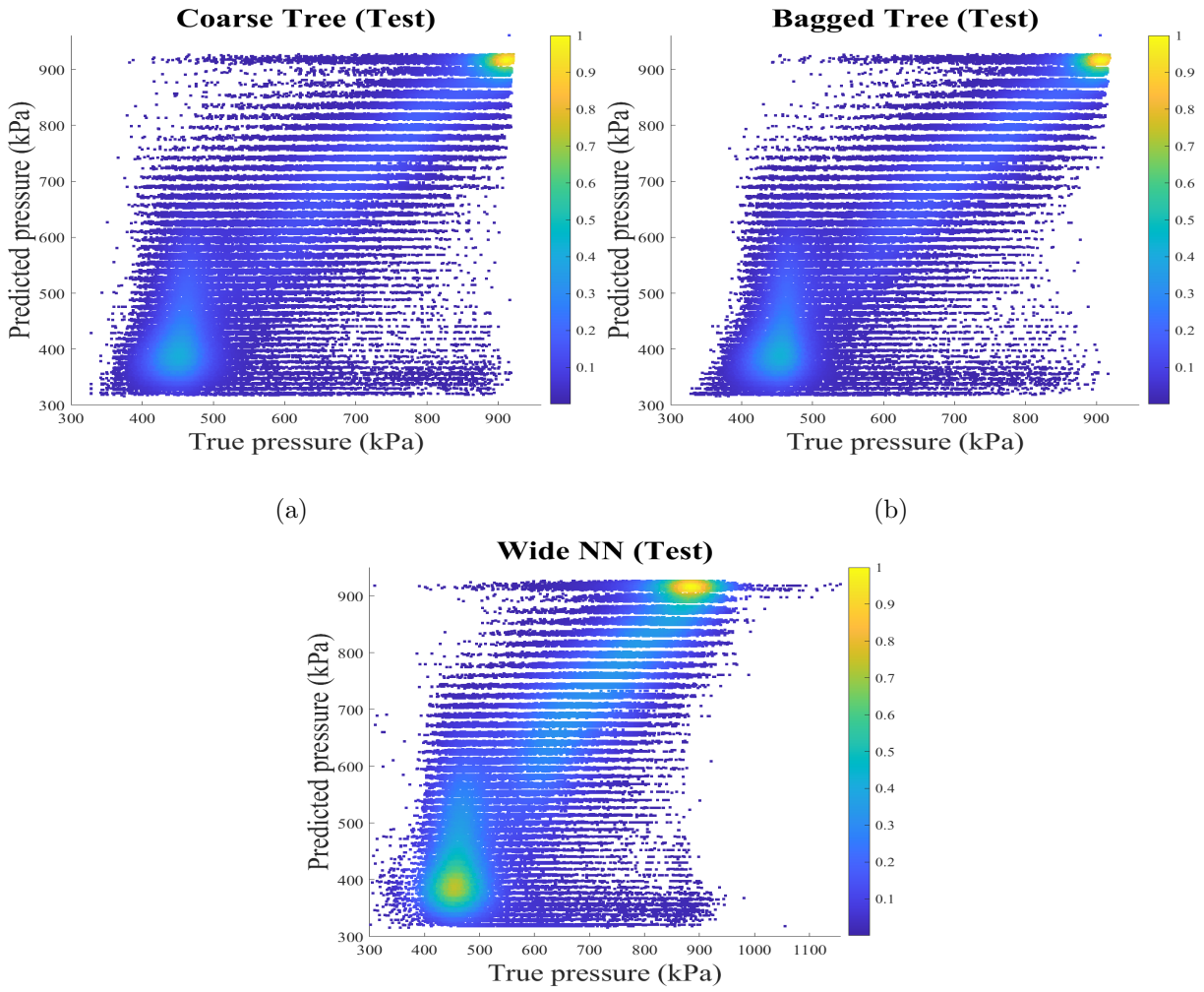


Figure 3.5: 2D histogram plots with color maps representing the density of predicted versus true pressure values categorized into ten classes for the validation dataset using various ML algorithms: (a) Coarse Tree, (b) Bagged Tree, and (c) Wide Neural Network (WNN).

of 17.44%, 18.76%, and 18.88%, respectively. In addition, the FT and BET models had the highest correlation coefficients at 0.86 and 0.83, respectively, with a significant difference from the other ML approaches. Moreover, the best achievable R-squared value for the BET in the cross-validation dataset is 0.48, indicating that it explains 48% of the variability seen in the humidity.



(a) (b)

(c)

Figure 3.6: Scatter density graphs with color maps of expected vs. actual pressure of the (all months) test dataset generated by the following machine learning algorithms; (a) coarse tree, (b) Bagged tree, and (c) Wide Neural Network (WNN).

For the testing dataset, the ML methods that achieved the minimal RMSE and MSE values of the predicted humidity are (21.9%, 479.51%^{2}), (24.04%, 578.03%^{2}), and (24.5%, 600.05%^{2}) using BET, CT, and MT. Meanwhile, the best achieved MAE values are 17.14%, 18.43%, and 18.45% for the BET, FT, and MT. The BET, MT, and CT algorithms obtained the best correlation coefficient values, which are 0.70, 0.62, and 0.62, respectively. As can be noticed, the ensemble and decision tree models have the highest correlation coefficient.

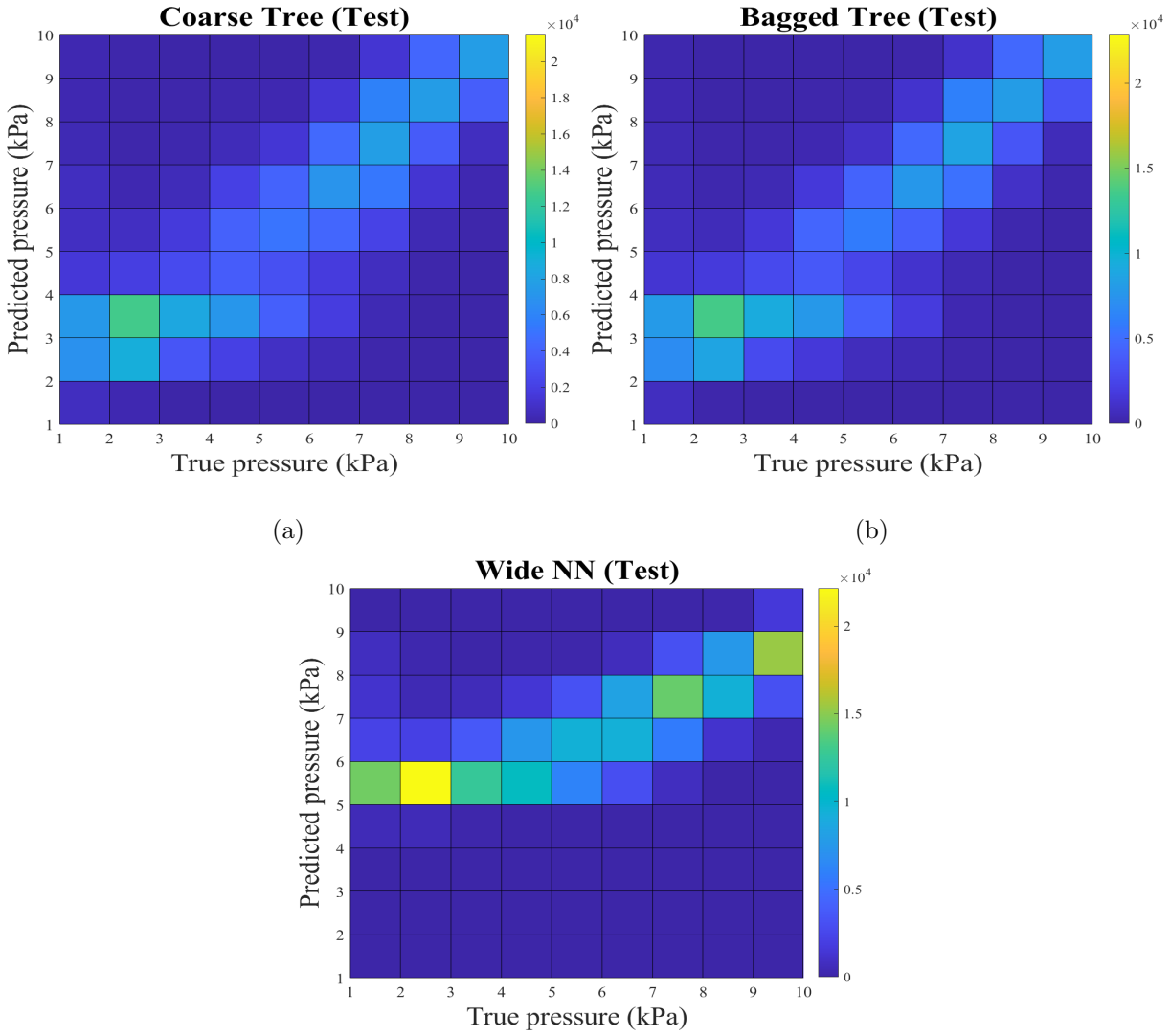


Figure 3.7: 2D histogram plots with color maps representing the density of predicted versus true pressure values categorized into ten classes for the testing dataset using various ML algorithms: (a) Coarse Tree, (b) Bagged Tree, and (c) Wide Neural Network (WNN).

Furthermore, the BET technique has the highest R-squared value (0.49) for the testing dataset.

Similar to the pressure value estimation, all of the neural networks, decision trees, linear regression, and ensemble trees can predict up to 1406412, 836348, 519682, and 121471 observations per second, respectively. The neural networks and FDT had the slowest training

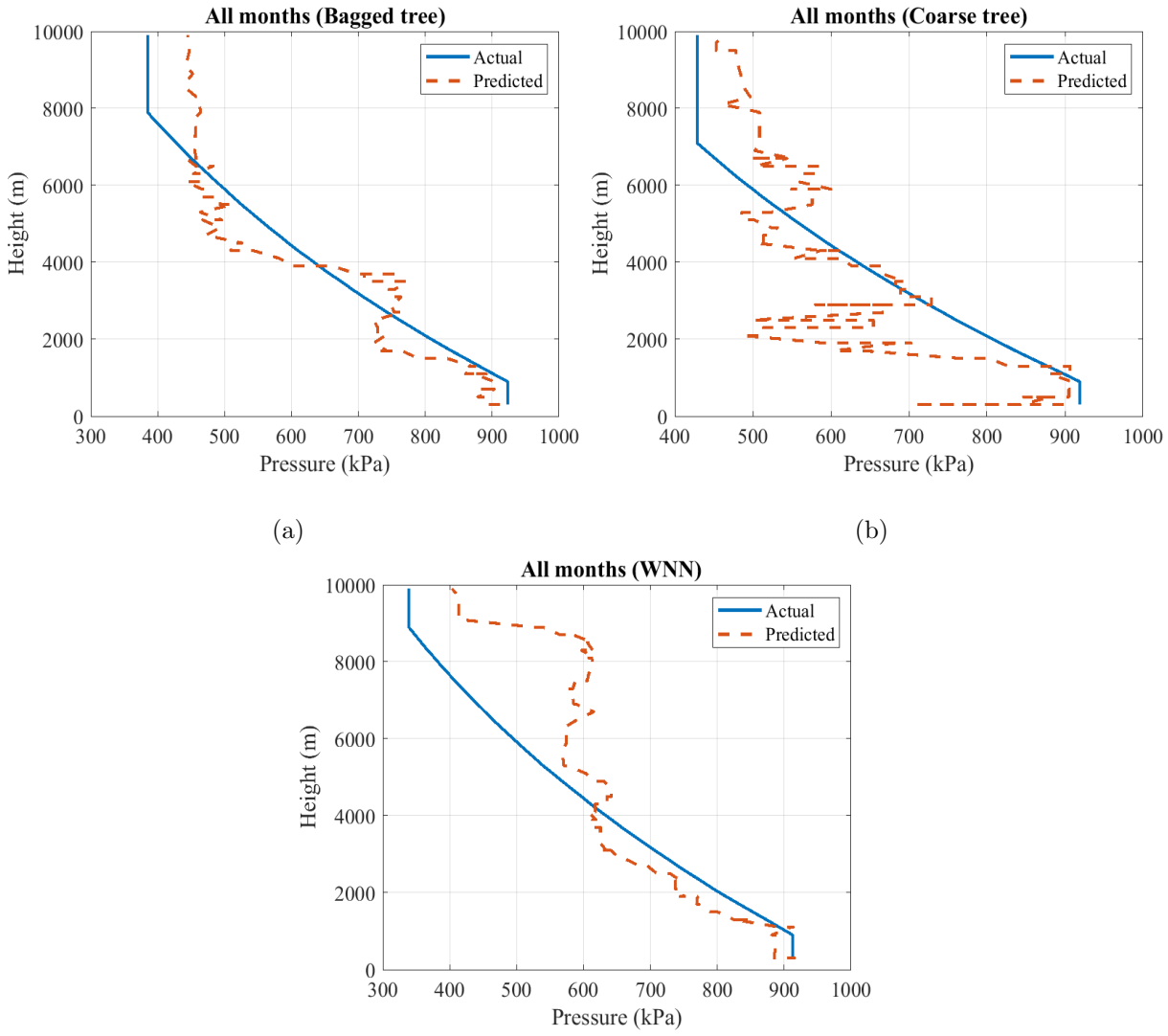


Figure 3.8: Vertical profiles of predicted and actual pressure values for the three ML algorithms. (a) Bagged tree, (b) Coarse tree, and (c) Wide Neural Network (WNN).

time compared to the other models. The linear regression model and the BoostedET are the fastest models throughout the training phase, with times of roughly 53 seconds and 803 seconds, respectively. As a result, the Bagged ensemble tree is the best machine learning model for predicting relative humidity when compared to the other proposed ML techniques.

Table 3.4: Comparison of the machine learning models' performance in forecasting humidity across six months of cross-validation data.

Model Type	RMSE (Valid) (%)	MSE (Valid) (%²)	ρ (Valid)	R^2 (Valid)	MAE (Valid) (%)	Prediction Speed (obs/sec)	Training Time (sec)
ILR	25.13	631.34	0.57	0.33	20.73	519682.08	53.05
FT	26.26	689.62	0.86	0.27	18.88	836348.75	25183.75
MT	24.75	612.49	0.76	0.35	18.76	463455.07	9708.12
CT	24.17	584.14	0.69	0.38	18.99	511306.23	878.89
BoostedET	24.92	621.14	0.59	0.34	20.69	121471.29	803.27
BET	22.2	493.02	0.83	0.48	17.44	18553.43	3137.75
NNN	24.93	621.59	0.58	0.34	20.46	934521.39	5927.35
MNN	24.68	609.08	0.59	0.35	20.16	749117.18	12012.23
WNN	24.47	598.7	0.6	0.36	19.92	887005.17	22554.27
BiNN	24.71	610.64	0.59	0.35	20.2	986172.83	12031.56
TriNN	24.7	609.85	0.6	0.35	20.18	1406412.82	15824.96

Table 3.5: A comparison of the performance of machine learning models in predicting humidity across a six-month test dataset.

Model Type	RMSE (Test) (%)	MSE (Test) (%²)	MAE (Test) (%)	ρ (Test)	R^2 (Test)
ILR	25.14	632.02	20.74	0.57	0.33
FT	25.88	670.02	18.43	0.6	0.29
MT	24.5	600.05	18.45	0.62	0.36
CT	24.04	578.03	18.81	0.62	0.39
BoostedET	24.93	621.68	20.7	0.59	0.34
BET	21.9	479.51	17.14	0.70	0.49
NNN	24.94	622	20.43	0.58	0.34
MNN	24.84	616.95	20.31	0.59	0.34
WNN	24.5	600.15	19.97	0.6	0.36
BiNN	24.77	613.76	20.27	0.59	0.35
TriNN	24.65	607.73	20.13	0.6	0.35

The partial dependency relationship (PDR) plots for the four input features (Velocity, SNR, SW, and projected pressure) vs. relative humidity are shown in Figure 3.9. The Bagged tree displays a nonlinear connection with humidity in the PDR for Velocity and SW values. When the SNR is less than -10 dB, it can be shown that there is a linear relationship between the SNR and the humidity for the BET and CT. The link between predicted pressure and relative humidity is linear in the last plot for all the models presented.

Figures 3.10 and 3.11 illustrate scatter density plots with color maps and 2D histogram plots of the predicted versus true relative humidity values using three different algorithms. All of these results are based on the six months of cross-validation data. These results show that the bagged tree model performs better than the other two models. For the testing dataset, Figures 3.12 and 3.13 show typical scatter density plots with color maps and 2D

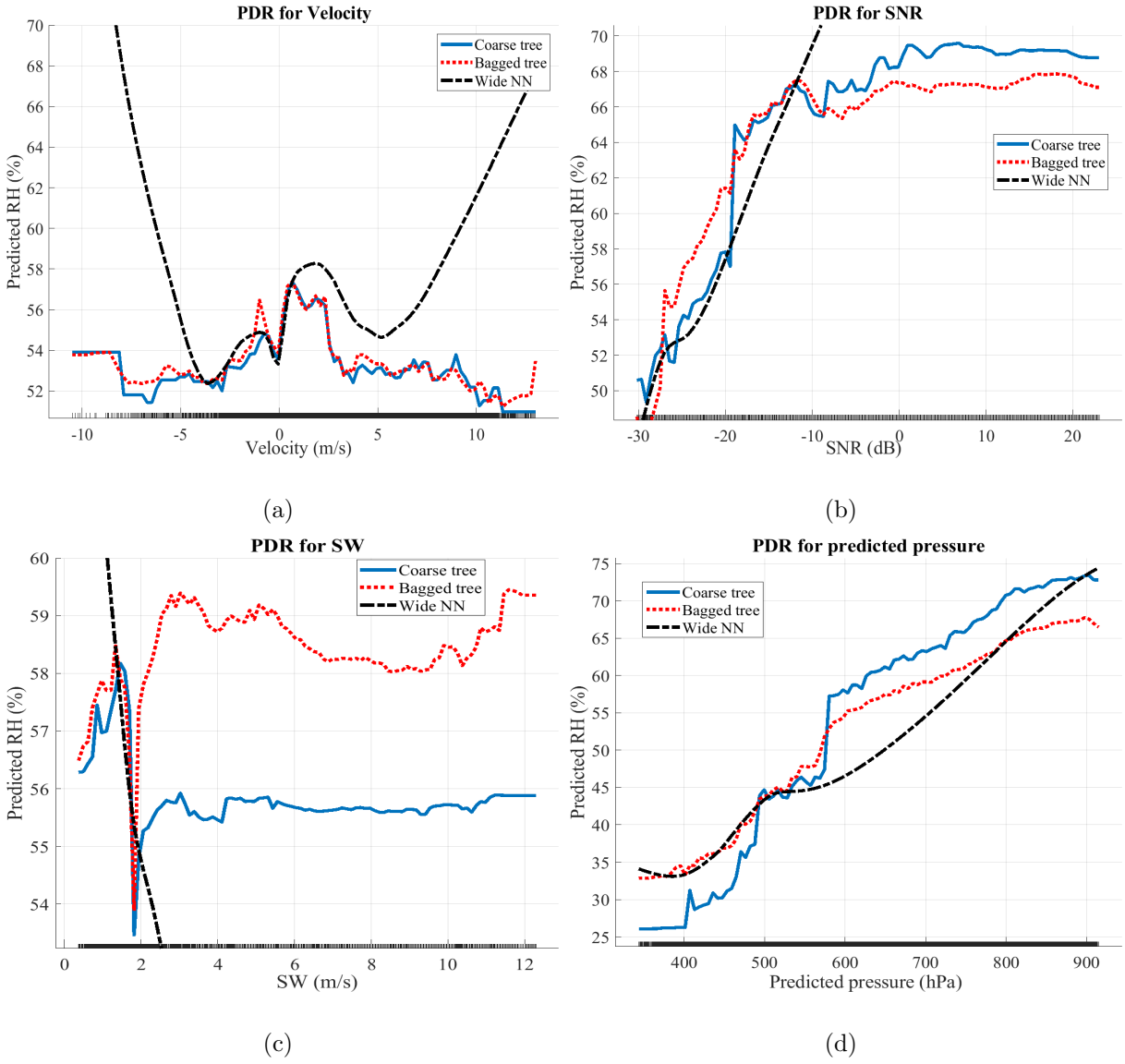
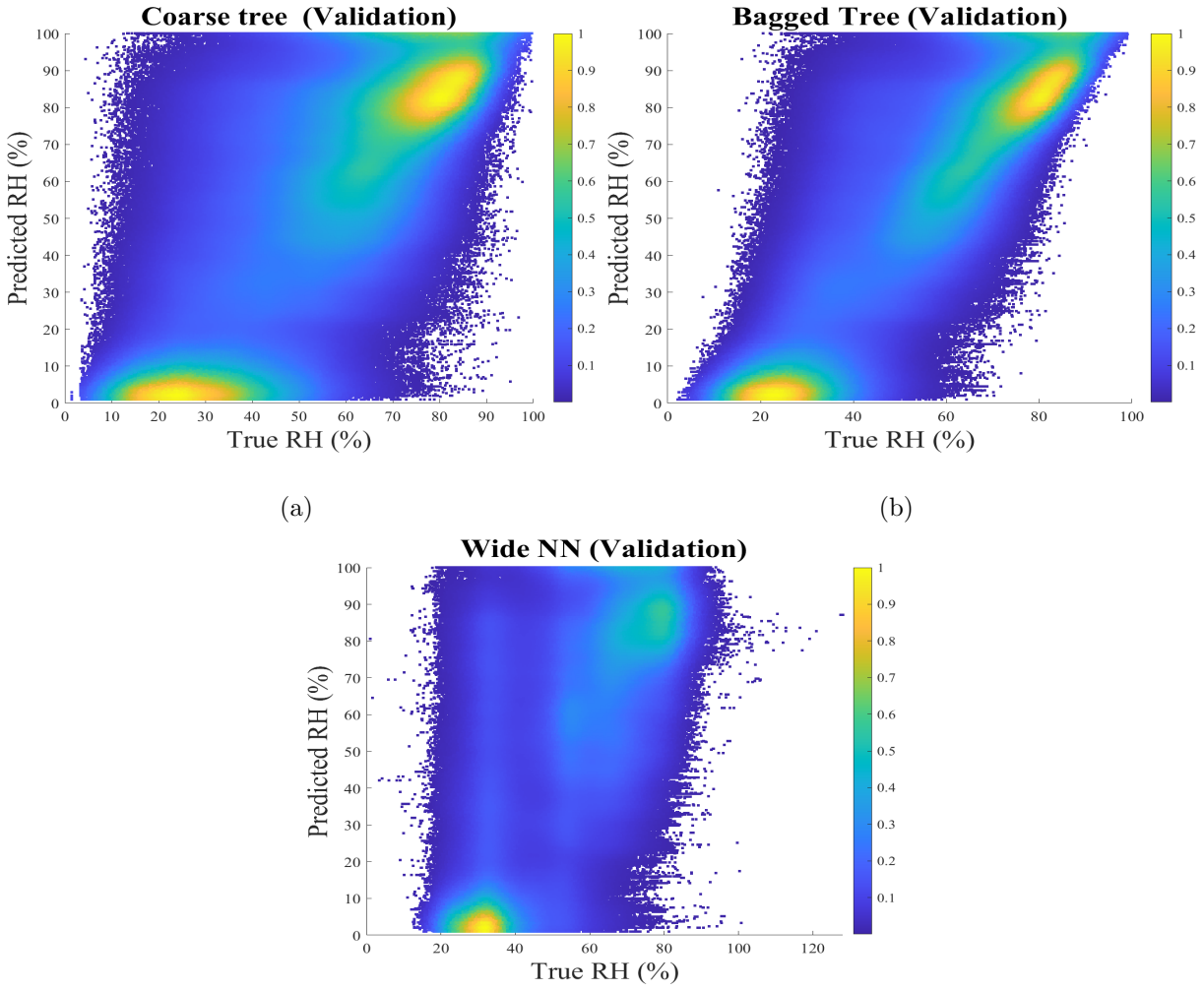


Figure 3.9: PDRs between predicted relative humidity and input variables of the whole six-month dataset. PDR for (a) velocity, (b) SNR, (c) Spectrum Width (SW), and (d) predicted pressure.

histogram plots with squares representing the predicted relative humidity versus the actual humidity values. Similarly, the bagged tree model outperforms all of the other techniques.

As shown in Figure 3.14, a humidity profile is generated progressively from predicted humidity values using the best ML model (BET) compared to the corresponding radiosonde profile. The plotted humidity profile was selected randomly from the six-month dataset



(a) (b)

(c)

Figure 3.10: Scatter density plots with a color map of predicted vs. actual humidity of the validation dataset (all months) using several ML algorithms; (a) coarse tree, (b) bagged tree, and (c) Wide Neural Networks (WNN).

to demonstrate the BET method's ability to estimate humidity at different periods. The graph depicts the BET technique prediction of relative humidity at various heights ranging from 0 to 10 km. The BET could estimate humidity at various heights with good accuracy, particularly below 5 km.

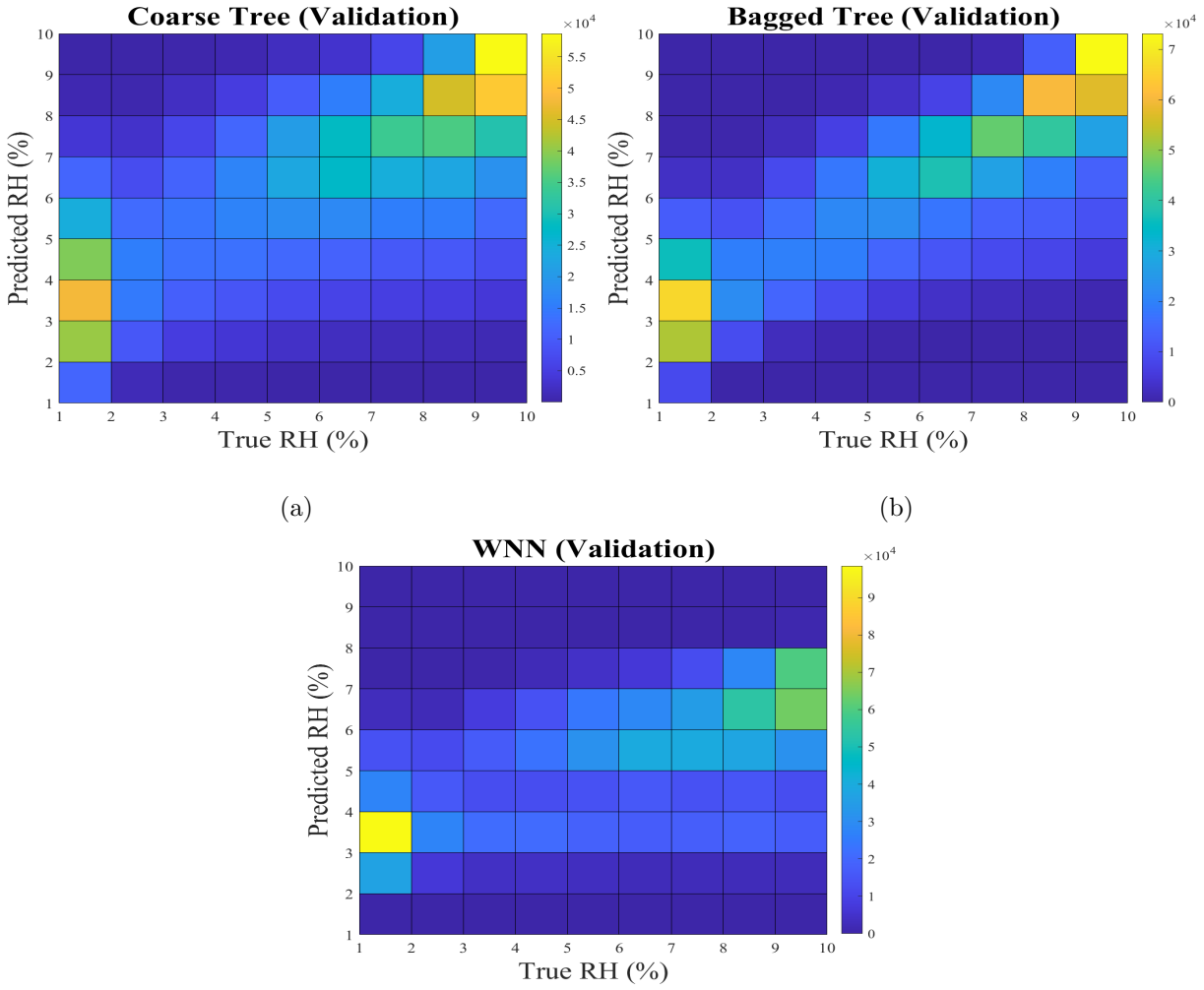


Figure 3.11: Different scatter density plots with color maps of predicted vs. true humidity of the (all months) validation dataset utilizing the following ML algorithms; (a) coarse tree, (b) Bagged tree, and (c) Wide Neural Networks (WNN).

3.4.1.3 Computational Efficiency

The suggested machine learning algorithms' computational efficiency is assessed in terms of training time and memory usage. Tables 3.2 and 3.4 show a detailed comparison of several models in predicting atmospheric pressure and humidity with six months of cross-validation data. The Interactions Linear Regression (ILR) model was shown to be the most time-efficient in the context of pressure prediction, taking just 37.51 seconds for training. The

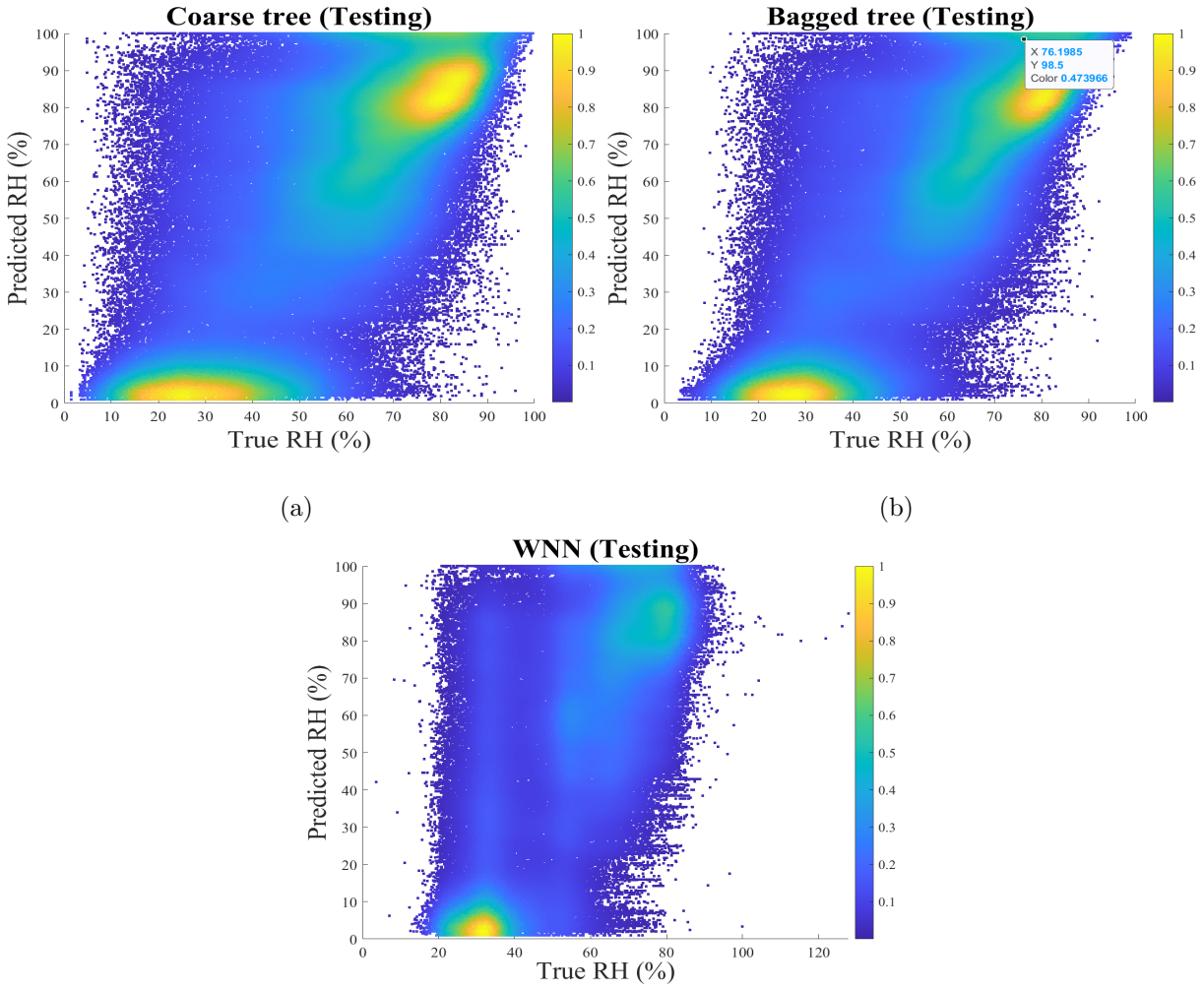
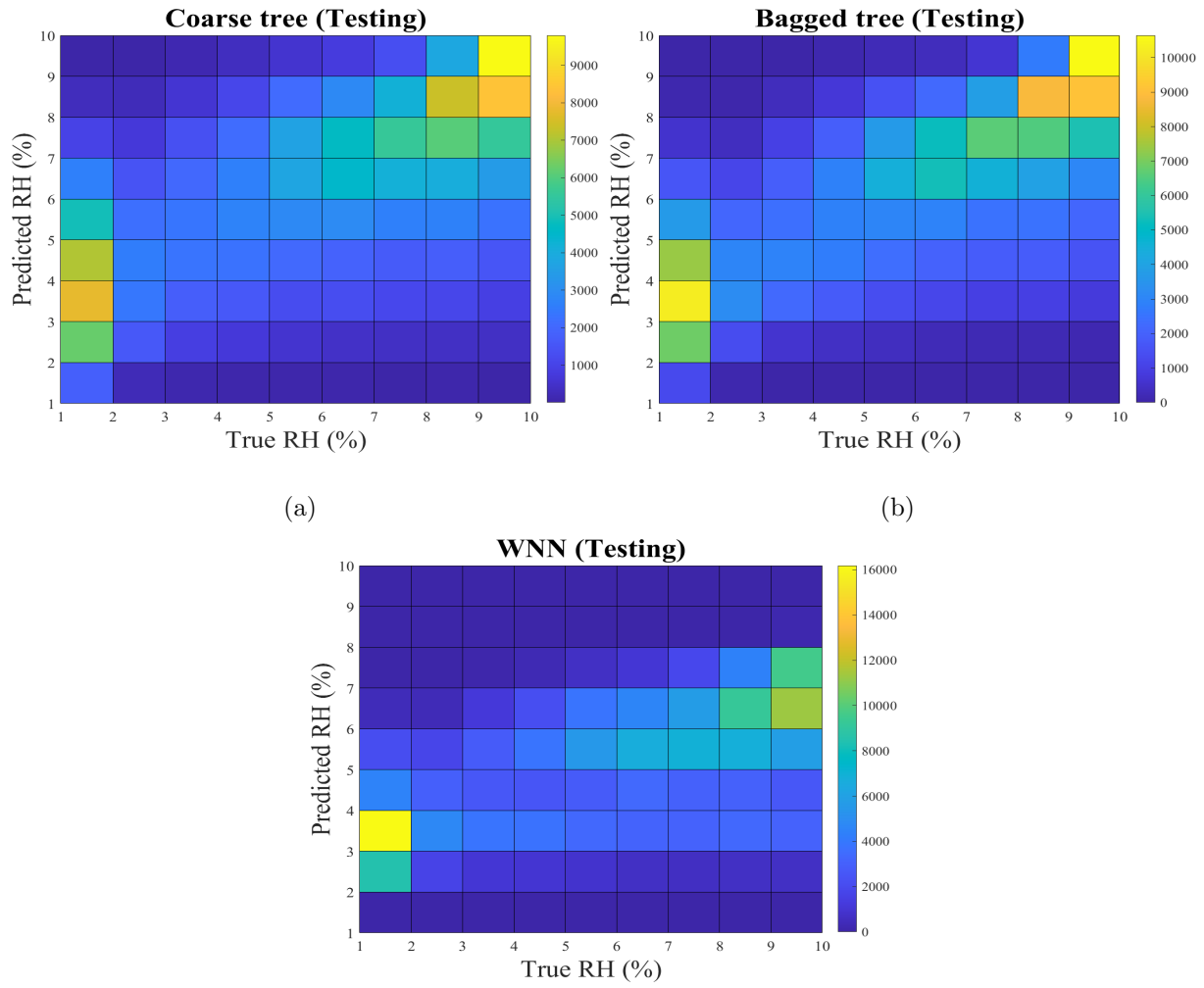


Figure 3.12: Scatter density graphs with a color map of predicted vs. true humidity of the (all months) test dataset created by the following machine learning algorithms: (a) coarse tree, (b) Bagged tree, and (c) Wide Neural Networks (WNN).

Wide Neural Network (WNN) model, on the other hand, required the most training time, clocking in at 18863.05 seconds. In addition, ensemble methods such as the Boosted Ensemble Tree and Bagged Ensemble Tree models showcased average training times. Similar patterns were seen during humidity prediction, when the ILR model displayed efficiency in training time, in contrast to the Wide Neural Network model's long training period. As



(a) (b)

(c)

Figure 3.13: Different scatter density graphs with color maps of expected vs. actual humidity of the (all months) test dataset using the ML algorithms; (a) coarse tree, (b) Bagged tree, and (c) Wide Neural Networks (WNN).

shown in Table 3.6, the lowest, maximum, and mean memory consumption values are summarized using the whole six-month dataset, providing an understanding of these methods' computational needs and resource use. The Coarse Tree approach used the least memory for pressure prediction, with a mean memory use of 426.6462 Mbytes. In contrast, the WNN algorithm consumed the most memory, with a mean of 2.2607e+03 Mbytes. Humidity prediction followed a similar trend, with the Coarse Tree approach using the least amount of

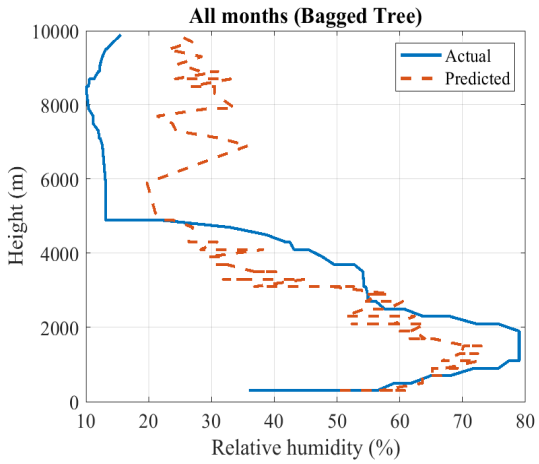


Figure 3.14: The Bagged tree algorithm's vertical profile of predicted and true relative humidity values.

Table 3.6: Comparison of the most significant machine learning models' performance in terms of memory consumption.

	Min memory consumption (Mbytes)	Max memory consumption (Mbytes)	Mean memory consumption (Mbytes)
Coarse tree (6 months, pressure prediction)	90	595	426.6462
Bagged tree (6 months, pressure prediction)	140	2139	1.0172e+03
WNN (6 months, pressure prediction)	666	3301	2.2607e+03
Coarse tree (6 months, humidity prediction)	469	809	668.9355
Bagged tree (6 months, humidity prediction)	351	2412	1.4913e+03
WNN (6 months, humidity prediction)	416	3086	2.1123e+03

memory (668.9355 Mbytes) while the WNN algorithm used the most (2.1123e+03 Mbytes). These memory usage patterns indicate significant differences in processing needs across different machine learning algorithms and prediction workloads. These findings have practical significance, especially in resource-constrained circumstances where memory restrictions may be challenging.

3.4.2 Month-By-Month Evaluations

3.4.2.1 Pressure Estimation

During various months and different seasons, there can be significant changes in pressure and relative humidity. As a result, estimation performance of each individual month is further examined. To conduct this analysis, the dataset for each month includes all the Doppler spectral moments and radiosonde data collected during that month are used. The total samples collected in January, February, March, April, May, and June are 158766, 185791, 194460, 272808, 286727, and 308140, respectively.

All the datasets for different months were trained using the same machine-learning models. However, for the sake of brevity, only the best-performing ML algorithms in the decision tree, ensemble tree, and ANN are listed. Tables 3.7 and 3.8 include detailed data on each month's effectiveness of the top three machine learning algorithms. In addition, these tables include information on all performance evaluation methodologies, including root mean squared error (RMSE) (kPa), mean squared error (MSE) (kPa^2), correlation coefficient (dimensionless), mean absolute error (MAE) (kPa), prediction speed (observations/second), and training time (seconds).

According to both tables, the BET technique outperformed the WNN and CT methods in terms of RMSEs, MSEs, and MAE in all months, with a substantial difference. This demonstrates that BET has the best estimation accuracy for pressure. Furthermore, based on cross-validation dataset analysis, the BET technique achieved the following correlation coefficient values: 0.93, 0.94, 0.95, 0.95, 0.93, and 0.9 for January, February, March, April, May, and June, respectively. Based on testing dataset analysis, the following correlation values were obtained: 0.93, 0.94, 0.95, 0.95, 0.93, and 0.91 for January to June.

In addition, the BET model consistently outperformed the other models regarding R-squared values for both cross-validation and testing datasets across all months. For example, the BET in the March dataset obtained the highest R-squared value of 0.88, indicating that

the BET model accounts for 88% of the variability in the pressure variable. These findings demonstrate that the BET technique is more reliable and stable than the other methods in terms of having minimal errors in pressure predictions across all datasets. On average, the wide neural networks (WNNs) can predict the highest number of observations per second, while the bagged ensemble trees can observe the least observations per second. However, regarding the training speed, the WNN takes the most time, followed by the BET and finally by the CT. To summarize, it seems clear that the Bagged ensemble tree is the best machine learning model for effectively predicting pressure.

Table 3.7: Comparison of the performance of ML models in predicting pressure (each month) using cross-validation data.

Model	RMSE (Valid) (kPa)	MSE (Valid) (kPa^2)	ρ (Valid)	R^2 (Valid)	MAE (Valid) (kPa)	Pred. Speed (obs/sec)	Train Time (sec)
CT (Jan)	84.26	7099.94	0.91	0.82	58.49	535609.04	11.28
BET (Jan)	80.33	6452.44	0.93	0.84	55.53	22469.62	161.90
WNN (Jan)	85.54	7317.74	0.90	0.81	60.80	1006409.75	1930.91
CT (Feb)	78.89	6222.91	0.92	0.82	54.84	850895.20	22.05
BET (Feb)	73.44	5393.84	0.94	0.85	51.05	32731.09	175.29
WNN (Feb)	79.86	6377.20	0.91	0.82	57.79	1010450.46	1965.07
CT (March)	70.98	5037.66	0.94	0.86	49.95	1291171.51	18.34
BET (March)	66.40	4409.26	0.95	0.88	46.77	43060.84	157.34
WNN (March)	71.38	5095.00	0.93	0.86	51.38	1081371.59	2161.29
CT (April)	71.91	5171.15	0.93	0.85	51.25	414760.43	25.77
BET (April)	67.18	4513.79	0.95	0.87	47.70	18572.45	332.55
WNN (April)	72.47	5252.10	0.92	0.85	52.43	1051270.32	3352.45
CT (May)	83.71	7007.81	0.90	0.80	57.49	1403378.45	32.53
BET (May)	76.77	5894.36	0.93	0.83	52.82	45653.43	261.50
WNN (May)	85.83	7367.31	0.89	0.79	61.09	1069922.17	3299.32
CT (June)	104.02	10821.15	0.85	0.68	72.15	435601.58	34.81
BET (June)	94.54	8938.68	0.90	0.74	65.77	20260.96	378.90
WNN (June)	107.69	11597.88	0.82	0.66	78.60	1063646.44	3864.20

Table 3.8: Comparison of ML algorithms in predicting pressure in the test dataset for each month.

Model	RMSE (Test) (kPa)	MSE (Test) (kPa^2)	MAE (Test) (kPa)	ρ (Test)	R^2 (Test)
CT (Jan)	83.41	6956.67	58.08	0.91	0.82
BET (Jan)	78.97	6236.99	54.80	0.93	0.84
WNN (Jan)	84.56	7150.80	60.13	0.90	0.82
CT (Feb)	77.65	6030.15	54.03	0.92	0.83
BET (Feb)	71.85	5162.92	49.92	0.94	0.85
WNN (Feb)	79.38	6301.04	57.35	0.91	0.82
CT (March)	70.89	5025.62	49.74	0.94	0.86
BET (March)	65.82	4332.83	46.28	0.95	0.88
WNN (March)	72.09	5197.21	51.51	0.93	0.86
CT (April)	73.03	5333.27	51.48	0.93	0.85
BET (April)	67.38	4540.44	47.38	0.95	0.87
WNN (April)	73.91	5463.33	52.88	0.92	0.85
CT (May)	82.88	6869.09	56.80	0.90	0.80
BET (May)	75.31	5671.30	51.69	0.93	0.84
WNN (May)	86.11	7414.59	61.26	0.88	0.79
CT (June)	102.65	10536.88	71.14	0.85	0.69
BET (June)	92.80	8611.90	64.40	0.91	0.75
WNN (June)	106.89	11425.78	78.09	0.82	0.66

Figure 3.15 displays multiple pressure profiles created from predicted pressure values in all six months using the best ML model (BET) compared to the related radiosonde profile. The BET model has high accuracy in predicting pressure at various heights, as seen in the monthly plots. Also, the plots show good agreement between the predicted and truth

pressure values, indicating that the BET model effectively captures the underlying patterns and trends in the data.

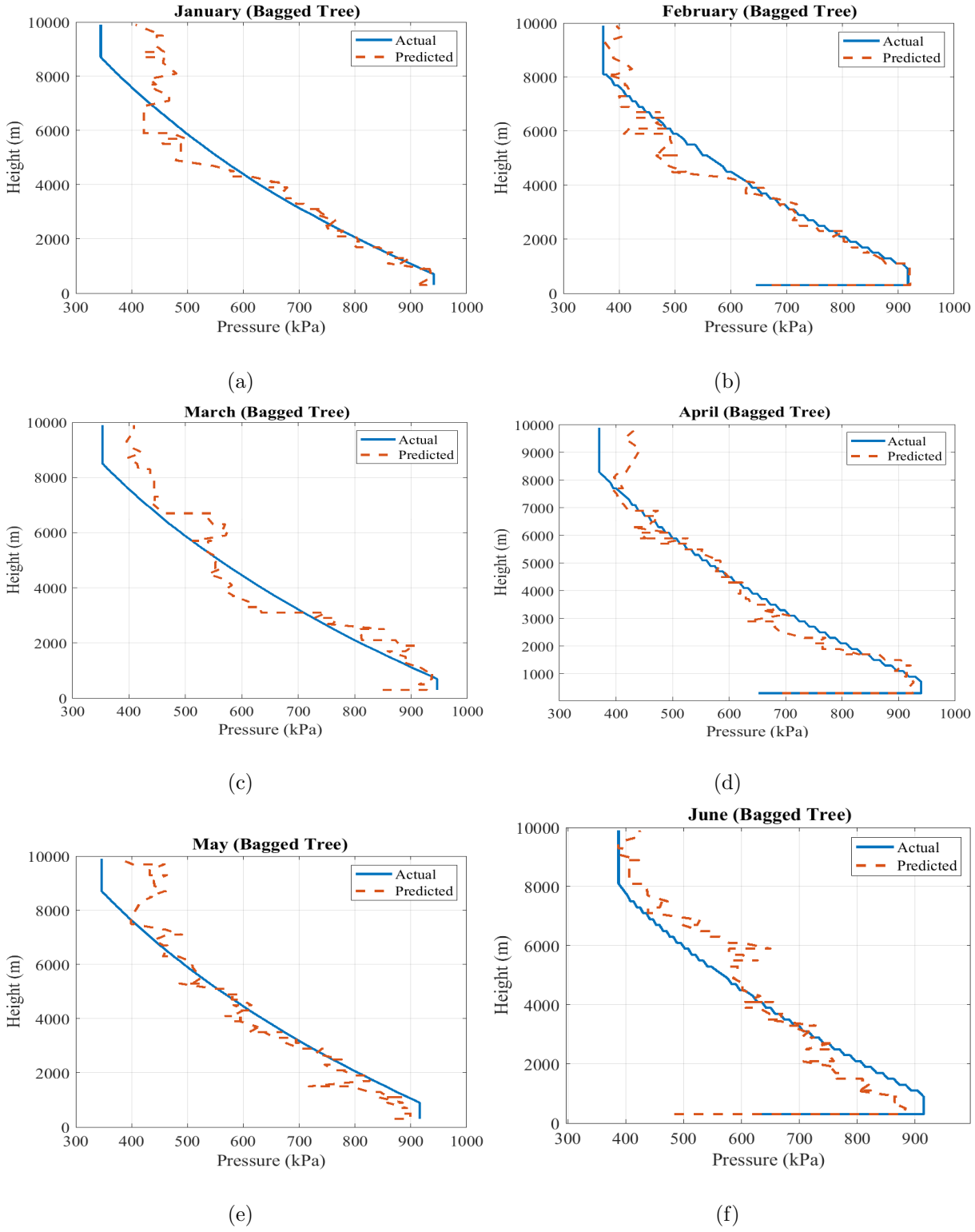


Figure 3.15: Vertical pressure profiles of the Bagged tree during the following months: (a) January, (b) February, (c) March, (d) April, (e) May, and (f) June.

Figure 3.16 depicts the partial dependency relationship (PDR) plots for the four input features (Pr, Velocity, SNR, and SW) versus predicted pressure. Each subfigure shows a comparison between the different six months for each input feature versus predicted pressure using the BET model. The BET model behaves differently depending on the predicted pressure readings across different months. It shows a generally linear relationship with predicted pressure for Pr and SNR values, although this relationship is not perfectly linear and exhibits some complexity. When the velocity is greater than zero, an inverse relationship with predicted pressure is observed, including complex variations. Similarly, the SW feature shows an inverse relationship with predicted pressure when SW is less than two, with additional nuances in the data.

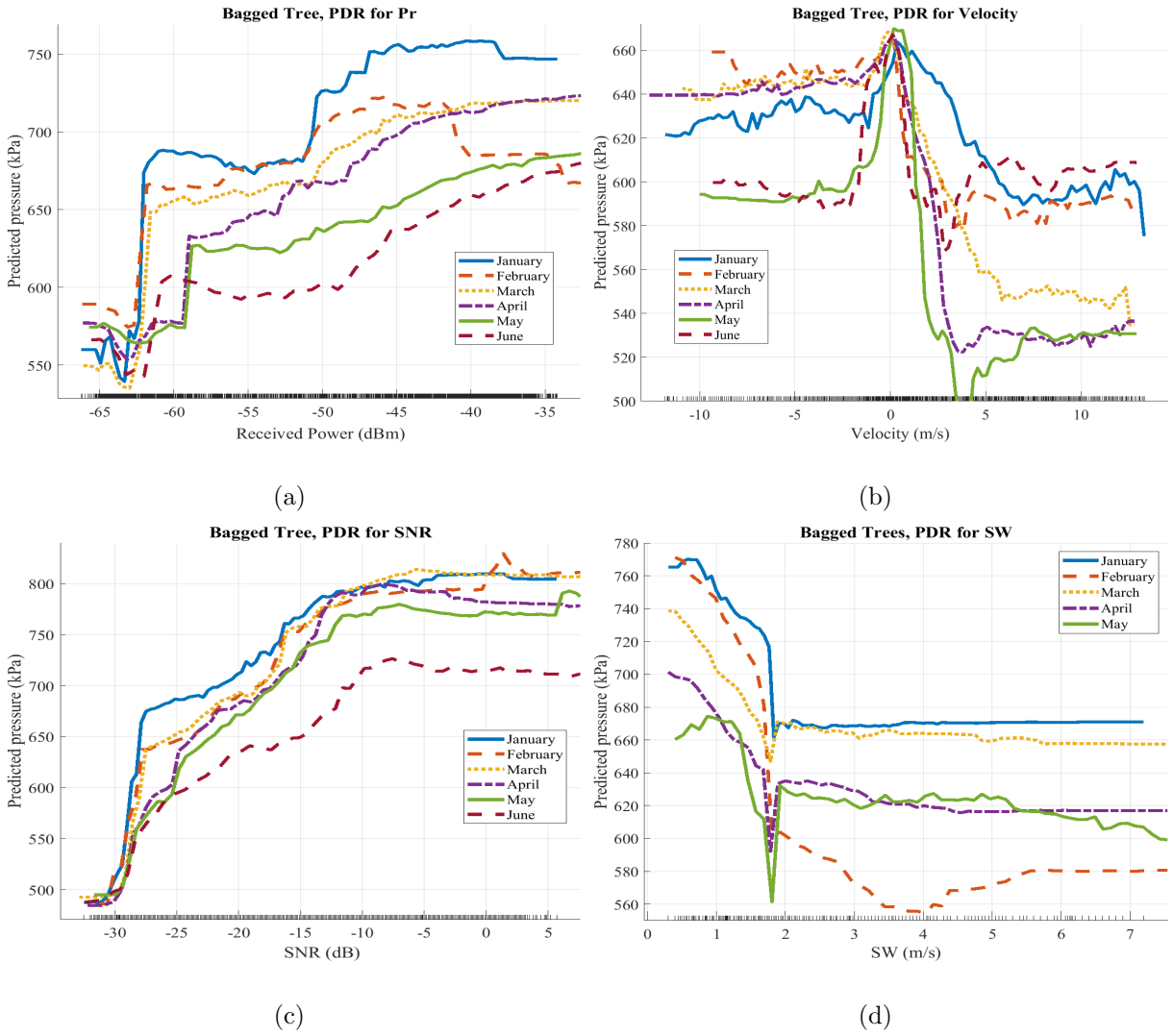


Figure 3.16: Partial Dependence Relationship (PDR) plots between predicted pressure and input features (for each month individually). PDR for (a) Received power (P_r) and predicted pressure, (b) velocity and predicted pressure, (c) SNR and predicted pressure, and (d) Spectrum Width (SW) and predicted pressure.

Figures 3.17 and 3.18 show scatter density plots with color maps and 2D histograms of predicted pressure using the BET model versus the true pressure data, respectively. These results are based on cross-validation data for each month. The bagged tree model works well since most scatter points cluster around the center. Figures 3.19 and 3.20 for the testing

dataset show typical scatter density plots with color maps, and 2D histograms with squares of the BET's predicted pressure vs. true pressure data.

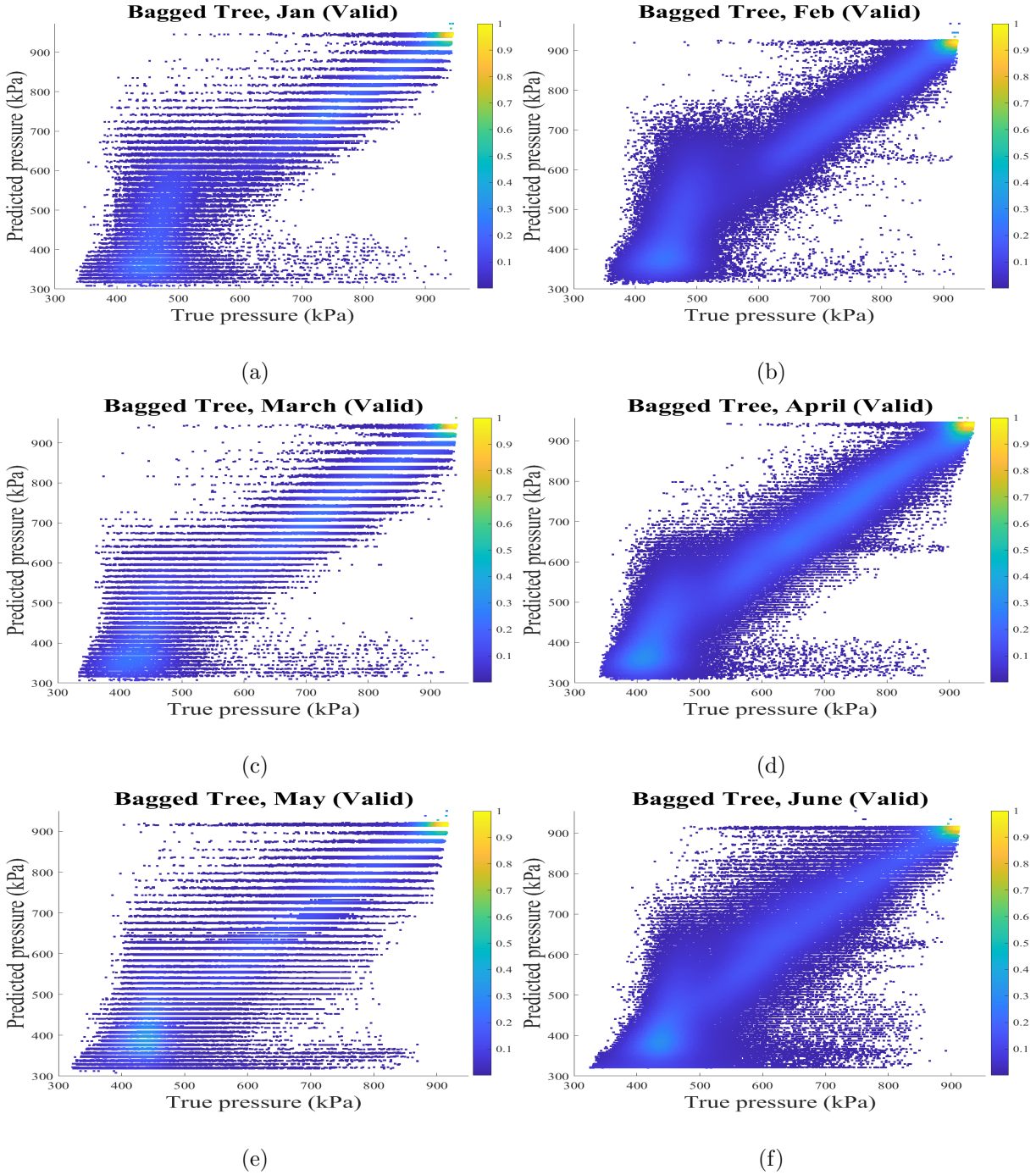
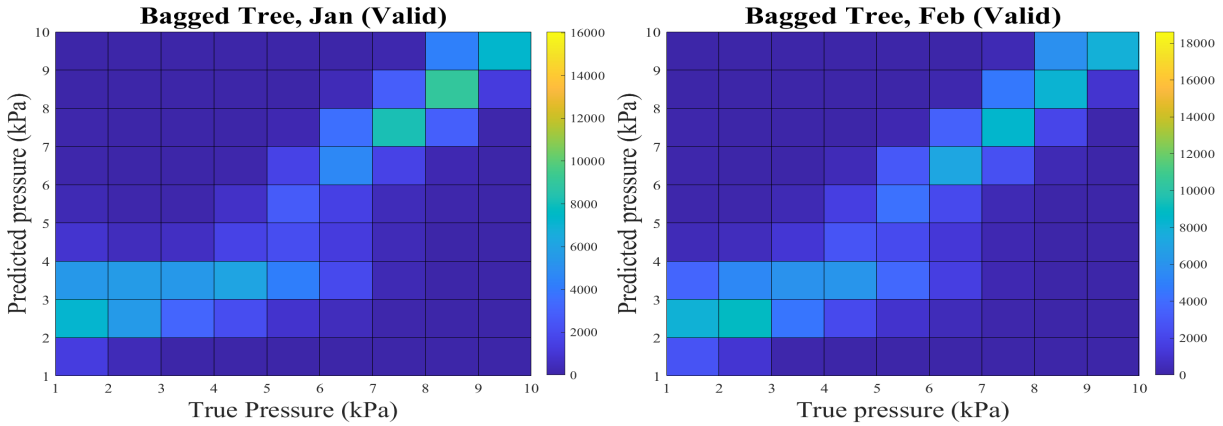
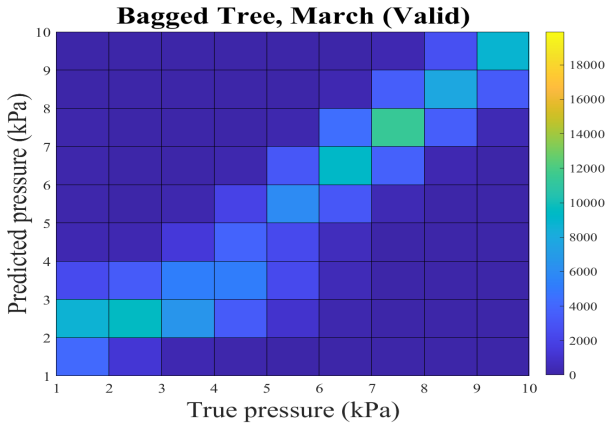


Figure 3.17: Scatter density graphs of predicted vs. true pressure values of the cross-validation dataset using the bagged tree model for the following months: (a) January, (b) February, (c) March, (d) April, (e) May, and (f) June.

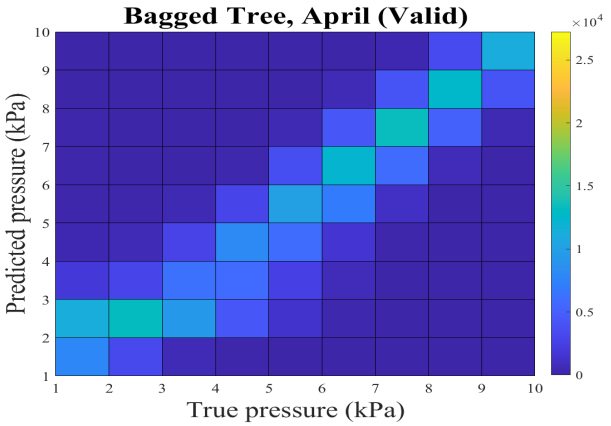


(a)

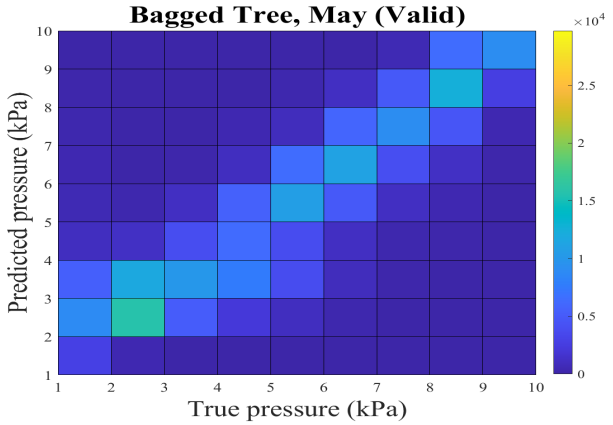
(b)



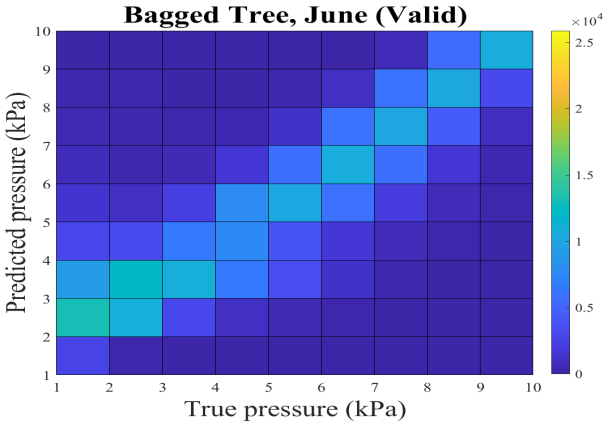
(c)



(d)



(e)



(f)

Figure 3.18: 2D histogram graphs of predicted vs. true pressure values from the cross-validation dataset using the bagged tree model for the following months: (a) January, (b) February, (c) March, (d) April, (e) May, and (f) June.

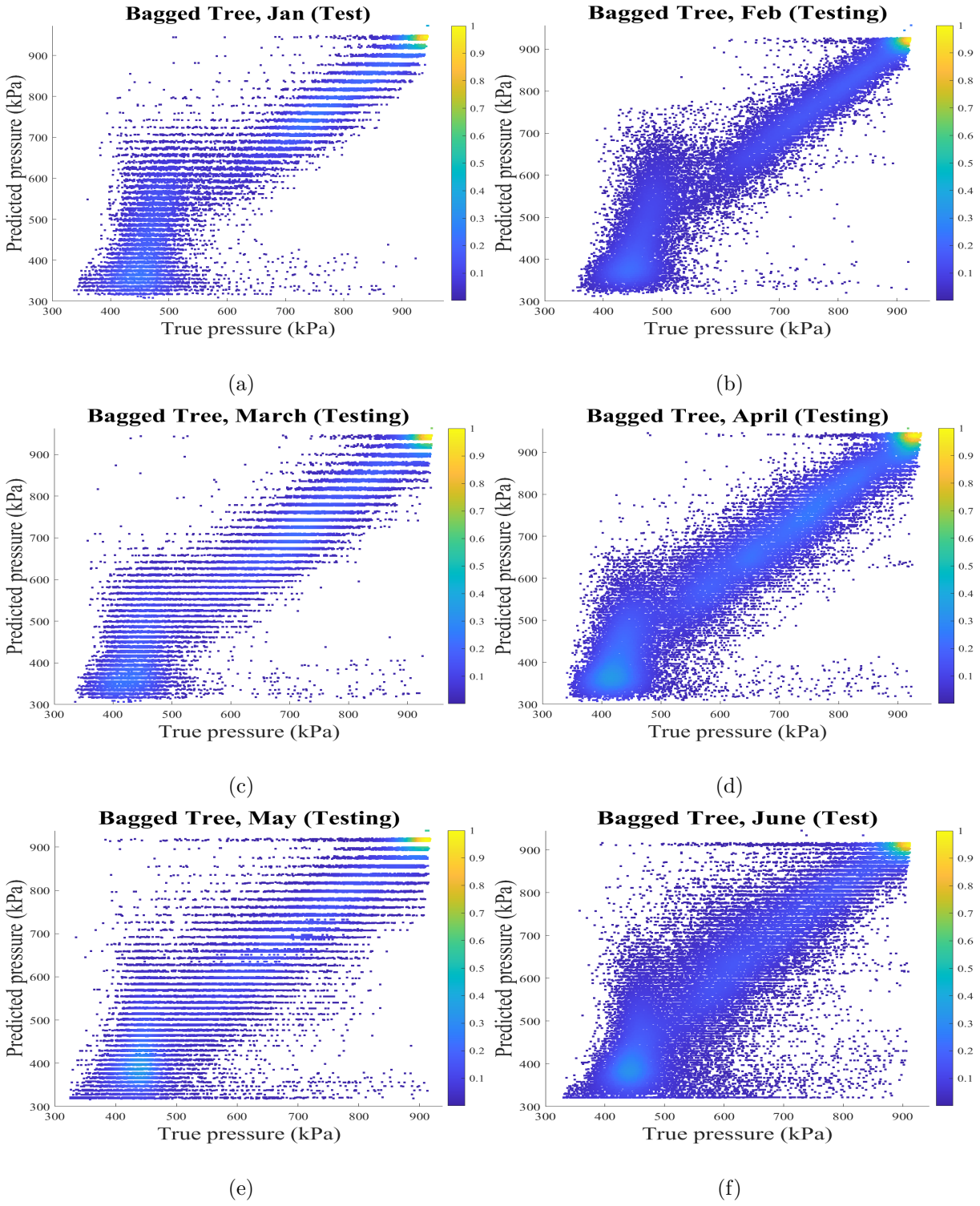
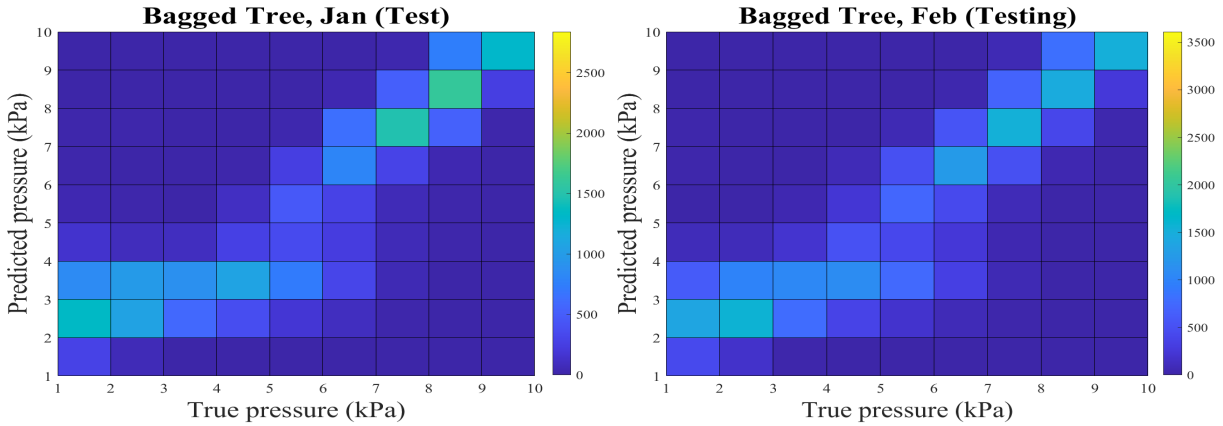
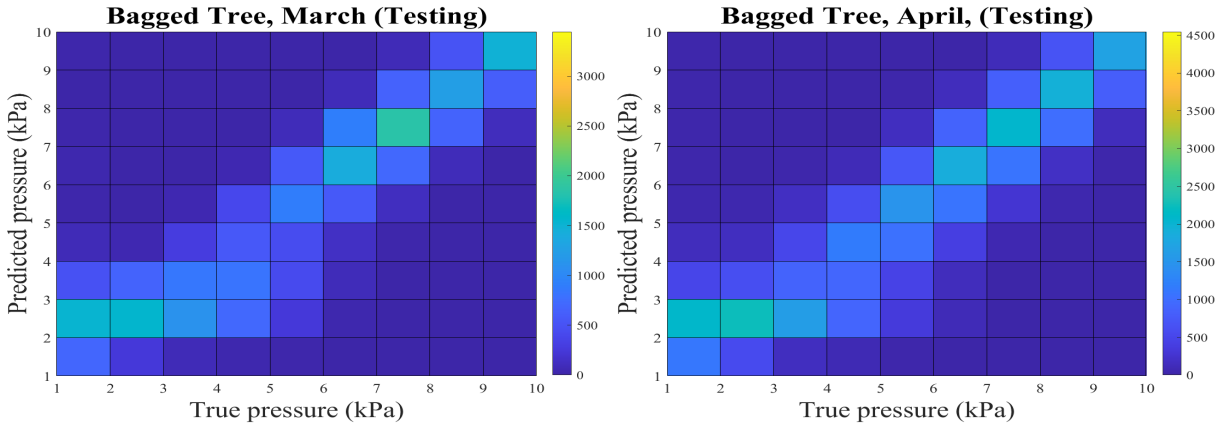


Figure 3.19: Scatter density graphs showing the test dataset's predicted vs. true pressure values using the Bagged Tree model for the following months: (a) January, (b) February, (c) March, (d) April, (e) May, and (f) June.



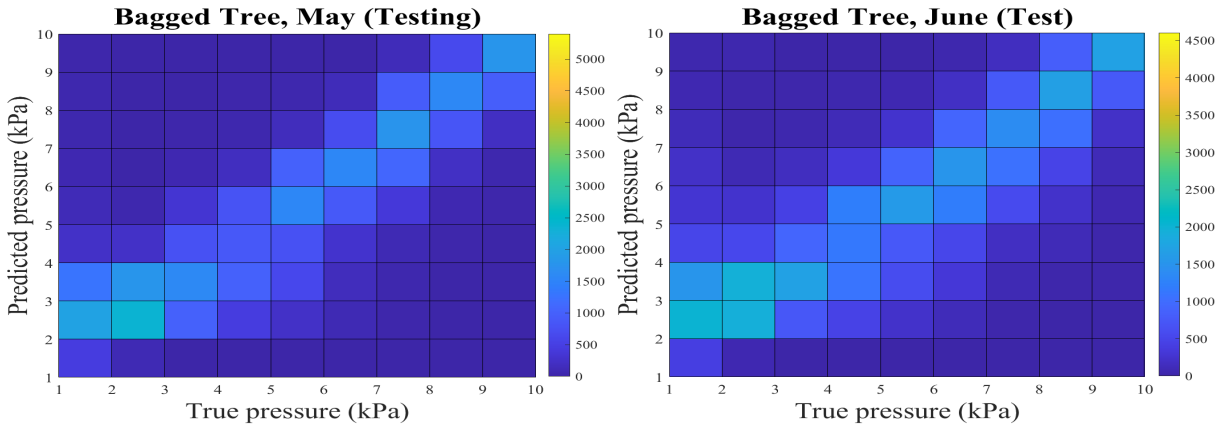
(a)

(b)



(c)

(d)



(e)

(f)

Figure 3.20: 2D histogram plots of predicted vs. true pressure values from the test dataset using the bagged tree model for the following months: (a) January, (b) February, (c) March, (d) April, (e) May, and (f) June.

3.4.2.2 Humidity Estimation

Each month's dataset, including the Doppler spectral moments and predicted pressure, was trained using the same machine learning algorithms as in the previous section to predict relative humidity. The results of the Decision Tree, Ensemble Tree, and ANN models were then compared. Tables 3.9 and 3.10, which exhibit the cross-validation and testing datasets, include detailed data on each month's results of the three machine learning algorithms. In addition, these tables include results for all the performance metrics: root mean squared error (RMSE) (%), mean squared error (MSE) ($\%^2$), correlation coefficient, mean absolute error (MAE) (%), prediction speed (observations/second), and training time (seconds). According to both tables, the BET model surpasses the WNN and CT models in all months in terms of RMSEs, MSEs, and MAE. The BET produced good correlation coefficient values based on cross-validation dataset analysis, ranging from 0.83 to 0.94 for May and June. The corresponding values for the testing dataset analysis ranged from 0.71 to 0.91 for May and June, respectively. Further, the BET had the best R-squared values for cross-validation and testing datasets across all months. For example, the highest R-squared value for the BET was 0.78 in the March dataset. Overall, the Bagged Ensemble Tree is the most accurate machine learning model for forecasting relative humidity based on the results from each individual month.

Table 3.9: Comparison of ML algorithms for predicting relative humidity using cross-validation dataset (by month).

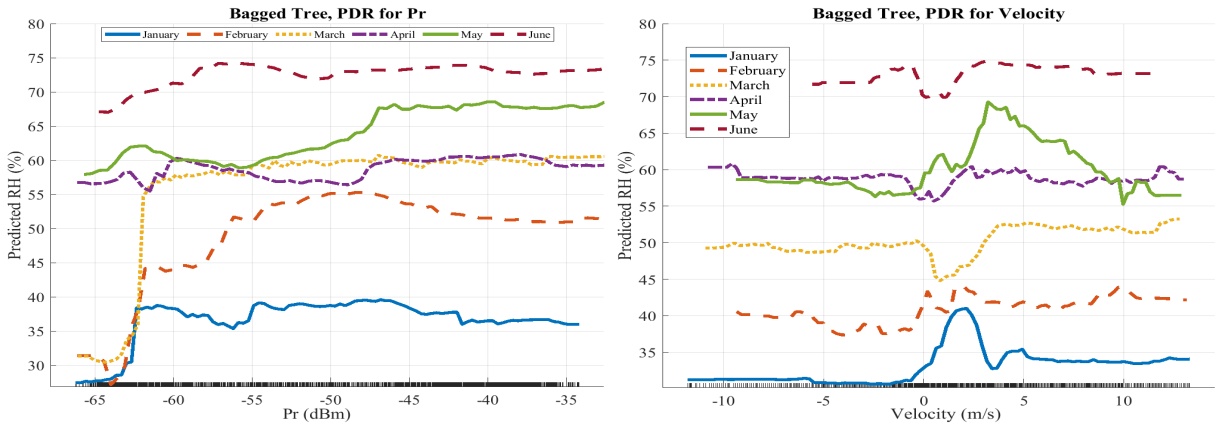
Model	RMSE (Valid) (%)	MSE (Valid) (%²)	ρ (Valid)	R^2 (Valid)	MAE (Valid) (%)	Pred. Speed (obs/sec)	Train Time (sec)
CT (Jan)	19.56	382.81	0.83	0.6483	13.14	1027135.92	23.38
BET (Jan)	17.52	306.97	0.91	0.7179	11.81	43860.84	146.46
WNN (Jan)	19.58	383.73	0.80	0.6475	13.66	1005029.16	1708.79
CT (Feb)	19.29	372.43	0.80	0.5713	14.06	1146322.46	19.66
BET (Feb)	17.33	300.55	0.89	0.6540	12.59	31802.08	212.85
WNN (Feb)	19.53	381.61	0.74	0.5607	14.83	829839.38	2221.16
CT (March)	19.58	383.51	0.86	0.6998	13.89	1172163.53	21.25
BET (March)	17.86	319.23	0.92	0.7501	12.70	41404.25	200.97
WNN (March)	19.56	382.92	0.83	0.7003	14.20	1049074.34	2155.52
CT (April)	22.69	514.93	0.74	0.4769	17.60	1382439.82	28.19
BET (April)	20.77	431.45	0.86	0.5617	16.16	39426.27	325.61
WNN (April)	22.67	514.33	0.69	0.4775	18.16	1112739.98	3041.01
CT (May)	18.70	349.71	0.66	0.3290	14.41	1112555.65	30.83
BET (May)	16.53	273.32	0.83	0.4756	12.68	36747.93	393.39
WNN (May)	19.19	368.52	0.54	0.2929	15.42	1143452.58	3183.13
CT (June)	12.49	156.00	0.83	0.6310	8.07	1321420.39	35.26
BET (June)	9.68	93.83	0.940	0.7780	6.31	41432.77	389.53
WNN (June)	15.74	248.00	0.64	0.4133	11.74	1169097.01	3393.19

Table 3.10: Comparison of ML algorithms' predictions of relative humidity in the test dataset (by month).

Model	MAE (Test) (%)	MSE (Test) ($\%^2$)	RMSE (Test) (%)	Correlation Coefficient (Test)	R^2 (Test)
CT (Jan)	13.04	375.62	19.38	0.81	0.65
BET (Jan)	11.52	293.05	17.12	0.86	0.73
WNN (Jan)	13.79	384.59	19.61	0.80	0.65
CT (Feb)	13.80	365.06	19.11	0.76	0.57
BET (Feb)	12.20	286.98	16.94	0.82	0.67
WNN (Feb)	14.83	385.61	19.64	0.74	0.55
CT (March)	13.95	386.69	19.66	0.84	0.70
BET (March)	12.51	314.26	17.73	0.87	0.75
WNN (March)	14.37	388.55	19.71	0.83	0.70
CT (April)	17.54	511.72	22.62	0.69	0.47
BET (April)	15.86	419.43	20.48	0.76	0.57
WNN (April)	18.20	515.54	22.71	0.69	0.47
CT (May)	14.17	341.44	18.48	0.60	0.35
BET (May)	12.34	262.10	16.19	0.71	0.50
WNN (May)	15.52	372.39	19.30	0.54	0.29
CT (June)	7.72	146.06	12.09	0.81	0.66
BET (June)	5.91	84.07	9.17	0.90	0.80
WNN (June)	11.77	247.82	15.74	0.64	0.42

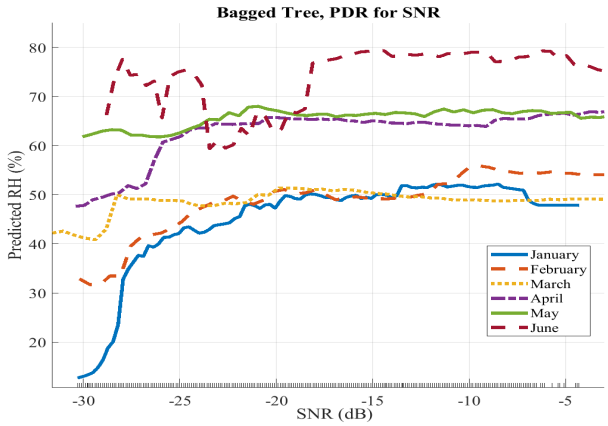
The partial dependency Relationship (PDR) plots for the five input variables (P_r, Velocity, SNR, SW, projected pressure) against relative humidity are shown in Figure 3.21. Each feature shows a meaningful correlation with relative humidity, but the strength and nature of these relationships vary across different months. Figures 3.22 and 3.23 demonstrate scatter density plots and 2D histograms of the estimated humidity using the BET model versus the true humidity values. The results of these two figures are based on cross-validation datasets collected separately for each month. The scatter density plots and the 2D histograms with colormaps of the BET's predicted humidity versus actual humidity data

for the testing dataset are shown in Figures 3.24 and 3.25. These figures show that the bagged tree model works well since most of the scatter points cluster around the center line. Figure 3.26 shows different humidity profiles generated from predicted humidity values over all six months using the BET method compared to the corresponding radiosonde profile. The BET method accurately estimates humidity at various altitudes for all the six months.

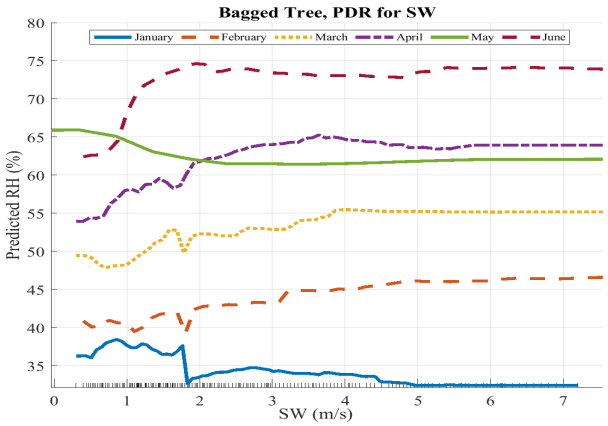


(a)

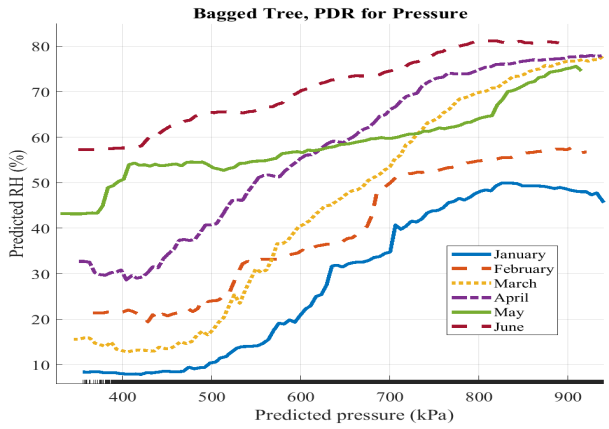
(b)



(c)

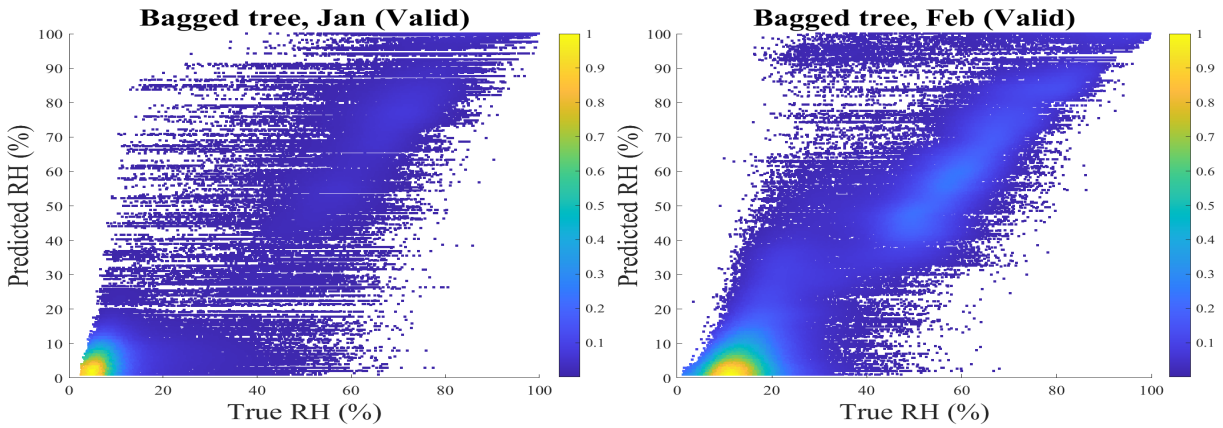


(d)



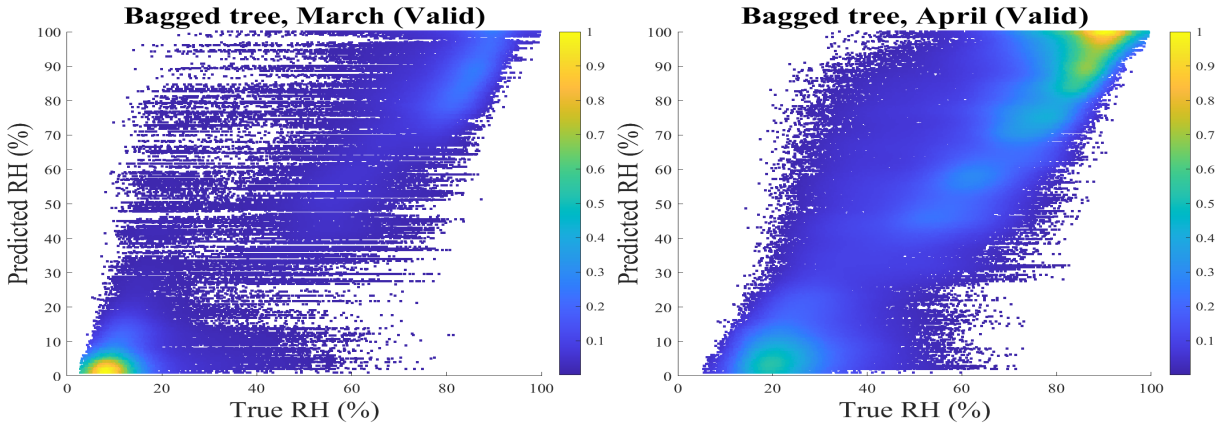
(e)

Figure 3.21: PDRs for predicted relative humidity and input variables (for each month individually). PDR for (a) P_r and predicted RH, (b) velocity and predicted RH, (c) SNR and predicted RH, (d) SW and predicted RH, and (f) predicted pressure and predicted RH.



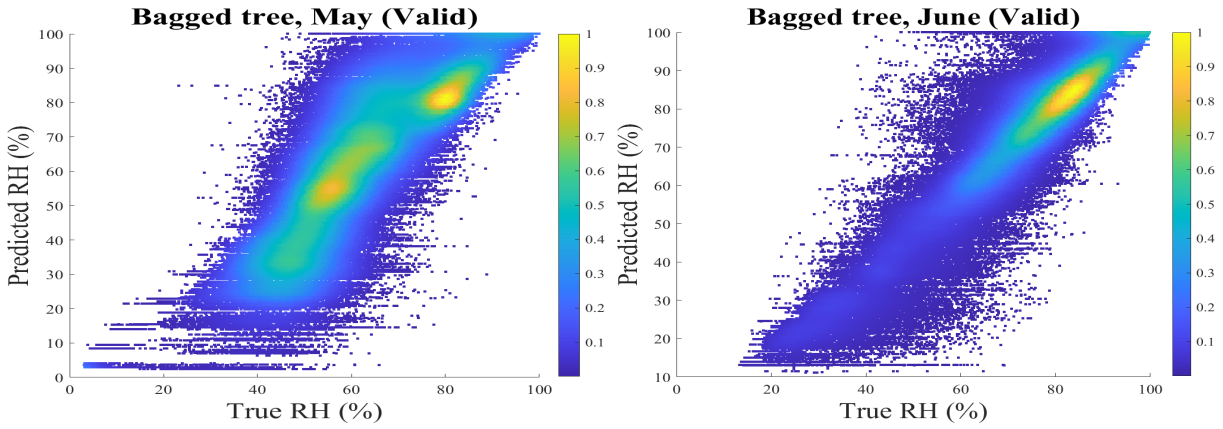
(a)

(b)



(c)

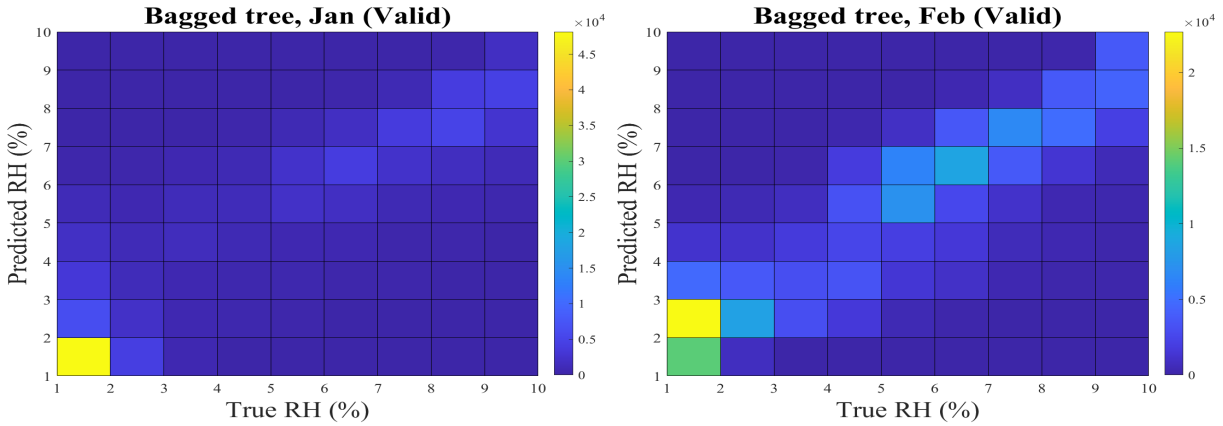
(d)



(e)

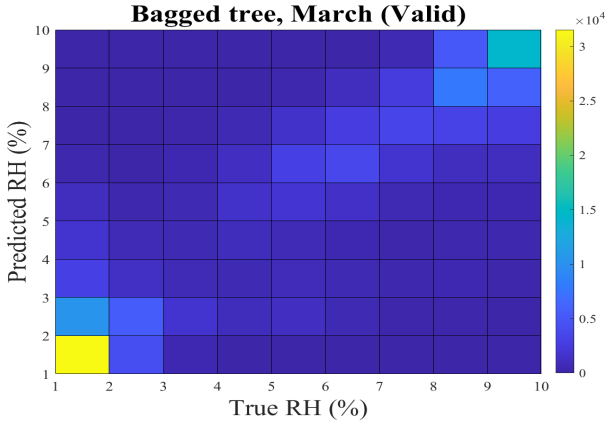
(f)

Figure 3.22: Scatter density graphs showing the cross-validation dataset's predicted vs. true humidity values using the bagged tree model for the following months: January (a), February (b), March (c), April (d), May (e), and June (f).

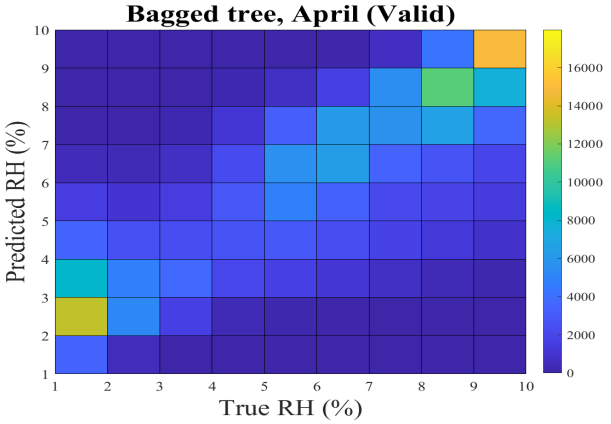


(a)

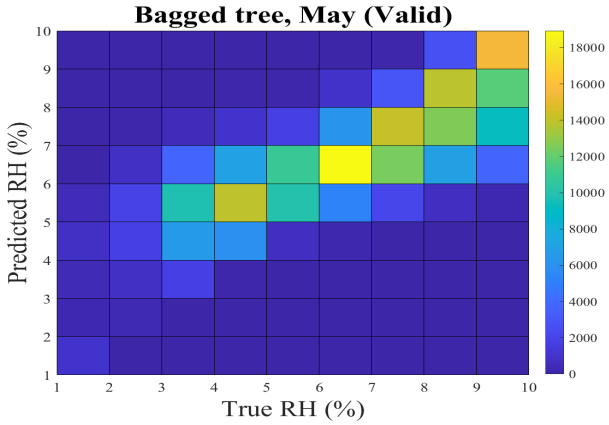
(b)



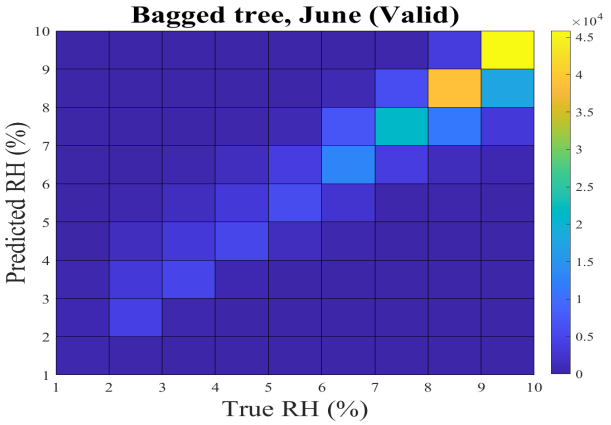
(c)



(d)

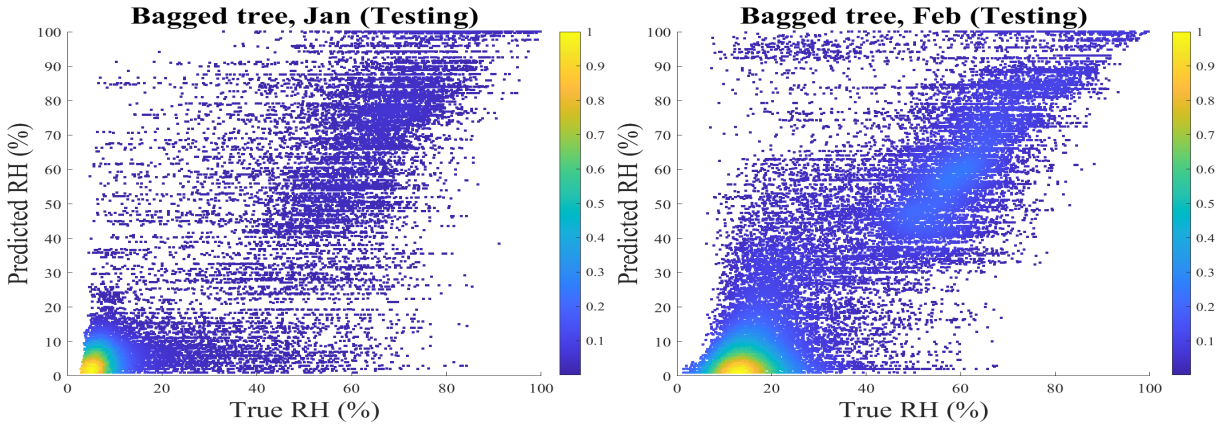


(e)



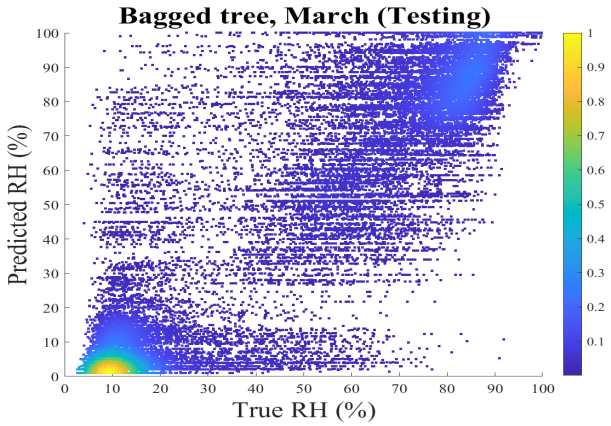
(f)

Figure 3.23: 2D histograms of predicted vs. true humidity values from the cross-validation dataset using the bagged tree model for the following months: (a) January, (b) February, (c) March, (d) April, (e) May, and (f) June.

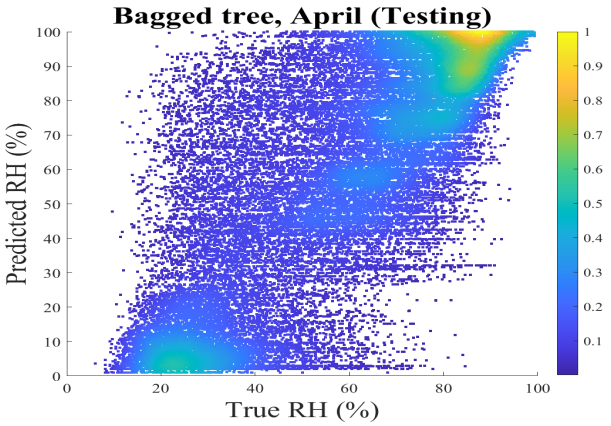


(a)

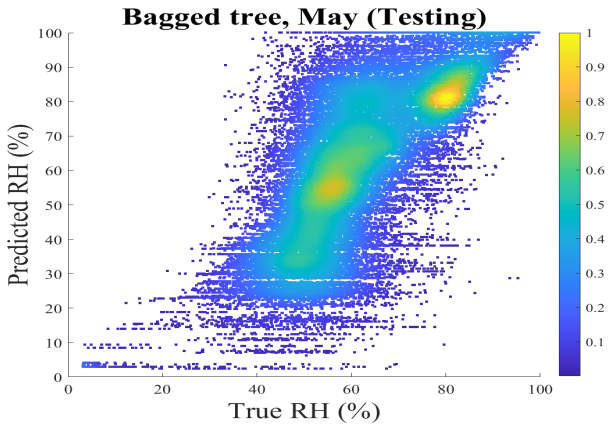
(b)



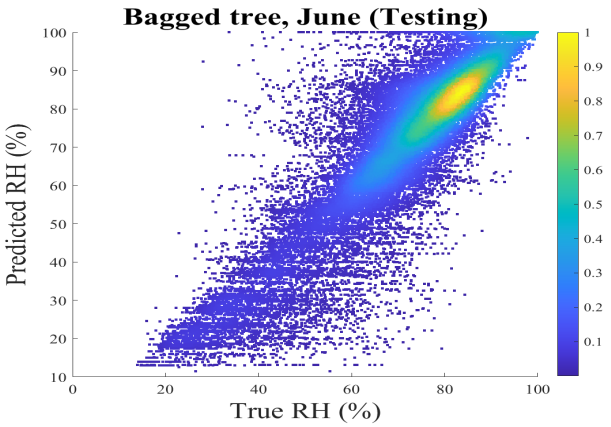
(c)



(d)

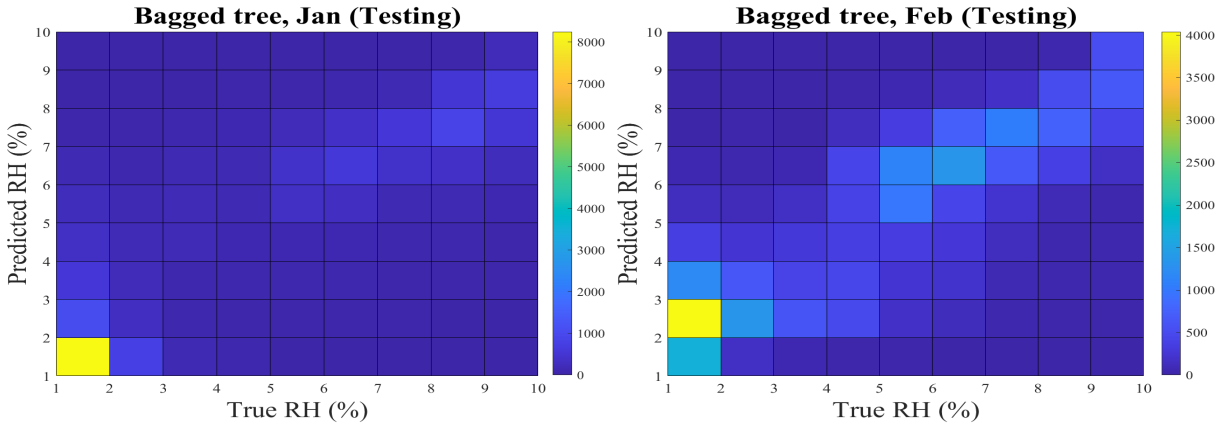


(e)



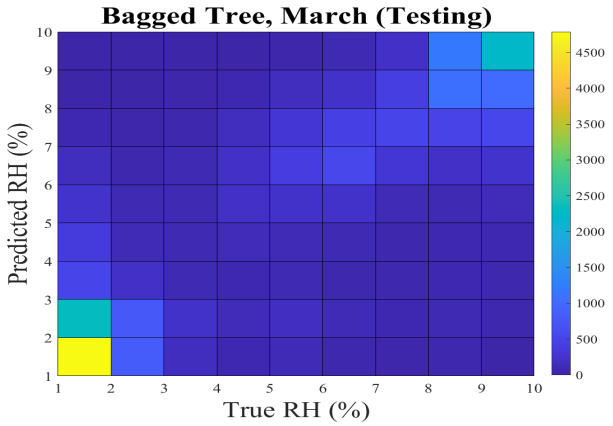
(f)

Figure 3.24: Scatter density graphs showing the test dataset's predicted versus true relative humidity results for each month separately using the Bagged Tree model: (a) January, (b) February, (c) March, (d) April, (e) May, and (f) June.

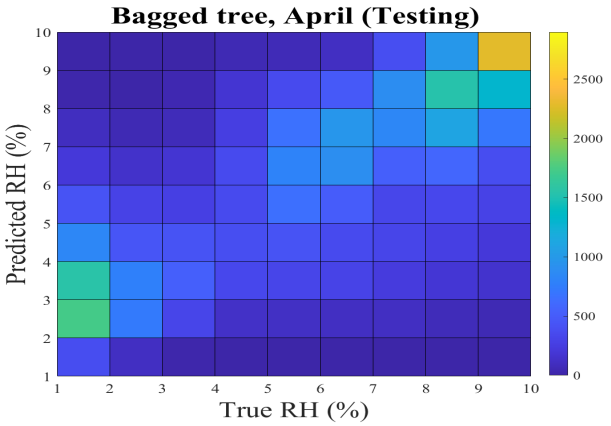


(a)

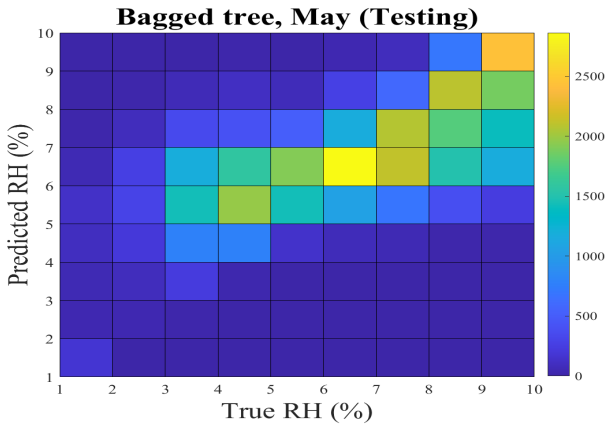
(b)



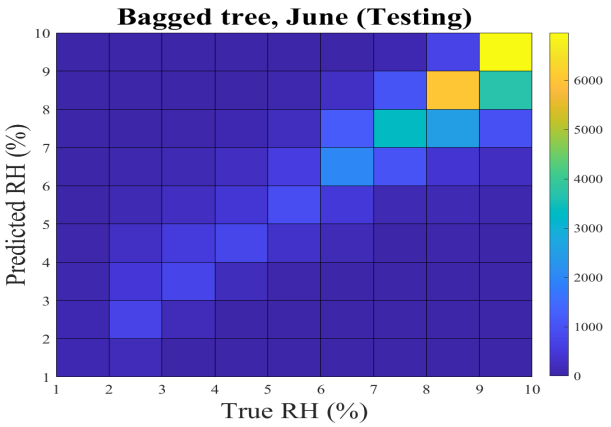
(c)



(d)



(e)



(f)

Figure 3.25: 2D histogram plots showing the estimated versus true relative humidity (RH) from the test dataset for each month independently using the Bagged Tree model: (a) January, (b) February, (c) March, (d) April, (e) May, and (f) June.

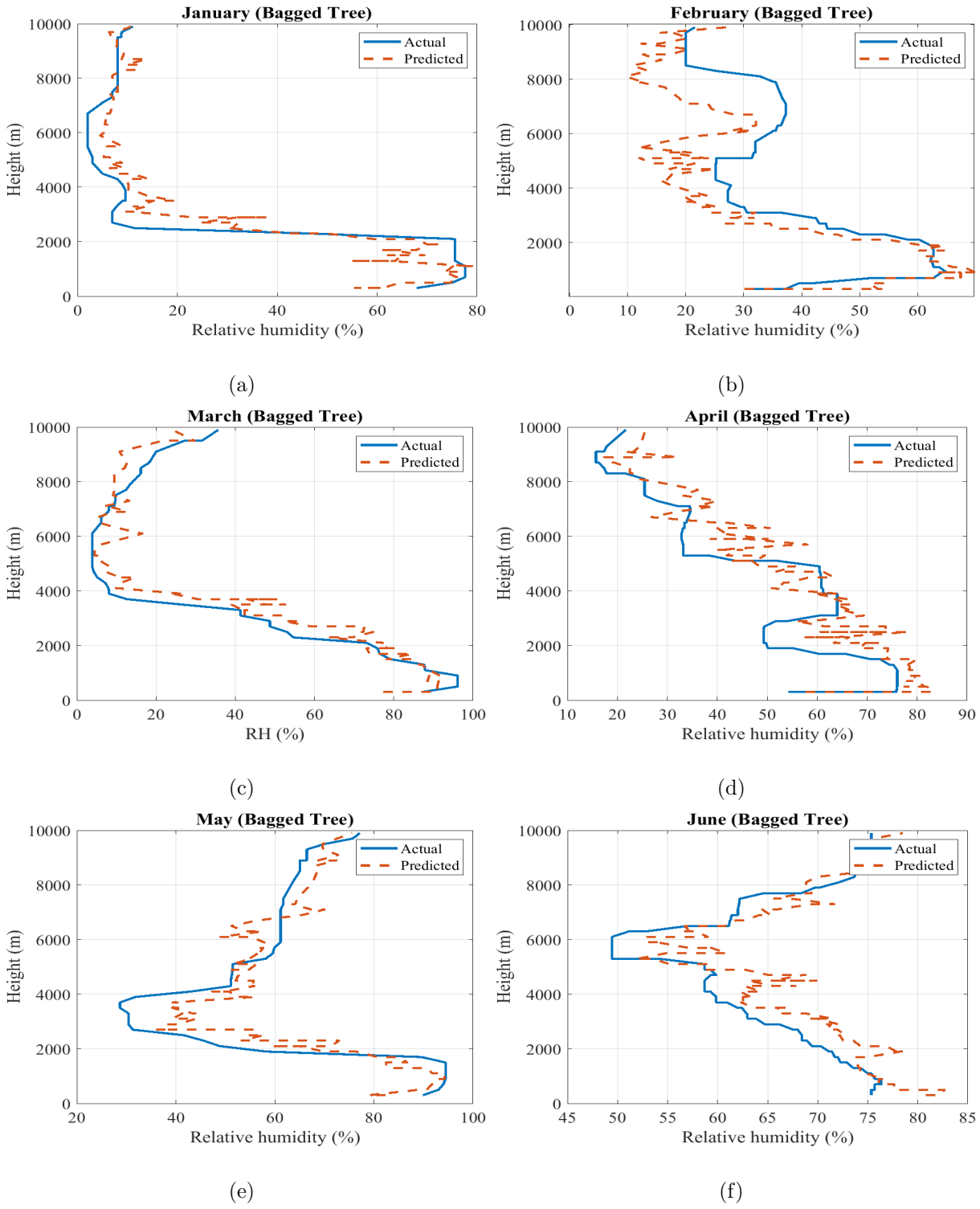


Figure 3.26: Vertical relative humidity profiles of the Bagged tree in (a) January, (b) February, (c) March, (d) April, (e) May, and (f) June.

3.4.3 Advantages and Limitations of the Study

The current study has shown numerous significant strengths and improvements in atmospheric humidity estimation. This approach uses raw moment data from a wind profiler radar as machine learning feature variables, simplifying and decreasing the model's complexity. Additionally, the lack of dependency on accurate wind estimates, temperature, and pressure data, which are sometimes unavailable, emphasizes the resilience and usefulness of the suggested technique. As part of the implementations, an intermediate training stage using synthetic pressure data initially introduced in this study was applied. This cascade ML system has demonstrated promising application potential in calculating various atmospheric parameters. Lastly, the bagging learning tree algorithm's superior computational efficiency and prediction efficacy distinguish this study and improve the reliability and performance. The limitations include the reliance on high-quality I/Q or moment data from the profiler radar, which is a significant restriction. Noise, data incompleteness, or inconsistencies in this data might impact the model's performance. Furthermore, the model's generalization ability may be limited because it was trained on data from a specific location and time. Environmental and climatic variability may challenge the model's prediction accuracy across different geographical areas and time intervals.

3.5 Conclusions

This chapter proposes a novel cascaded machine learning (ML) approach for estimating atmospheric relative humidity from wind profiler radar data. Unlike existing methods, this approach uses moment data from the raw spectrum as ML feature variables and predicts relative humidity (RH) in two cascaded steps. The bagging learning tree algorithm is identified as the most effective and computationally efficient method for predicting pressure and RH. The proposed approach improves reliability and performance compared to physical models and does not require precise wind estimates, temperature, and pressure parameters

from radiosonde data that are not always available. The results show that this approach can provide reasonably accurate humidity estimates. In addition, the cascaded ML solution produces an intermediate training stage with "synthetic" pressure data, which is applied to the humidity estimation problem for the first time. The findings of this study demonstrate the significant potential of machine learning techniques in enhancing atmospheric humidity estimation, particularly when traditional physical models face limitations due to data gaps or uncertainties. The successful application of the bagged learning tree algorithm and the innovative cascading ML approach in this context not only highlights the adaptability and robustness of ML models but also sets a precedent for future research in atmospheric science and related fields.

Chapter 4

Improvement of Cloud Liquid Water Content Profiling

4.1 Introduction

The Earth's atmosphere is a dynamic system that constantly interacts with and influences the environment and human society. Cloud Liquid Water Content (CLWC), expressed in grams per cubic meter (g/m^3), represents the liquid water concentration within a cloud and holds special significance within this system (Wallace and Hobbs 2006). Clouds' liquid water content is pivotal in numerous atmospheric and environmental processes. It defines cloud features, influences global climate patterns, and critically affects precipitation and extreme weather events, such as typhoons, hurricanes, and severe storms (Shukla and Sud 1981; Wang et al. 2021). Such climatic phenomena can dramatically impact agricultural production, infrastructure, and the economy (Ogunjo et al. 2019). Its significance extends to the Earth's hydrological cycle, impacting evaporation, condensation, and precipitation processes. This balance has implications for water reserves, agricultural yields, ecosystems, and human habitation patterns (Techel and Pielmeier 2011). In aviation, specific LWC values can lead to ice accretion on aircraft components, posing significant safety threats (Korolev et al. 2007). Additionally, inaccurate LWC observations can lead to poor predictions of visibility conditions, further increasing risks in aviation scenarios (Gultepe et al. 2019). LWC is also crucial in remote sensing. The calibration and interpretation of satellite-based instruments

heavily rely on accurate LWC measurements (Guo et al. 2023). The importance of LWC estimation extends to urban planning, especially in densely populated regions where precise LWC estimates are essential for developing effective drainage systems and flood mitigation strategies (Notaro et al. 2015).

Predicting and understanding CLWC levels remains a substantial challenge due to the influence of LWC on cloud microphysics, as a process that regulates cloud optical properties (Morrison et al. 2020). This interaction affects both shortwave and longwave radiations and affects the Earth's energy balance and major climate cycles (Brun 1989; Thies et al. 2017). This challenge is highlighted in urban areas such as Hong Kong, where diverse geography, complex terrain, and closeness to large bodies of water produce a network of microclimates, making accurate LWC estimation more challenging using traditional physical models. CLWC retrieval has achieved important advancements over the past decades. Initial efforts by (Grody 1976) using a scanning microwave spectrometer (SCAMS) established a relationship between brightness temperatures at various frequencies and CLWC. During this time, there was an advancement in determining cloud liquid water and total precipitable water (TPW), which improved both modeling and measuring capabilities (Alishouse et al. 1990; Greenwald et al. 1993; Nimnuan et al. 2017; Weng and Grody 1994; Weng et al. 1997). Notably, techniques for recovering LWC data from visible wavelengths over land were introduced and summarized in the integrated approach by (Weng and Grody 1994). This approach was effective at retrieving CLWC under diverse atmospheric conditions (Weng et al. 1997). Advances in radar technology have provided the path for new approaches to estimate LWC in clouds, such as employing single-wavelength cloud radar (Ge et al. 2023). In particular, notable work by (Grody et al. 2001) and (Weng et al. 2003) has supported obtaining CLWC from low-Earth orbit satellites. Satellite-based remote sensing, despite its broad geographic coverage, faces limitations due to its spatial resolution. This resolution is often too broad to accurately detect the rapid changes in clouds or their fine-scale structures. This limitation occurs because satellite sensors cannot identify features smaller than their

resolution threshold (Zhu et al. 2018). In contrast, ground-based data sources like CloudNet offer detailed local insights but are subject to biases such as sensor wetting and evaporation losses (Illingworth et al. 2007). Recent technological developments have combined active and passive remote sensing approaches (Dong et al. 2022; Nandan et al. 2022).

Though challenges persist, some satellite-based low-cloud liquid-water path retrieval techniques have adopted machine learning (ML) models to enhance accuracy. The study by (Kim et al. 2020) provides a clear example of this approach, introducing a machine-learning method for retrieving the Liquid Water Path (LWP). LWP measures the total amount of liquid water per unit area in a column of air extending from the cloud base to the top height, which is obtained by integrating the liquid water content (LWC) from the cloud base to the top height. The method uses data from the Spinning Enhanced Visible and Infrared Imager (SEVIRI) satellite and has been validated against ground truth data from CloudNet stations. Nevertheless, the study also reveals limitations, such as difficulties in accurately handling high LWP values due to a lack of comprehensive training samples, resulting in increased bias at these levels. Additionally, the model's performance in different geographical regions or under diverse cloud conditions remains to be tested, suggesting that the method requires further refinement and validation.

The study in this chapter addresses these challenges with a novel ML approach for predicting CLWC profiles using radar profilers, utilizing a large dataset of 2,592,950 samples. This is a significant improvement over previous studies, many of which relied on smaller, more regionally constrained datasets and did not specifically use radar profilers, limiting their accuracy and applicability across diverse atmospheric conditions. In contrast, this study focuses on radar profiler data, ensuring that the model is robust and generalizable in real-world meteorological settings. One of the key advancements of this work is the introduction of innovative data preprocessing techniques that significantly improve the correlation between

local radiosonde measurements and global satellite-based ERA5 data. Previous models often suffered from biases due to the mismatch between different data sources, resulting in less accurate CLWC predictions. This study achieves greater accuracy and consistency by addressing these biases, especially in complex geographic regions like Hong Kong, where microclimates can complicate traditional models. Furthermore, many prior studies relied on static physical models, which are limited by assumptions about atmospheric behavior and often struggle to capture the dynamic, non-linear relationships between atmospheric parameters. This study, however, leverages the power of machine learning to model these complex relationships more effectively, particularly by utilizing radar profiler data, which provides a more direct and reliable source for CLWC prediction. The scale and depth of the dataset used in this study surpass previous efforts, enabling more comprehensive training of the ML model, which leads to better performance in real-world applications. In comparison to earlier works, this study provides improved accuracy, better generalizability, and the ability to handle a wide range of atmospheric conditions—particularly important for regions with complex terrains and climates. The novel approach introduced here sets a new standard for CLWC estimation by addressing the weaknesses of past methodologies and leveraging radar profiler data to offer a robust, scalable solution.

This chapter is organized as follows: Section 4.2 introduces the overview of the approach and processing flow. Section 4.3 provides detailed descriptions of the datasets used in this study. Section 4.4 covers the climatology of the Hong Kong area and data groups. Section 4.5 provides detailed information about the approach and methodology developed. The results and discussions are presented in Section 4.6. Further conclusions and a summary are provided in Section 4.7.

4.2 Overview of Approach and Processing Flow

ERA5 climate dataset from the European Centre for Medium-Range Weather Forecasts (ECMWF) is used as a basic data source. This dataset includes atmospheric measurements such as temperature, pressure, and humidity, which are essential for the detailed profiling of CLWC. The ERA5 dataset is carefully cross-verified using radiosonde observations from the Hong Kong Observatory (HKO). These radiosonde measurements, known for their precision, are interpolated using machine learning algorithms, guaranteeing that the ERA5 data quality is consistent with the observed atmospheric parameters. A critical component is the data cleaning procedure, in which a carefully selected metaheuristic optimization method is used to refine the ERA5 dataset parameters. This approach improves the dataset's quality by enhancing the correlation between the ERA5 input features, including pressure, temperature, relative humidity, and specific humidity, and the CLWC values. This ensures that the machine-learning algorithms are trained on a representative dataset with minimal noise. With the pre-processed dataset, ML algorithms are then trained to predict the CLWC levels. Figure 4.1 provides a flow chart that illustrates all the steps involved in the methodology. Each of these steps will be explored in greater detail in the following sections.

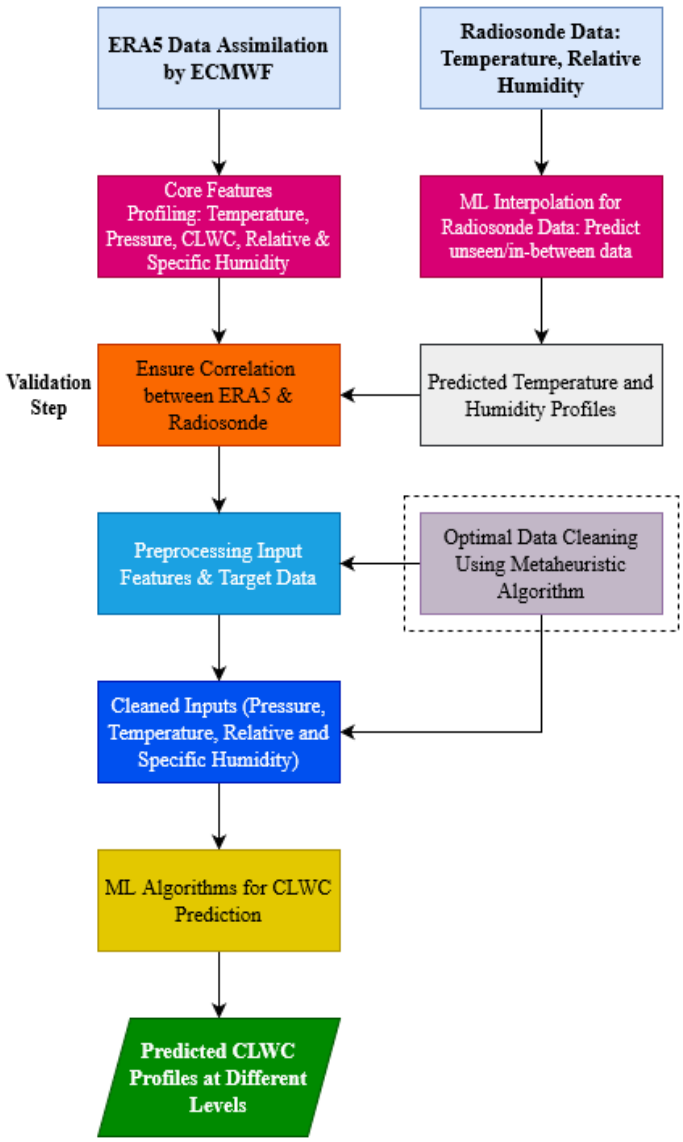


Figure 4.1: A flowchart illustrating the complete processing flow for predicting Cloud Liquid Water Content (CLWC) profiles.

4.3 Data Description and Analysis

4.3.1 ERA5 Data

The ERA5 dataset from the ECMWF offers full atmospheric coverage from pole to equator. Its temporal resolution is achieved through hourly updates of various factors relating to

the atmosphere, land, and seas. This resolution is beneficial for observing the intricate nature of weather systems and detecting subtle climate change variations (Albergel et al. 2018; Malardel et al. 2016). Regarding spatial resolution, ERA5 datasets cover about 31 km horizontally and 37 vertical levels up to an altitude of 80 km. The data is distributed on a 0.25° x 0.25° grid. ERA5 includes a 10-member ensemble to enhance its robustness, offering insights into the reanalysis's uncertainty, both temporally and geographically (Malardel et al. 2016). The study focuses on the Hong Kong region for the year 2021. The ERA5 data selected for this study encompassed the exact latitude and longitude coordinates forming a targeted grid over the Hong Kong area at the following coordinates: 113.49E - 22.33N, 113.74E - 22.33N, 113.99E - 22.33N, 114.24E - 22.33N, 113.49E - 22.08N, 113.74E - 22.08N, 113.99E - 22.08N, 114.24E - 22.08N. The investigation spanned a range of atmospheric parameters at 37 different pressure levels, encompassing temperature profiles, relative and specific humidity levels, and cloud liquid water content profiles derived using the ideal gas law and specific cloud liquid water content profiles. Figure 4.2 shows the study region with all the longitudes and latitude coordinates in the Hong Kong area.

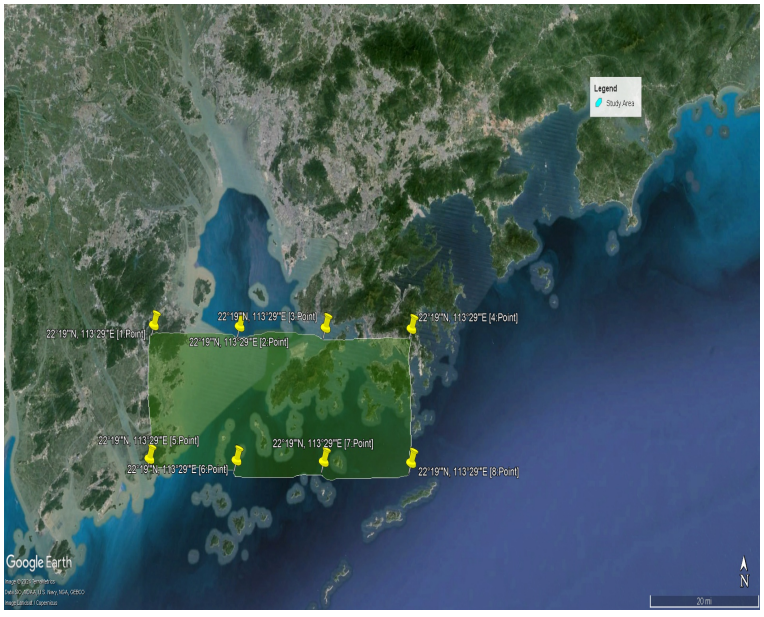


Figure 4.2: Map showing the study region within Hong Kong, highlighted by the shaded area. The coordinates mark the boundaries of the area.

Figure 4.3 plots the frequency distributions of CLWC (Kg/m^3), relative humidity (%), specific humidity (Kg/kg), and temperature (K) from the ERA5 dataset for 2021 in the HK area to represent these data visually at various pressure levels. The histograms display hourly data occurrences, providing a detailed view of the atmospheric parameters throughout the year without averaging into daily or monthly means. The CLWC plot indicates a predominance of lower values, suggesting that less cloud water content is more frequent. The distribution for relative humidity shows higher frequencies of lower values, which gradually decrease as humidity levels increase, while specific humidity trends towards lower values with an extended tail at higher levels, reflecting varied atmospheric moisture conditions. The temperature's multimodal distribution reflects the diverse thermal conditions experienced over different seasons and geographical locations throughout the year. These detailed distributions capture the diversity and consistency of atmospheric conditions and serve as a statistical foundation for this climatological investigation.

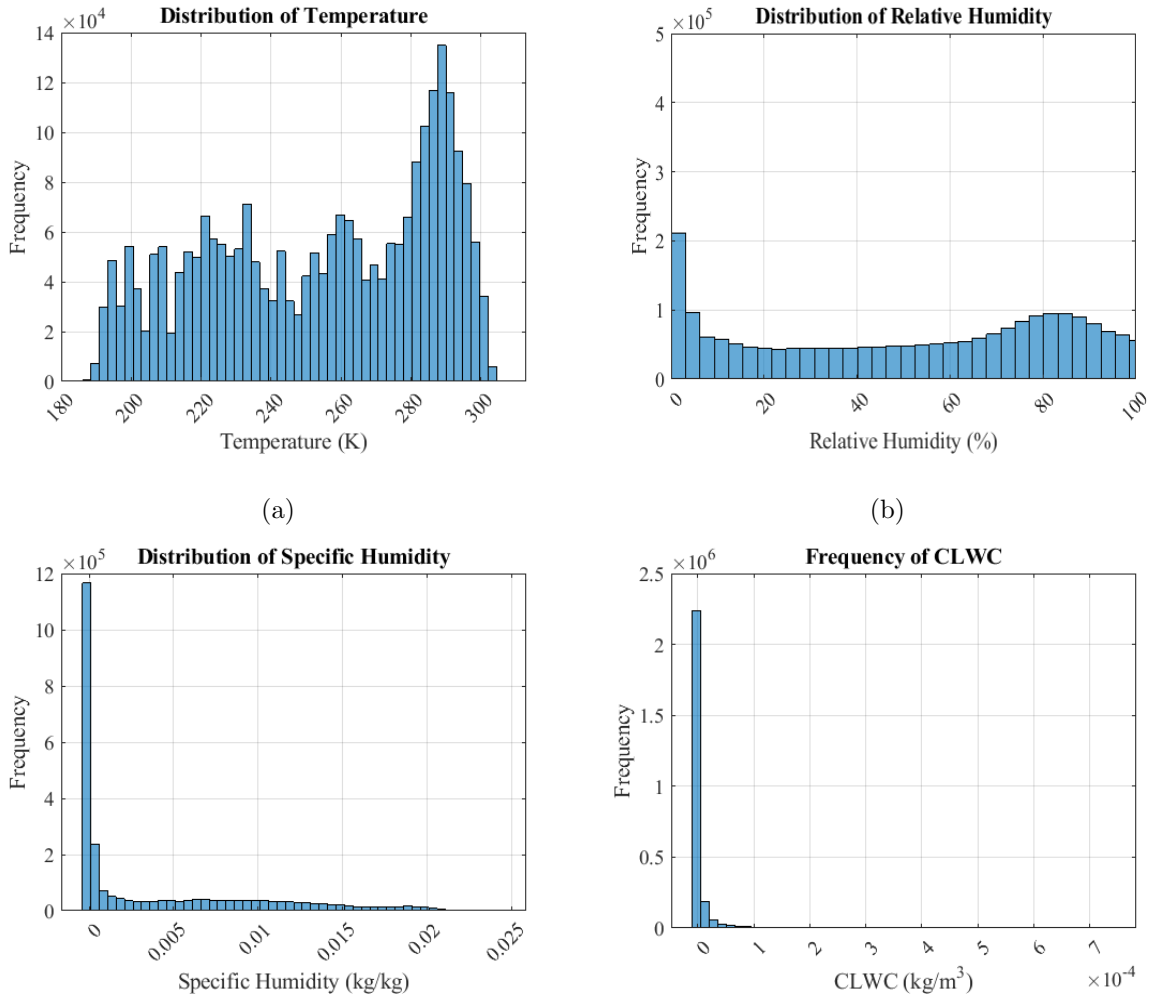


Figure 4.3: Frequency distributions of atmospheric parameters in the ERA5 dataset for the Hong Kong area throughout 2021. The histograms display the hourly occurrences of atmospheric parameters at various pressure levels, illustrating the distributions of (a) temperature (K), (b) relative humidity (%), (c) specific humidity (Kg/kg), and (d) Cloud Liquid Water Content (CLWC) (Kg/m^3).

4.3.2 Radiosonde Data

Radiosonde data from the Hong Kong Observatory (HKO) served as the ground truth to validate the integrity of the ERA5 dataset. These observations are obtained every twelve

hours, with each sample comprising around 2,331 measurements from 66 meters up to 30,215 meters above ground level. In addition to the primary observations of temperature, relative humidity, and pressure, the radiosondes provide data on wind direction, dew point, and wind speed, which helps to understand the atmospheric profile up to about 24 kilometers above ground level (Amaireh et al. 2023a).

4.4 Climatology and Data Groups in Hong Kong

Hong Kong is characterized by a subtropical climate that shifts towards temperate for nearly half the year. Gentle winds, abundant sunlight, and moderate temperatures mark the late autumn and early winter months, making November and December the most favorable. Conversely, January and February often bring cloudier skies and cold fronts that can drop temperatures below 10°C in urban areas and below freezing in higher terrains. The spring months introduce higher humidity and reduced visibility, complicating air and ferry transportation. Conditions are hot and humid from May to August, interrupted by frequent morning showers and thunderstorms. Afternoon temperatures often exceed 31°C, with nights around 26°C. July sometimes offers a brief break with a dry spell lasting one or two weeks. Typhoon season extends from May to November, peaking between July and September. Annually, about 30 tropical cyclones form, with half reaching typhoon strength, significantly influencing local weather when they approach or pass near Hong Kong. The region's annual rainfall shows dramatic spatial variation, heavily concentrated between May and September (Hong Kong Observatory 2021).

The year 2021, on the other hand, was consistently warm in Hong Kong. January was cooler and drier than usual, with an average temperature of 16.2°C and lots of sunshine. The warming trend began in February due to a weak northeast monsoon, leading to an unusually mild winter. March was the hottest on record, with significantly higher temperatures

and minimal rainfall. This continued into April, marked by dry conditions and an upper-air anticyclone. Because of low rainfall, May was the warmest on record, making spring (March to May) the warmest ever. Heavy rain from tropical influences arrived in June, but temperatures remained above average. July's heat was caused by a strong upper-air anticyclone, leading to high temperatures and slightly less rain than usual. August had more clouds and some heavy rains, but overall, there was less rainfall than expected. September's dry conditions resulted in record heat, influenced by the subtropical ridge. October saw increased rainfall due to tropical cyclones, making it a wetter month. In November, the weather returned to being sunny and dry. December's high temperatures helped make 2021 the hottest year on record for Hong Kong, highlighting a year of intense heat and notable climate differences from the average (Hong Kong Observatory 2021).

Figure 4.4 shows the monthly variation of four meteorological parameters in 2021. Temperature remained stable throughout the year, with slight changes with seasons. Relative humidity varied widely due to temporary weather patterns. Specific humidity was more consistent, with occasional spikes with increased atmosphere moisture. The CLWC subfigure shows a distinct seasonal trend in CLWC, with major peaks occurring between March and October. These months have higher median values and a wider range of CLWC, indicating a time of more frequent cloud formation and more precipitation events. Outliers, which appear across all months, represent periodic exceptional CLWC values that might be connected to specific weather phenomena. The range of data within each month and the use of a logarithmic scale highlight CLWC's significant variability.

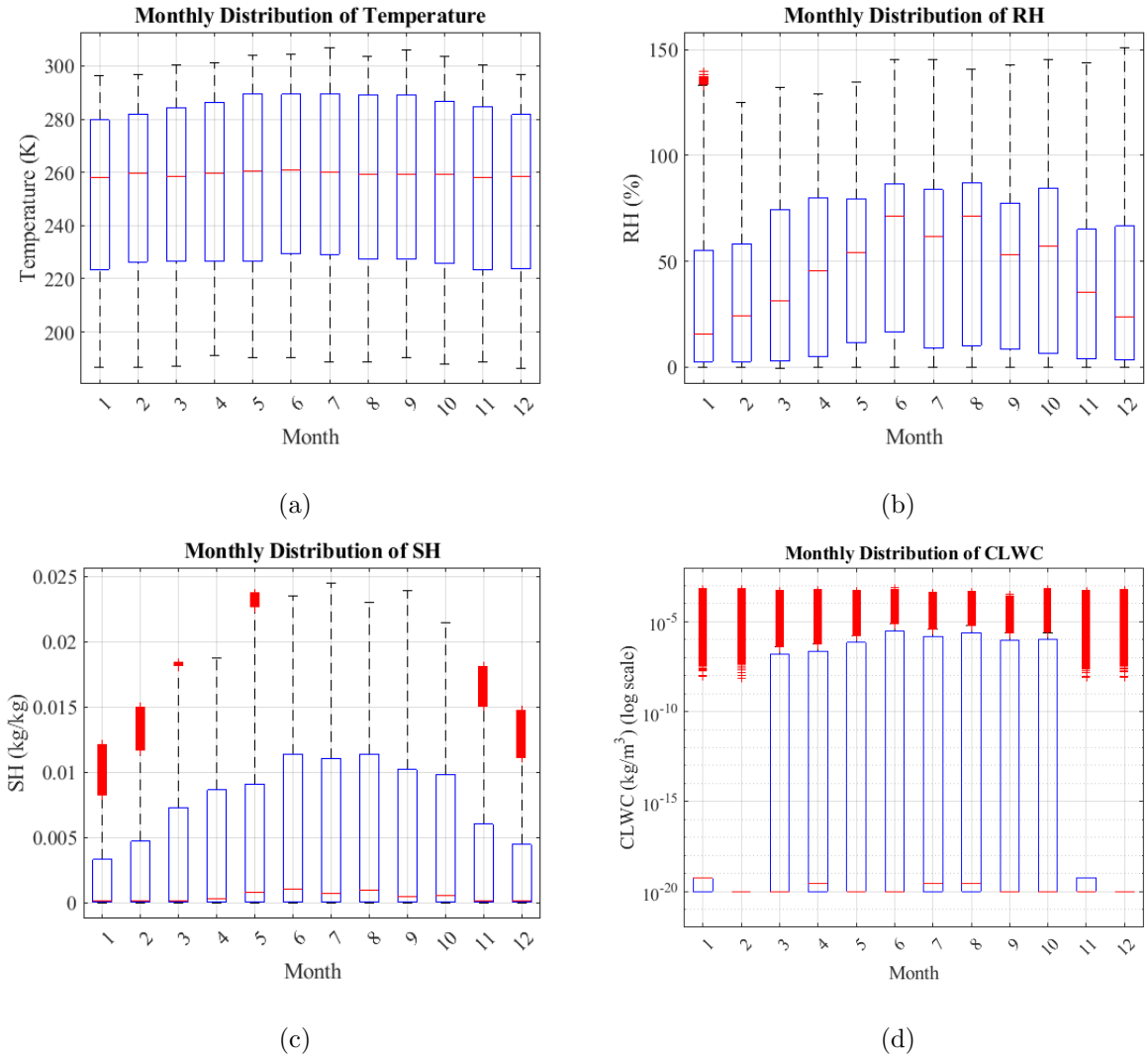


Figure 4.4: Monthly Distributions of Key Meteorological Parameters in ERA5 for 2021. Subfigures display (a) temperature (K), (b) relative humidity (%), (c) specific humidity (kg/kg), and (d) cloud liquid water content (CLWC)(Kg/m^3), showcasing seasonal variability and trends.

A detailed analysis of each month's weather parameters is important for effective data grouping and training of the ML algorithm. It examines meteorological factors from the ERA5 dataset that have influences. Table 4.1 provides a statistical summary of the selected meteorological parameters from the ERA5 dataset for each month of the year. It includes the

mean and standard deviation (STD) of Cloud Liquid Water Content (CLWC), temperature, specific humidity (SH), and relative humidity (RH).

Next, the ML training approach categorizes months with similar weather patterns into groups, i.e., January and February from Group 1, characterized by cooler, drier conditions. Group 2 includes March and April, marking the start of warmer weather and increasing humidity, signaling spring. Group 3, from May to August, represents the summer peak with the highest moisture and temperatures. Group 4, comprising September and October, reflects the transition from peak summer. Finally, Group 5, with November and December, indicates the onset of cooler, drier winter weather. This grouping allows machine learning models to be tailored to the data trends of each group category.

Table 4.1: Monthly mean and standard deviation values of CLWC, Temperature, Specific Humidity, and Relative Humidity for Hong Kong in 2021 based on ERA5 data.

Month	CLWC Mean (kg/mˆ3)	CLWC STD (kg/mˆ3)	Temp Mean (K)	Temp STD (K)	SH Mean (kg/kg)	SH STD (kg/kg)	RH Mean (%)	RH STD (%)
Jan	5.26E-06	3.13E-05	250.64	31.08	0.0018	0.0029	30.24	31.37
Feb	4.66E-06	2.54E-05	252.24	31.33	0.0024	0.0035	32.71	31.00
Mar	7.13E-06	2.77E-05	253.38	31.80	0.0034	0.0047	38.63	35.38
Apr	8.66E-06	3.26E-05	254.29	32.28	0.0039	0.0052	45.08	36.09
May	4.28E-06	1.59E-05	256.08	33.48	0.0047	0.0063	49.12	34.71
Jun	6.56E-06	2.36E-05	256.71	33.30	0.0054	0.0068	56.16	37.33
Jul	4.93E-06	1.82E-05	256.80	33.26	0.0054	0.0069	51.26	37.22
Aug	5.71E-06	2.00E-05	256.14	33.19	0.0055	0.0068	55.90	37.59
Sep	3.15E-06	1.03E-05	255.98	33.33	0.0049	0.0065	46.83	34.11
Oct	8.75E-06	3.28E-05	254.68	32.56	0.0045	0.0057	49.97	37.36
Nov	3.53E-06	2.22E-05	252.85	31.90	0.0030	0.0041	37.87	32.02
Dec	5.94E-06	2.83E-05	251.85	30.95	0.0024	0.0035	35.17	33.84

4.5 Detailed Methodology

4.5.1 Correlation Between ERA5 and Radiosonde Data

Establishing a reliable correlation between the data sources is crucial for proving that ERA5 can dependably recreate actual atmospheric conditions observed by verified ground-based instruments. The agreement between ERA5 and radiosonde measurements is validated by calculating the correlation factors of temperature and relative humidity profiles obtained

from ERA5 and radiosonde observations. Among the challenges of such correlation is the problem of matching each sensor's different sample times and altitudes. Attempts to align these differences using methods such as interpolation and nearest-value assignment could have been more effective as they resulted in poor correlations and low confidence in the quality of the data alignment for both temperature and relative humidity.

To address the challenge, Machine Learning (ML) techniques were used to predict the radiosonde data values for any date at 0 and 12 hours, as well as at various altitudes, even without direct measurements. Figure 4.5 illustrates the flow chart detailing the process of interpolating the in-between data for radiosonde temperature and relative humidity. This is achieved through two entirely separate training processes, with inputs including altitude, months, days, and hours. The outputs for training the ML models are the true radiosonde values: temperature for the first model and relative humidity for the second. Two Fine Tree decision tree models, with a defined minimum leaf size of 4, were developed to predict and interpolate temperature and relative humidity. These algorithms were designed to output weather parameters that align with those recorded by radiosonde instruments. A total of 958,617 samples were incorporated throughout the training and testing phases. The data was partitioned such that 15% constituted the testing dataset while the remaining 85% was used for training. Additionally, the training process was enhanced with 5-fold cross-validation to ensure the robustness and generalizability of the models.

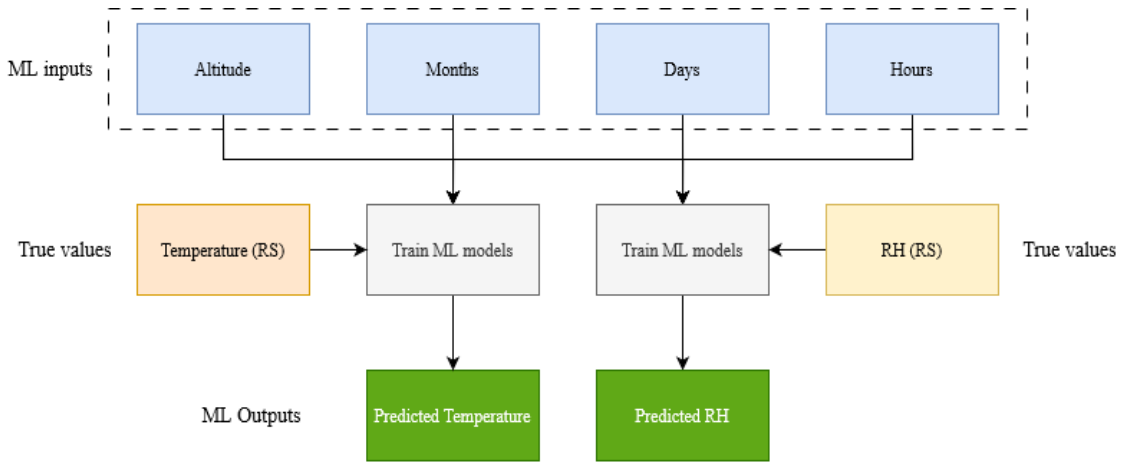


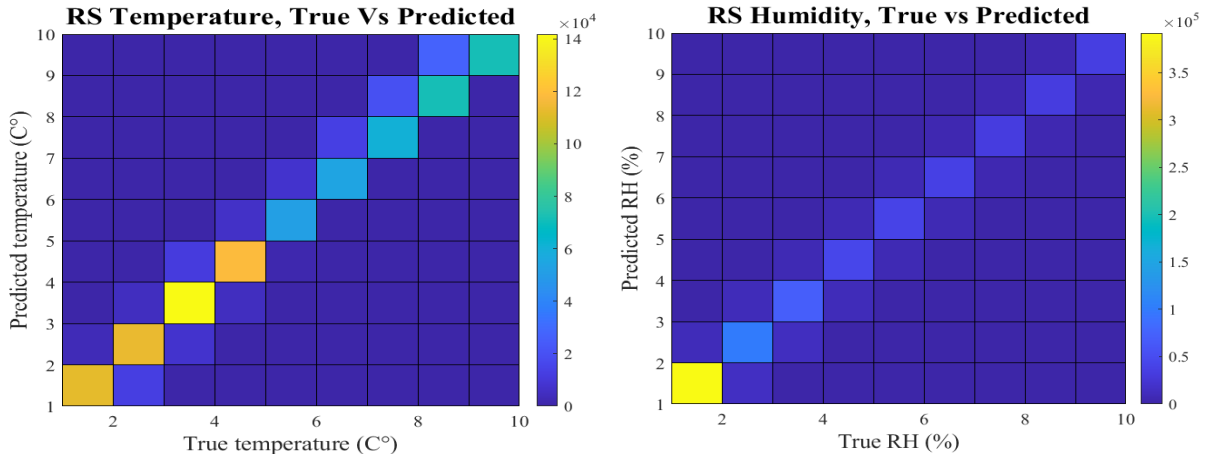
Figure 4.5: Flow chart representing the separate ML training processes for interpolating radiosonde temperature and relative humidity data. Inputs include altitude, months, days, and hours, with true radiosonde values serving as outputs for ML models' training.

According to Table 4.2, which presents the performance of the trained ML models for interpolating temperature and relative humidity in the testing dataset, the Fine Tree (FT) model demonstrated high accuracy. It achieved an R-squared value of 0.9983 and a correlation coefficient of 0.9992 in estimating temperature. Despite the various variables affecting atmospheric moisture, it maintained its accuracy in humidity prediction, as indicated by an R-squared of 0.9894 and a correlation value of 0.9965. The models' robust predictive ability is shown by low RMSE and MAE values for temperature (1.3007 °C and 1.0164 °C, respectively) and slightly higher values for relative humidity (3.0501% and 1.8297%, respectively).

Table 4.2: The performance of machine learning models in interpolating the temperature and relative humidity for the radiosonde data in the testing dataset.

Model Type	RMSE	MSE	RSquared	MAE	Correlation coefficient
Fine Tree (Temperature (℃))	1.3007	1.6917	0.9984	1.0164	0.9992
Fine Tree (Relative Humidity (%))	3.0501	9.3032	0.9894	1.8300	0.9965

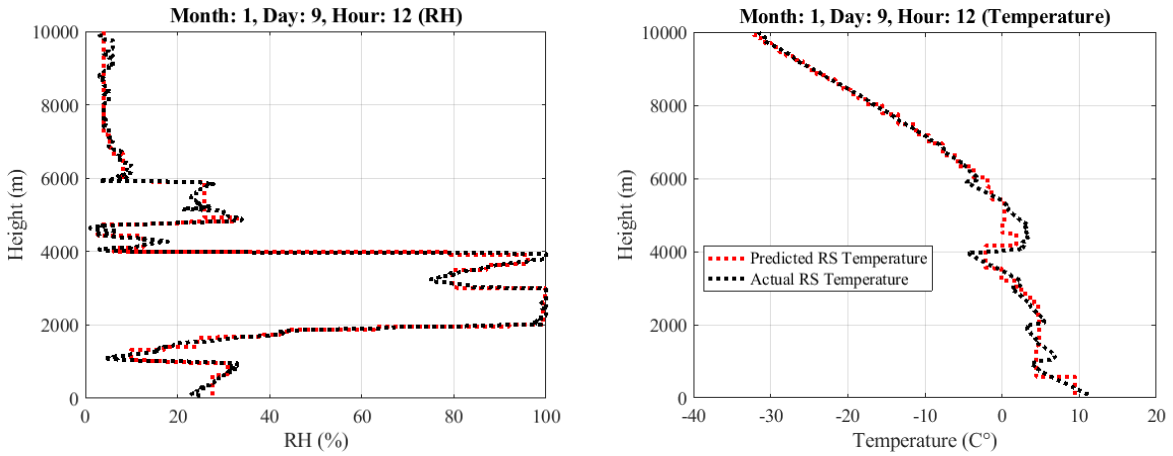
Figure 4.6 shows density plots that display the accuracy of decision tree models in predicting temperature and humidity values. In both plots, the x-axis represents the true classes, and the y-axis represents the predicted classes. Classes were used instead of real values to simplify the visualization and highlight prediction accuracy. The diagonal trend indicates that predicted values closely align with actual measurements. The color intensity reveals the frequency of predictions, with brighter areas marking the most common predictions. The temperature plot shows a more even distribution, suggesting high accuracy across various values. The humidity plot, with its focused bright area, indicates precise predictions within a narrower range.



(a) (b)

Figure 4.6: 2D histogram plots showing the interpolated (predicted) vs. true values for (a) temperature and (b) relative humidity of the testing radiosonde dataset using fine trees. The color map indicates the frequency of data points, with different colors representing varying densities.

Figure 4.7 presents a comparison of the ML-based interpolation method against actual radiosonde profiles for relative humidity and temperature. The decision tree model's interpolation for these parameters shows high precision, closely matching the actual data. Specifically, Figure 4.7 (a) illustrates how well the model captures the variability of humidity with altitude, while Figure 4.7 (b) demonstrates the model's ability to accurately reproduce temperature profiles, reflecting sharp variations across different altitudes. For the profiles shown in this figure, the ML model achieves correlation coefficients of 0.99422 for temperature and 0.9953 for relative humidity. The values demonstrate the higher prediction capability of the fine tree for the shown data.



(a) (b)

Figure 4.7: Comparison of the ML-based interpolation method against actual radiosonde profiles for (a) relative humidity and (b) temperature. The predicted data via fine trees strongly correlates with actual data (temperature: 0.99422, RH: 0.9953).

These results verified the fine trees' ability to predict temperature and relative humidity at any time and any height within the 0-25 km range. Thus it is used to interpolate temperature and relative humidity within the radiosonde dataset. As a result, a matched dataset was created that included temperature and relative humidity measurements from both ERA5 and radiosonde data at the same altitudes, dates, and times. The selection of ERA5 data was strategically based on the coordinates of the radiosonde data in Hong Kong, specifically North 22.4°, West 113.9°, South 22.3°, and East 114°.

The next step is to calculate the correlation coefficient between the temperature and relative humidity from the interpolated radiosonde data and ERA5 sources for January, February, March, April, May, and June datasets, as well as for all six months combined, as shown in Table 4.3. According to Table 4.3, the calculated correlation coefficients from January to June 2021 displayed an average of 0.7808 for relative humidity when comparing ERA5 and radiosonde data. Monthly correlations for this period showed values of 0.7435 in January, 0.7495 in February, 0.7697 in March, 0.7404 in April, 0.7976 in May, and 0.8159 in June, respectively. Temperature data correlations for the same months were consistently

higher, with January at 0.9486 and June at 0.96666, indicating a strong alignment of the two data sources. Only the first six months of temperature and relative humidity data are mentioned because this is all the radiosonde data obtained from the Hong Kong Observatory (HKO).

Table 4.3: Correlation coefficients between ERA5 and radiosonde measurements.

	Temperature correlation	Relative humidity correlation
January	0.9486	0.7435
February	0.9570	0.7495
March	0.9683	0.7697
April	0.9686	0.7404
May	0.9686	0.7976
June	0.9667	0.8159
All six months	0.9632	0.7808

In addition, Figure 4.8 compares the profiles of temperature and relative humidity from ERA5 measurements and predicted radiosonde values using a machine learning interpolation method. Profiles 1 and 2 exhibit a strong correlation in temperature data and show correlation coefficients of 0.96959 and 0.97004, respectively, which indicate high accuracy in ERA5-based profile estimates across different altitudes. Regarding relative humidity, Profiles 3 and 4 also show significant agreement, with correlation coefficients of 0.954 and 0.933, respectively, despite the inherent variability of atmospheric moisture.

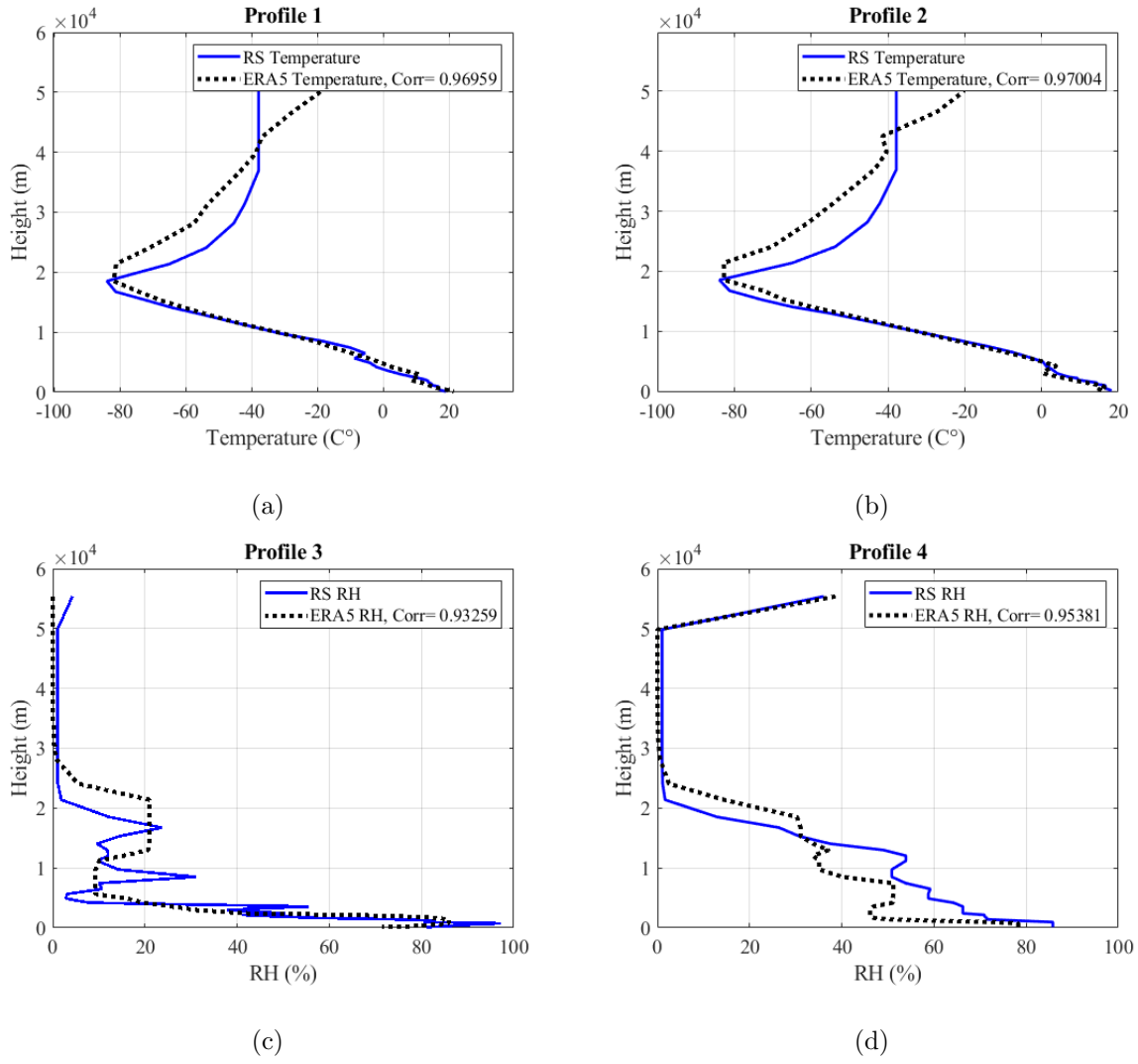


Figure 4.8: Atmospheric parameter profiles from ERA5 and Radiosonde Data. Subfigures (a) and (b) show temperature profiles 1 and 2, while subfigures (c) and (d) illustrate relative humidity profiles 3 and 4.

4.5.2 Data Quality Control and Prepossessing

4.5.2.1 Data Quality Enhancement: Overview

Outlier data can significantly impact the reliability and accuracy of predicting results (Teh et al. 2020). Excessive readings can cause these outliers due to exposure to sunlight or malfunctions working during critical data-collecting periods (Teh et al. 2020). Given these

challenges, data cleaning has evolved as an essential element of the preprocessing step. Data cleaning improves model accuracy, including removing or correcting data anomalies. It also enhances model robustness and interpretability, allowing more precise and more relevant insights into the relationships between input features and outputs (Kumar and Khosla 2018). Advanced metaheuristic algorithms are used in the data quality control process. Metaheuristic algorithms are motivated by high-level methods, allowing them to explore significant and complex solution spaces. These methods often involve generating diverse initial solutions, which are then iteratively improved through processes inspired by natural phenomena or human-designed heuristics (Amaireh et al. 2023c). Metaheuristics' fundamental strength resides in its ability to balance exploration and exploitation. This dual strategy enables them to avoid local optima while aiming for global optima.

In this study, a hybrid optimization algorithm combining Antlion (ALO) and Grasshopper (GOA) algorithms was enhanced for better performance (Amaireh et al. 2019, 2023c). The hybrid algorithm balances exploration and exploitation effectively, offering fast convergence in high-dimensional problem spaces. By leveraging the strengths of ALO's exploitation capabilities and GOA's exploration techniques, the algorithm is well-suited for data cleaning tasks, ensuring improved data reliability and quality.

4.5.2.2 Metaheuristic Data Preprocessing

As part of the hybrid optimization algorithm, an objective function is formulated as in equation 4.1:

$$fitness = \min\left(\frac{1}{|\text{corr}(cleaned\_input\_feature, Output\_target)|}\right) \tag{4.1}$$

This function aims to minimize the inverse of the absolute correlation between the cleaned input feature and the output target. The objective function is critical in guiding two key preprocessing steps, which are carried out with the help of two MATLAB functions; filloutliers and smoothdata. The filloutliers function is used to handle outliers, which is necessary

for identifying and correcting discrepancies in data, resulting in an intermediate cleaned dataset, cleaned_input_feature0. The smoothdata function then addresses short-term fluctuations, resulting in the final preprocessed feature set, cleaned_input_feature. Overall, the preprocessing is implemented using the following algorithm:

Metaheuristic Optimization for Data Preprocessing

Require: Dataset from weather sensors

Ensure: Optimized preprocessed dataset

- Initialize the metaheuristic algorithm parameters
- Generate initial population of solutions for input feature cleaning
- Define the objective function:

$$fitness(solution) = \frac{1}{|\text{corr}(cleaned\_input\_feature(solution), Target)|}$$

- **while** not end of algorithm **do**
  - For each solution in population do
    - Apply `filloutliers` function with a method from {'linear', 'spline', ..., 'makima'}
    - Apply `smoothdata` function with a method from {'movmean', 'movmedian', ..., 'sgolay'}
    - Calculate fitness of the cleaned input feature
  - Select best-performing solutions based on fitness scores
  - Perform crossover and mutation on selected solutions to create new population
  - Check termination criteria for input feature cleaning:

    if criteria met **then**

    Break and proceed to target data cleaning

else

Continue with new population

end if

- Initialize parameters for target data cleaning using optimized input features
- Repeat steps for target data cleaning
- Apply best cleaning parameters from both stages to the dataset
- Return the optimized preprocessed dataset

end while

The algorithm optimizes the correlation between the input features and the output target. The optimization process initializes the parameters for the metaheuristic algorithm, which prepares a diverse pool of potential solutions for cleaning the input features. The fitness of each solution is evaluated based on the improvement of the input-target correlation scores. Solutions with higher fitness are further refined. A termination check assesses whether the criteria for the first stage have been met. If not, the process is repeated. Once the requirements are met, it transitions to the second stage, which focuses on cleaning the target data. After completing both steps, optimum cleaning parameters are applied to the dataset.

Figure 4.9 shows a flowchart that displays the complete optimization process, from the start of the metaheuristic algorithm to the iterative enhancement of solutions to the application of optimal parameters.

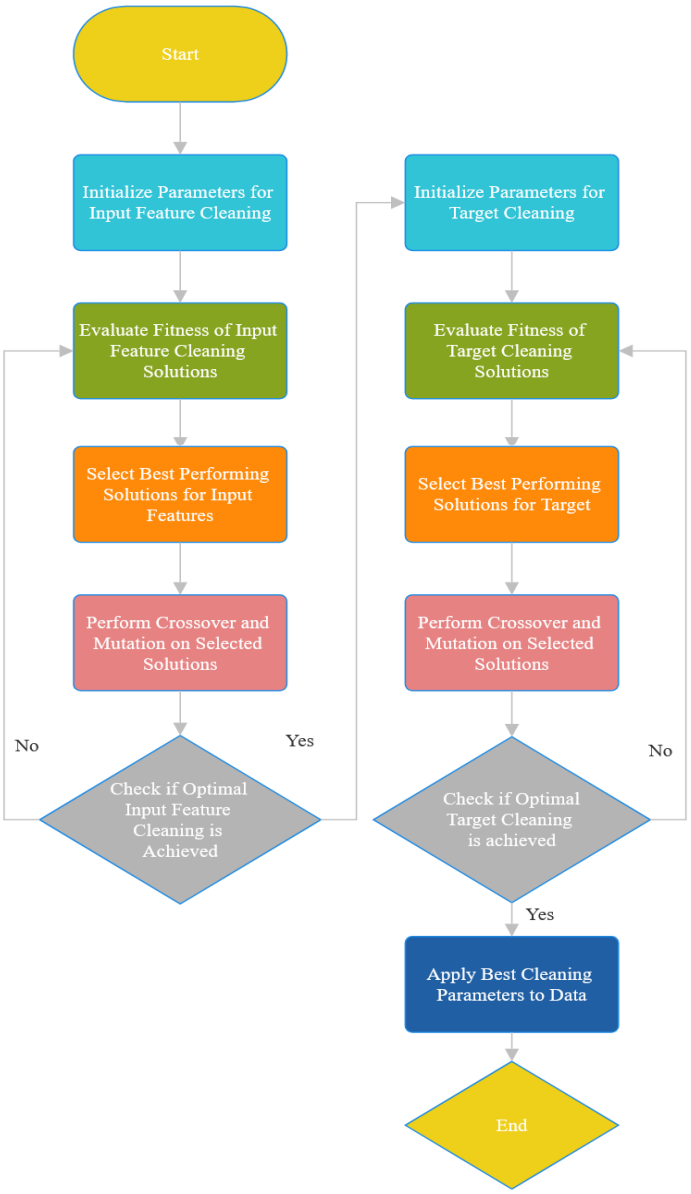


Figure 4.9: A flowchart that displays the complete cleaning optimization process.

4.5.2.3 Results of Hybrid Optimization Algorithm for Data Cleaning

MATLAB was used for all optimization runs during this study. The results here are from 40 independent runs, each consisting of 200 iterations. Forty search agents have been used in the optimization process. These runs were designed to find the most effective data-cleaning parameters using the hybrid optimization technique, as indicated by the objective

function described in previous subsections. The following two subsections discuss the results of preprocessing the entire 2021 ERA5 dataset and the individual data groups.

Data Cleaning Results of the Full-Year Dataset The effectiveness of the hybrid optimization technique in improving the entire 2021 ERA5 data is summarized in Table 4.4, which shows the correlation coefficients between the ML input features (Pressure, Temperature, Relative Humidity, and Specific Humidity) and the target output, CLWC, before and after cleaning. Notably, the 'Old' column represents the original correlations before cleaning, whereas the 'new' column represents the correlations after applying the optimized cleaning parameters. The fourth column explains the optimal parameters selected by the algorithm for data cleaning, presented in the format: [fillMethod, findMethod, smoothMethod, windowLengthForSmooth, percentileThresholdLower, percentileThresholdUpper, smoothingFactor]. Each entry consists of two sets of parameters: the first for cleaning the input features and the second for cleaning the target output.

Applying the hybrid optimization technique to the 2021 dataset resulted in improved data quality, as indicated by the increase in the correlation coefficients between the input features and the target variable (CLWC). For example, the correlation coefficient for pressure with CLWC increased from 0.21106 to 0.67546. Similar improvements were observed in temperature, relative humidity, and specific humidity after cleaning. These changes suggest that the optimization method effectively addressed data abnormalities, leading to a more accurate representation of the variables.

Data Cleaning and Preprocessing Results for Individual Groups The data preprocessing is then applied to the grouped datasets. Table 4.5 highlights the improvements in correlation coefficients between various input features and the target for all five dataset groups. Specifically, the optimization algorithm enhanced correlations across variables for

Group 1, with temperature increasing to 0.63219, specific humidity rising to 0.62938, and relative humidity increasing to 0.76979. Temperature and humidity correlations increased significantly in Group 2, with relative humidity achieving the maximum correlation of 0.79723. The results of Group 3 have been identified by a significant increase in pressure correlation to 0.81672. Group 4 improved effectively across variables, with relative humidity achieving a correlation of 0.74477. Group 5 continued the same significant enhancements, especially for relative humidity, which reached a correlation of 0.79155.

4.5.3 ML Algorithms and Methods Used in This Study

In this chapter, a diverse array of machine learning models was utilized to predict CLWC, including linear regression, regression decision trees, ensemble trees, and neural networks, each tailored to different complexities of the dataset. One key aspect of this study is the thorough evaluation of different machine learning models, where the ensemble trees, particularly Random Forests and Gradient Boosting, demonstrated the best performance. These models were chosen for their ability to manage high-dimensional atmospheric data and capture complex, non-linear relationships between features. Ensemble trees excelled at reducing overfitting and generalizing across diverse weather conditions, making them particularly well-suited for CLWC prediction. In contrast, neural networks did not perform as well as expected in this study. Despite their flexibility and capacity to model non-linear relationships, the neural network models struggled with overfitting and failed to generalize the complex atmospheric data used in this analysis effectively. This was likely due to the limited ability of the neural network to handle the high variability in the training data and the absence of sufficient data points to leverage the deep learning model's potential fully. Thus, the decision tree-based models, especially ensemble trees, emerged as this study's most reliable and effective approach for CLWC prediction. The combination of interpretability, robustness, and generalization ability made them the optimal choice for handling the intricacies of cloud microphysics and atmospheric processes, areas where traditional physical models have

struggled. These models are similar to those used in Chapter 3. The specific models and their hyperparameters can be found in Table 3.1.

4.6 Results and Discussion

This section describes the results of using various machine learning algorithms to predict Cloud Liquid Water Content profiles. The study utilizes meteorological data from Hong Kong for the year 2021. This dataset (containing 2,592,950 samples) originated from ERA5 data gathered across eight specific locations in Hong Kong and is analyzed at 37 atmospheric pressure levels throughout the year. It was then organized into one comprehensive dataset for the entire year, as well as five distinct groups based on specific meteorological characteristics described earlier. For the analysis, the data were partitioned into two distinct subsets: 85% was employed for a detailed cross-validation process, utilizing a k=5 fold method for thorough training and validation, while the remaining 15% was reserved for testing the models' performance.

To mitigate overfitting, the K-fold cross-validation technique was implemented. The former divides the data into five subsets, each used once as validation data, offering a reliable performance estimate on unseen data. This technique effectively balances bias and variance, limiting overfitting and ensuring accurate model performance evaluation. The first subsection of this analysis focuses on the outcomes of applying these algorithms to the full-year dataset. The second subsection shifts to analyzing the grouped data, adhering to the previously described categorization process. These analyses were conducted on a desktop computer equipped with an i7-2600K CPU @ 3.40 GHz and 24 GB of RAM.

4.6.1 Analysis of Results Based on the Full-Year Data

Table 4.6 shows detailed results of predicting CLWC based on the full-year data. The Fine Tree was the most accurate decision tree model, with the lowest RMSE and MSE, indicating

minimum prediction errors, and it has a high R-squared value, meaning a good fit to the data. The Medium Tree had slightly lower correlation coefficients than the Coarse Tree, with higher error metrics and a worse data fit.

In terms of ensemble models, the Bagged Tree outperformed the Boosted Tree, which had more prediction errors and a poorer data fit. However, the neural network models (NarrowNN, MediumNN, WideNN, BiLayeredNN, and TriLayeredNN) struggled significantly, evidenced by high RMSE and MSE values and negative R-squared scores. These outcomes suggest that while preventing the vanishing gradient problem, the ReLU activation function could not prevent "dead neurons" (inactive neurons causing zero gradients during backpropagation). This problem was made worse by the simple architecture of these networks, which was likely insufficient to capture the complex nonlinear dynamics of CLWC data, as well as inadequate complexity or improper initialization. Meanwhile, while linear regression has results that are more accurate than those of neural networks, it trails behind tree-based models.

These findings are consistent with the findings from the testing dataset, as reported in Table 4.7. Decision tree models (the Fine, Medium, and Coarse Trees) performed consistently and accurately with low MAE, MSE, and RMSE. The Fine and Medium Trees had similar R-squared and correlation values. Despite its effectiveness, the Coarse Tree had higher error metric values and a lower R-squared value.

The Bagged Tree succeeded again in the ensemble category, with the lowest error values and the most outstanding R-squared value, confirming its superior predictive accuracy and fit. The Boosted Tree, on the other hand, had larger error rates and a substantially lower R-squared value. In the testing dataset, neural network models continued showing limitations, with much larger error values and severely negative R-squared scores. Linear Regression performed well, exceeding neural networks but not matching the tree-based and bagged models.

Table 4.6: Performance results of Various Machine Learning Models for Predicting Cloud Liquid Water Content (CLWC) in 2021 Training Dataset.

Model Type	RMSE (kg/m^3) (Valid)	MSE ($((kg/m^3)^2)$) (Valid)	R^2 (Valid)	MAE (kg/m^3) (Valid)	Corr (Valid)
Fine Tree	1.41E-07	1.98E-14	0.85895	7.56E-08	0.9736
Medium Tree	1.39E-07	1.92E-14	0.86302	7.82E-08	0.9579
Coarse Tree	1.43E-07	2.03E-14	0.85525	8.34E-08	0.9392
Boosted Tree	1.71E-07	2.93E-14	0.79147	1.13E-07	0.8926
Bagged Tree	1.25E-07	1.57E-14	0.8880	7.42E-08	0.9678
NNN	0.000937	8.78E-07	-6260004	0.0006	0.2017
MNN	0.000995	9.91E-07	-7061305	0.0006	0.0104
WNN	0.000999	9.97E-07	-7109489	0.0006	0.0112
BiNN	0.000909	8.27E-07	-5892127	0.0004	0.0267
TriNN	0.000916	8.39E-07	-5982461	0.0005	0.0198
LR	2.16E-07	4.66E-14	0.667978	1.60E-07	0.8173

Table 4.7: Performance results of various machine learning models for predicting cloud liquid water content (CLWC) in 2021 testing dataset.

Model Type	MAE (kg/m^3) (Test)	MSE $(kg/m^3)^2$ (Test)	RMSE (kg/m^3) (Test)	R^2 (Test)	Corr (Test)
Fine Tree	7.26E-08	1.84E-14	1.36E-07	0.8692	0.9331
Medium Tree	7.58E-08	1.82E-14	1.35E-07	0.8704	0.9332
Coarse Tree	8.17E-08	1.96E-14	1.40E-07	0.8608	0.9279
Boosted Tree	1.14E-07	2.93E-14	1.71E-07	0.7913	0.8925
Bagged Tree	7.26E-08	1.50E-14	1.23E-07	0.8932	0.9452
NNN	0.00067	9.37E-07	0.00097	-6.665E6	0.2016
MNN	0.00064	9.51E-07	0.00098	-6.762E6	0.0114
WNN	0.00062	1.00E-06	0.001	-7.117E6	0.0127
BiNN	0.00039	9.84E-07	0.00099	-6.996E6	0.0273
TriNN	0.00064	9.84E-07	0.00099	-6.998E6	0.0227
LR	1.60E-07	4.67E-14	2.16E-07	0.6675	0.817

The bar plot in Figure 4.10 serves as an extended illustration of the Tables 4.6 and 4.7, capturing the comparative performance of the various machine learning algorithms across the entire 2021 year data. It visually compares the RMSE, MSE, and MAE across training and testing datasets, highlighting the better performance of decision tree models, especially Bagged Trees. In this figure, the decibel (dB) scale is used to represent error metrics such as RMSE, MSE, and MAE. Lower dB values indicate reduced error and, consequently, better model performance.

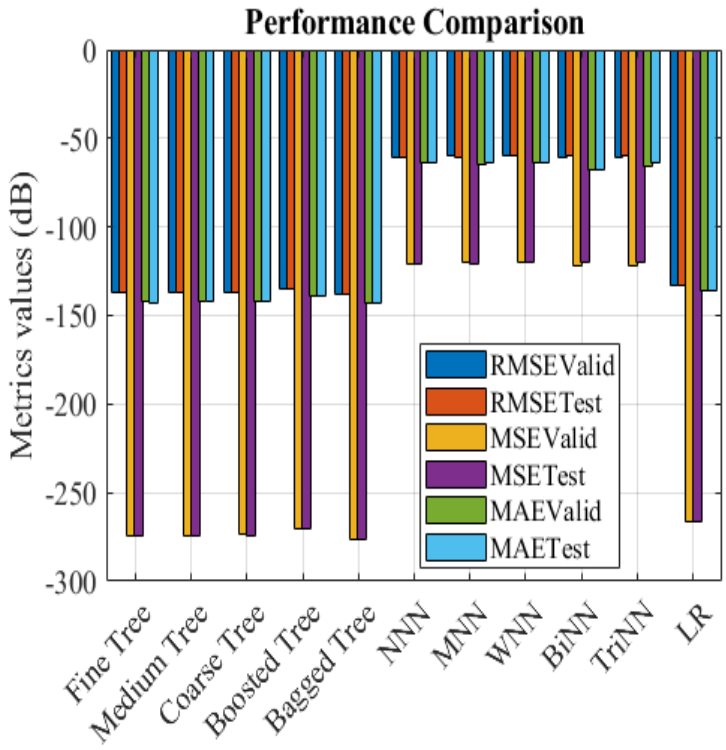


Figure 4.10: Comparative performance of various machine learning models on training and testing datasets for 2021, as measured by RMSE, MSE, and MAE metrics, presented on a logarithmic decibel (dB) scale. Lower dB values indicate smaller errors and better model performance.

Figures 4.11, 4.12, 4.13 and 4.14 present scatter and 2D histogram plots that provide a comprehensive visual narrative of multiple machine learning models' effectiveness in predicting CLWC across training and testing datasets.

The Fine Tree and Bagged Tree models result in good accuracy in the training dataset analysis, as shown in Figure 4.11. Their scatter plots are characterized by densely packed dots along the diagonal, suggesting that most predictions closely match the actual measurements. The Bagged Tree, in particular, has a tighter cluster of data points, suggesting its somewhat higher accuracy. The Medium Neural Network (MNN) and Linear Regression (LR) models have greater dispersion in their plots. The scatter plot of the MNN displays points that deviate significantly from the diagonal, especially at the extremities of the response axis,

demonstrating its negative R-squared values. Meanwhile, the LR model's figure shows a dense concentration at lower values that progressively disperses at higher levels.

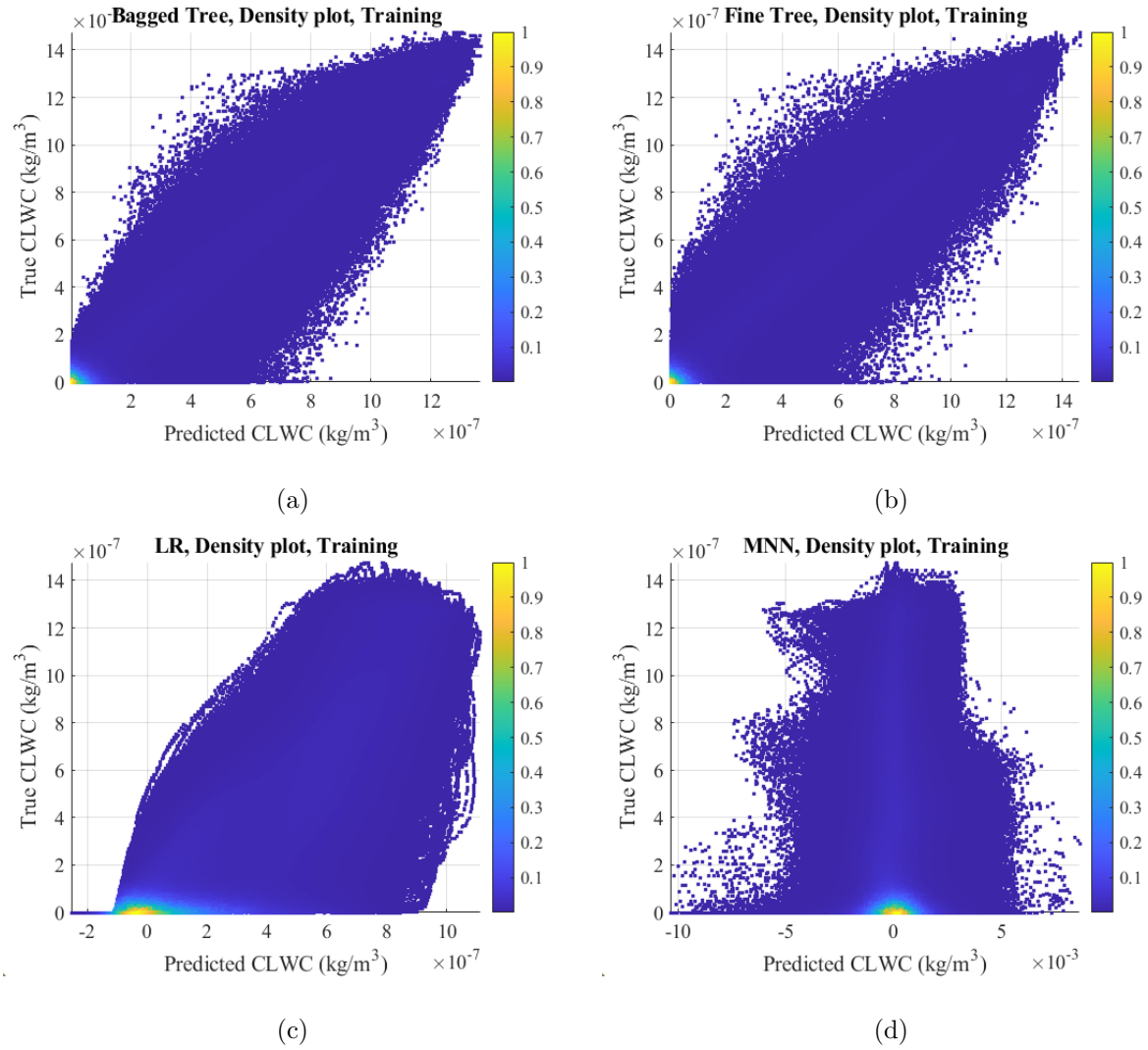


Figure 4.11: Scatter density plots with color maps of the expected vs. actual CLWC of the 2021 year training dataset using different ML algorithms; (a) Bagged tree, (b) Fine tree, (c) Linear regression, and (d) Medium neural network.

Figure 4.12 re-configures this visual analysis by displaying 2D histogram plots for the full-year training dataset. The plot of the Bagged Tree model shows a diagonal concentration of high-density values, particularly at lower response ranges, indicating its low-error, high-correlation profile estimations. The Fine Tree plot, on the other hand, while still showing

high diagonal density, has a slightly wider color spread. The MNN plot deviates significantly from the predicted diagonal pattern, illustrating the model's relatively poor performance.

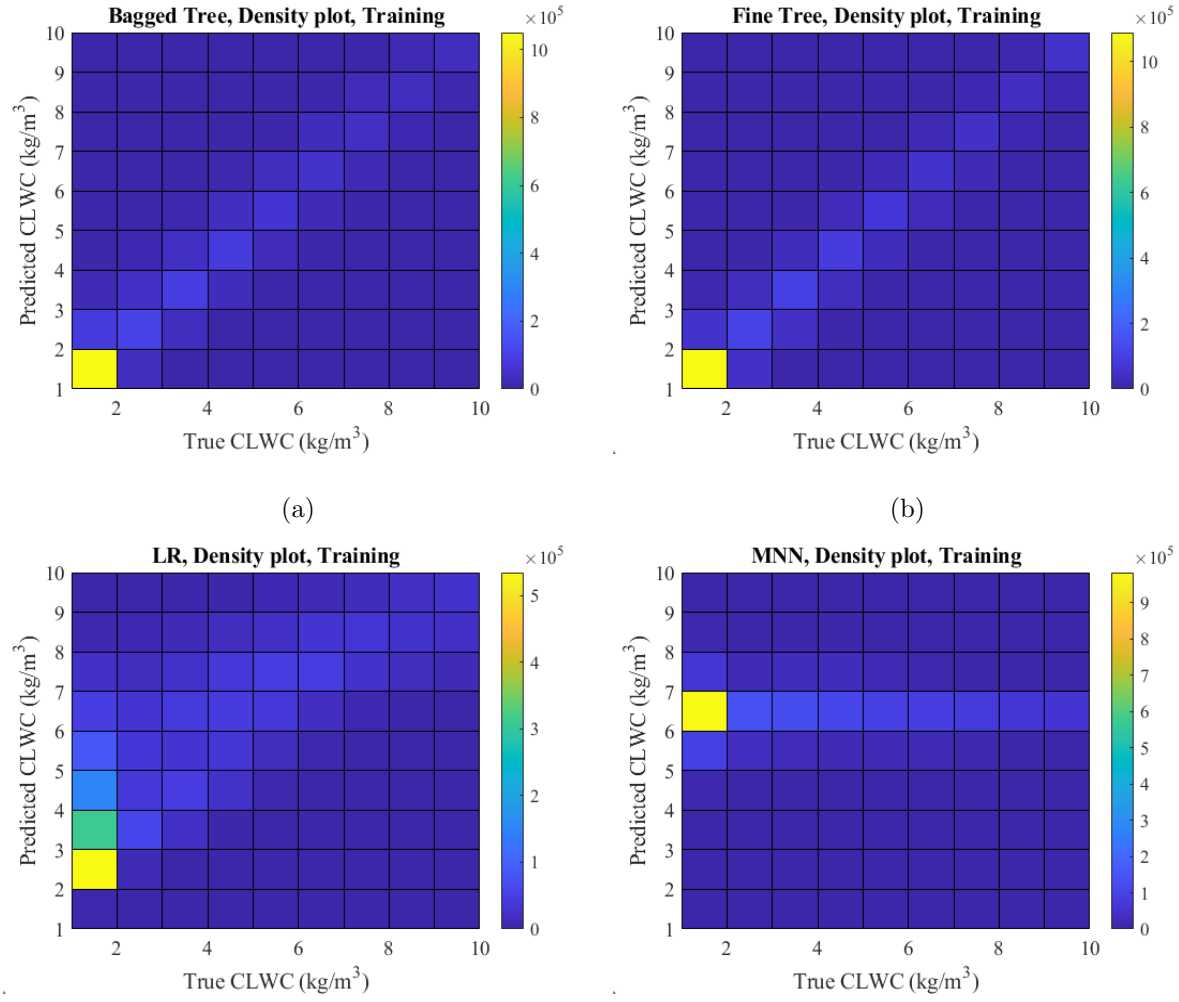


Figure 4.12: 2D histogram plots with color maps of expected vs. actual CLWC based on the 2021 year training dataset using different ML algorithms; (a) Bagged tree, (b) Fine tree, (c) Linear regression, and (d) Medium neural network. The color map indicates the frequency of data points, with different colors showing varying densities.

The scatter density plots for the testing dataset, as shown in Figure 4.13, present the tested and verified performances of the ML algorithms. The Bagged Tree model provides accurate predictions that closely match actual responses, producing a dense cluster along the

center line, which supports the high R-squared and Correlation Coefficient values. Although the Fine Tree model has a slightly more distributed scatter of points, it still has good prediction accuracy. In contrast, the MNN model's plot for the testing dataset deviates significantly from the predicted diagonal, with predictions dispersed over the plot, showing difficulty in generating correct predictions and potential overfitting or structural problems. The plot of the LR model indicates a band of modest density at the lower end of the predicted values, gradually decreasing with larger values, confirming the model's limits in capturing the entire complexity of the dataset.

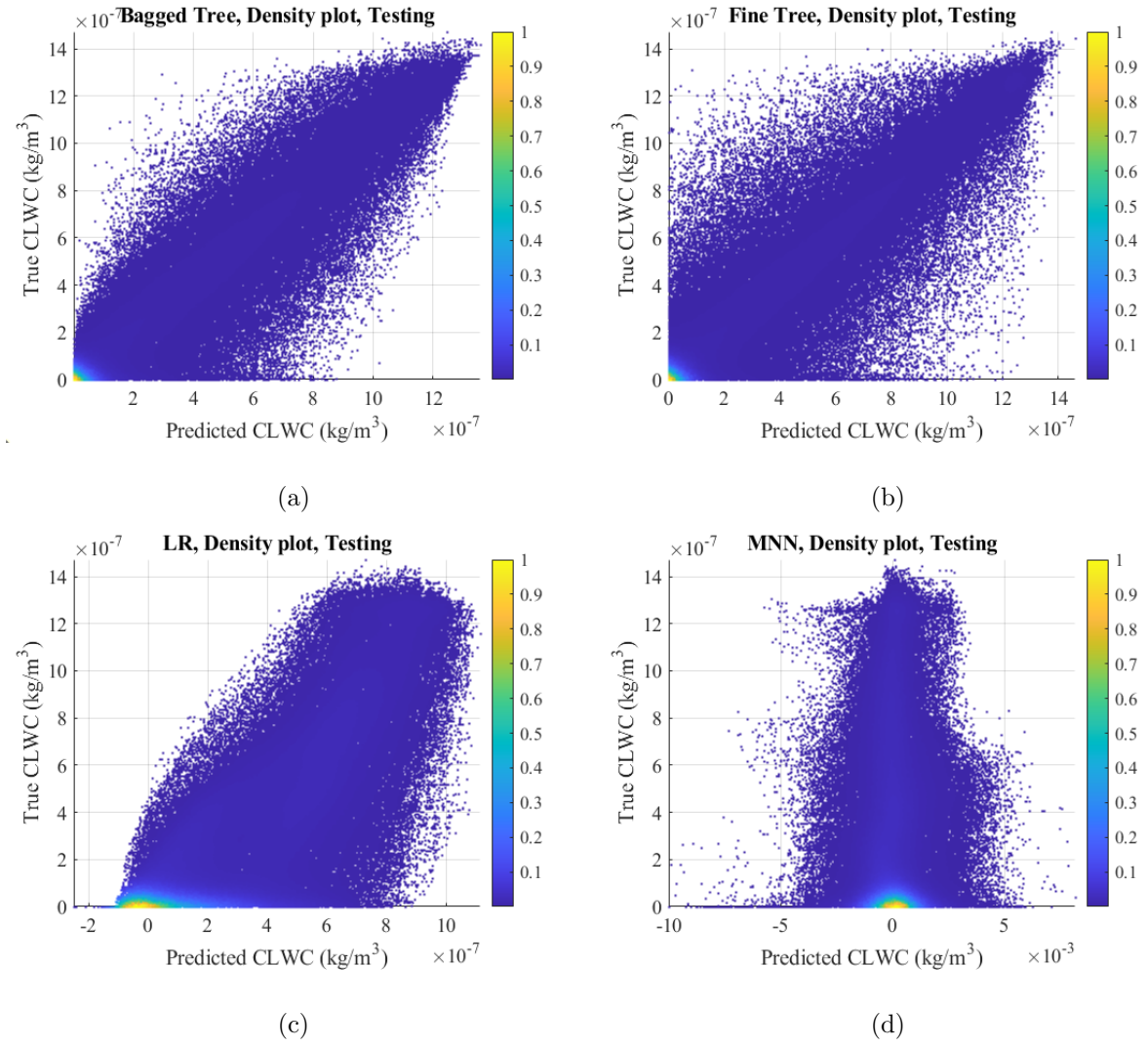


Figure 4.13: Scatter density plots with color maps of expected vs. actual CLWC based on the 2021 year testing dataset using different ML algorithms; (a) Bagged tree, (b) Fine tree, (c) Linear Regression (LR), and (d) Medium Neural Network (MNN) .

The 2D histogram plots in Figure 4.14 for the testing dataset reflect the same patterns observed during the training phase. The Fine Tree and Bagged Tree models maintain high-density values at decreasing response levels. The MNN model, on the other hand, continues to have fitting issues, with its plot displaying significant deviations from the diagonal even during testing. The LR model's 2D histogram plot shows a density gradient comparable to the training phase data.

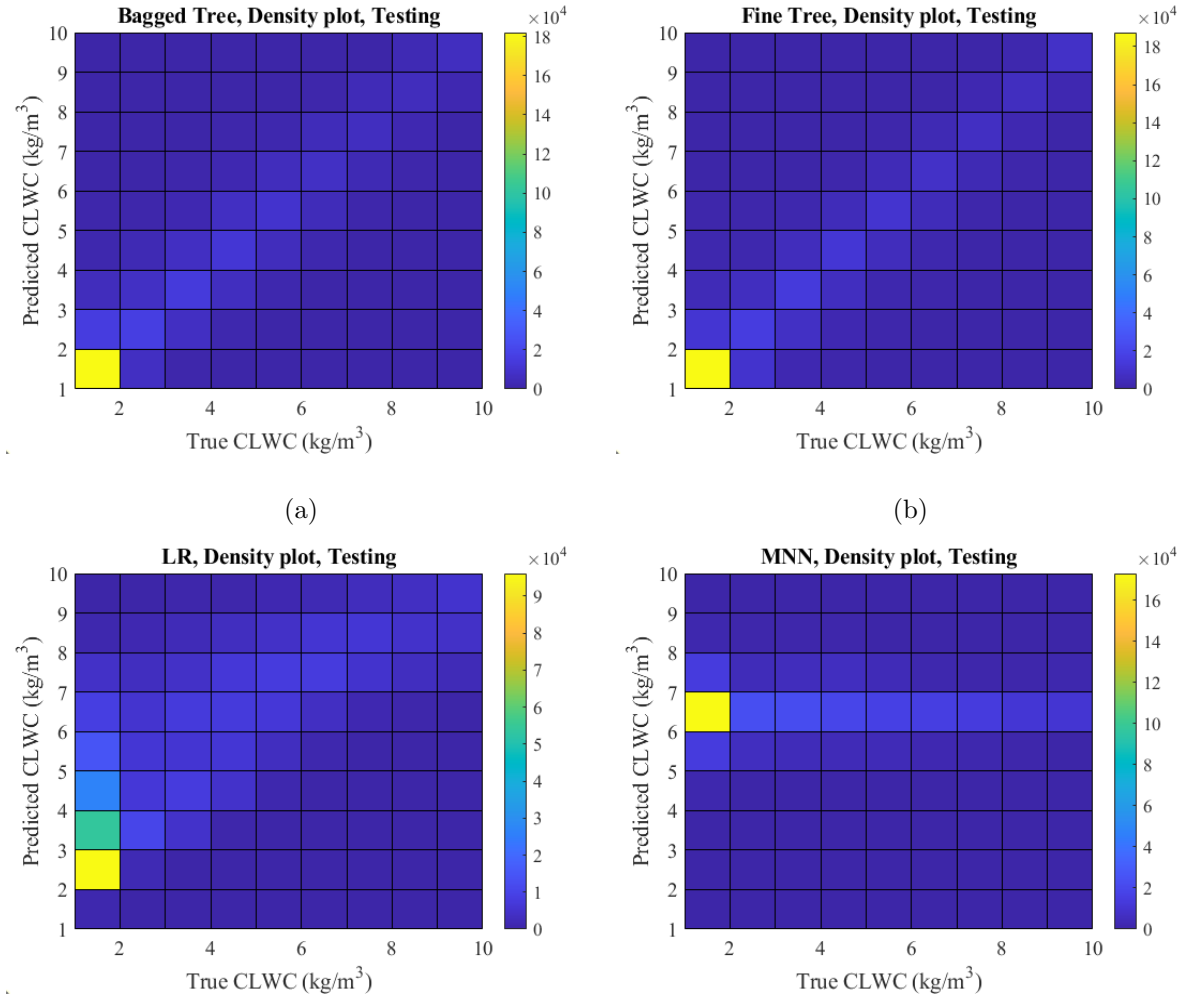


Figure 4.14: 2D histogram plots with color maps of expected vs. actual CLWC based on the 2021 year testing dataset using different ML algorithms; (a) Bagged tree, (b) Fine tree, (c) Linear regression, and (d) Medium neural network. The color map indicates the frequency of data points, with different colors showing varying densities.

The group of profiles presented in Figure 4.15 offers a comparative visualization of CLWC predictions from three distinct machine learning models (Bagged Trees, Fine Tree, and Linear Regression) against actual measured values from Radiosondes across varying pressure levels. These profiles are selected randomly at various times and under different environmental

conditions. The y-axis in each subfigure has been reversed to represent the actual altitudes accurately.

These sample profiles demonstrate that the Bagged Trees and Fine Tree models closely follow the actual CLWC values across the pressure levels. This consistency is particularly evident in profiles like a, b, d, and e, where the correlation coefficients are very high, suggesting high prediction accuracy. In contrast, the Linear Regression model, while generally matching the trend of actual CLWC values, tends to diverge at specific pressure levels. For instance, profiles b, e, and g show scenarios where Linear Regression predictions differ significantly from the real observations at nearly all pressure levels, resulting in the lowest correlation coefficients. This divergence highlights the model's limitations in adapting to the non-linear nature of atmospheric data, as linear techniques may not adequately capture the dynamics of cloud formation.

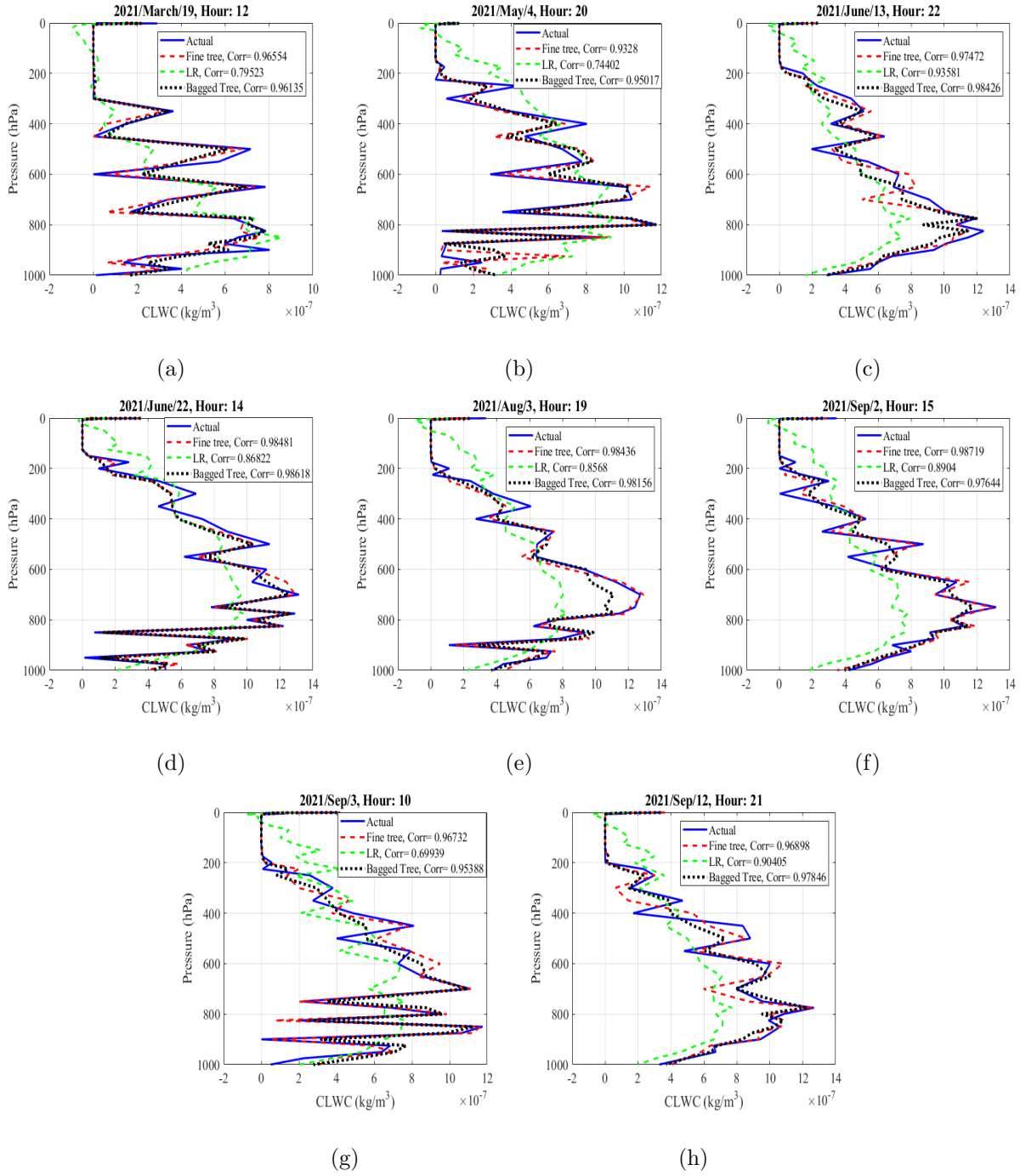


Figure 4.15: Vertical profiles of actual vs predicted CLWC values using Fine tree, linear regression, and Bagged tree.

Figure 4.16 shows Partial Dependency Relationship (PDR) plots generated by the Bagged Tree model for both training and testing datasets and provides a deeper understanding of how

different meteorological factors impact the prediction of the CLWC. These subfigures depict a visual link between the input features (pressure (kPa), temperature (k), relative humidity (RH) (%), and specific humidity (SH) (kg/kg)) and the predicted CLWC, demonstrating the interpretability and prediction capability of the Bagged Tree model.

The pressure dependency plot correlates with expected atmospheric conditions: as pressure decreases with altitude, predicted CLWC generally increases. This trend reflects the higher moisture capacity of warmer air at lower altitudes. However, between 200 to 700 hPa, CLWC first increases then decreases as altitude decreases, indicating the nuanced impact of temperature changes and condensation at different atmospheric layers. The temperature correlation exhibits a sharp CLWC increase as temperatures warm from 240 K to 250 K, aligning with the air's enhanced moisture capacity. However, between 250 K and 270 K, this upward trend in CLWC moderates, signaling a nearing saturation point. Beyond 270 K, a slight decrease in CLWC could indicate the beginnings of condensation, where the air's capacity for water vapor declines.

Similarly, the relationship with relative humidity is marked by a steady CLWC ascent from low RH levels, with a significant increase from 60% RH to full saturation, reflecting the exponential increase in moisture content as the air becomes fully saturated. The most notable increase in CLWC is observed as RH nears 100%, the point at which the air holds the maximum water vapor before excess moisture condenses.

Finally, the plot against specific humidity underscores an initial sharp CLWC rise with increasing SH up to 0.003 kg/kg, followed by a gradual decrease, where moisture addition is in equilibrium with the air's holding capacity.

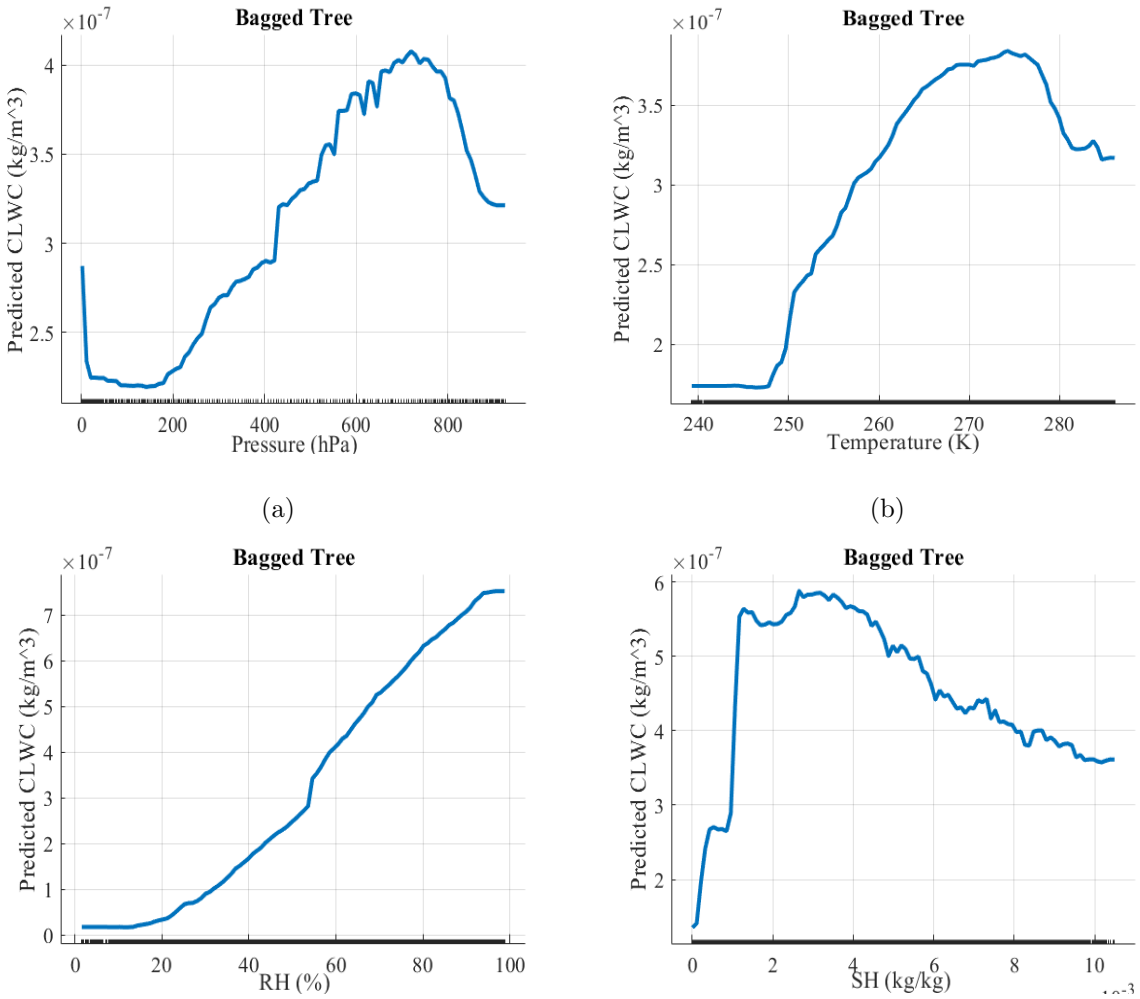


Figure 4.16: Partial Dependency Relationship (PDR) plots between predicted CLWC and (a) pressure (kPa), (b) temperature (K), (c) relative humidity (RH) (%), and (d) specific humidity (SH) (kg/kg) of the 2021 dataset.

4.6.2 Performance Evaluations Based on Grouped Datasets

The results from the grouped training dataset, as listed in Table 4.8, indicate that the Bagged Tree model has the best performance for the Group 1 dataset. It achieves the lowest RMSE (4.5517E-10 kg/m^3) and MSE (2.0718E-19 (kg/m^3)^2), a high R-squared (0.9214) as well as a strong Correlation Coefficient (0.9602). The Fine Tree and Medium Tree outperform

the Coarse Tree regarding lower RMSE and MSE. The WNN model, on the other hand, has a high RMSE and a negative R-squared, indicating a poor fit to the training data. The Bagged Tree model also performs better in the other groups, with the lowest RMSE and MSE and the greatest R-squared and Correlation Coefficient. While significantly less exact, the Fine Tree and Medium Tree still show good R-squared values. The WNN model continually trails, with a poor predicting ability. Similar results are observed when focusing on the testing dataset in Table 4.9. In Group 1, the Bagged Tree model has the lowest RMSE (4.4232E-10 kg/m^3) and MSE (1.9565E-19 (kg/m^3)^2), as well as a high R-squared (0.9257) and Correlation Coefficient (0.9624). The Fine Tree and Medium Tree models also perform well but with slightly higher RMSE and MSE values. The WNN model, on the other hand, suggests significant errors due to its very high RMSE and MSE and negative R-squared. In Groups 2 and beyond, the Bagged Tree model has the lowest RMSE and MSE and the greatest R-squared and Correlation Coefficient.

Both tables confirm the Bagged Tree model's better accuracy and consistency in predicting CLWC profiles across all groups. The Fine Tree and Medium Tree also demonstrate effective predictive capabilities, while the WNN model constantly underperforms, indicating its unsuitability for this specific task.

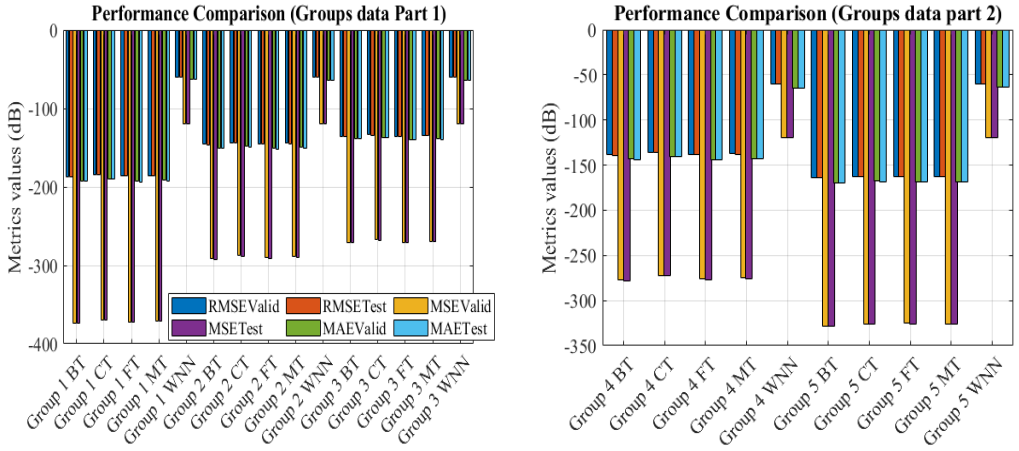
Table 4.8: Performance results of machine learning models across different groups in the 2021 training dataset for CLWC prediction.

Model Type	RMSE (kg/m^3) (Valid)	MSE $(kg/m^3)^2$ (Valid)	R^2 (Valid)	MAE (kg/m^3) (Valid)	Corr (Valid)
Group 1 Bagged Tree	4.5517E-10	2.0718E-19	0.9214	2.4597E-10	0.9602
Group 1 Coarse Tree	5.8547E-10	3.4278E-19	0.8699	3.1239E-10	0.9328
Group 1 Fine Tree	5.0861E-10	2.5869E-19	0.9018	2.3183E-10	0.9502
Group 1 MTree	5.2819E-10	2.7898E-19	0.8941	2.6449E-10	0.9458
Group 1 WNN	0.0010	1.0007E-06	-3.8E+11	0.0007	-0.0061
Group 2 Bagged Tree	5.0486E-08	2.5488E-15	0.9431	2.9379E-08	0.9713
Group 2 Coarse Tree	6.5912E-08	4.3443E-15	0.903	3.7781E-08	0.9503
Group 2 Fine Tree	5.6928E-08	3.2408E-15	0.9276	2.8926E-08	0.9634
Group 2 MTree	5.9069E-08	3.4892E-15	0.9221	3.2124E-08	0.9603
Group 2 WNN	0.0010	1.0022E-06	-2.2E+07	0.0006	0.0011
Group 3 Bagged Tree	1.7075E-07	2.9154E-14	0.9143	1.1475E-07	0.9564
Group 3 Coarse Tree	2.0711E-07	4.2896E-14	0.8739	1.3579E-07	0.9348
Group 3 Fine Tree	1.7442E-07	3.0422E-14	0.9105	1.0308E-07	0.9546
Group 3 MTree	1.8390E-07	3.3820E-14	0.9005	1.1515E-07	0.9491
Group 3 WNN	0.0010	9.9355E-07	-2.92E+5	0.0006	-0.0227
Group 4 Bagged Tree	1.1477E-07	1.3173E-14	0.9468	6.8201E-08	0.9733
Group 4 Coarse Tree	1.5916E-07	2.5333E-14	0.8977	9.1938E-08	0.9475
Group 4 Fine Tree	1.2584E-07	1.5837E-14	0.936	6.4137E-08	0.9677
Group 4 MTree	1.3629E-07	1.8575E-14	0.925	7.4436E-08	0.9618
Group 4 WNN	0.0010	1.0086E-06	-4.07E+5	0.0006	0.0093
Group 5 Bagged Tree	6.2127E-09	3.8598E-17	0.8962	3.4750E-09	0.9467
Group 5 Coarse Tree	7.1115E-09	5.0573E-17	0.864	3.9856E-09	0.9296
Group 5 Fine Tree	7.3600E-09	5.4169E-17	0.8543	3.7778E-09	0.9257
Group 5 MTree	7.0198E-09	4.9277E-17	0.8674	3.7834E-09	0.9317
Group 5 WNN	0.0010	1.0161E-06	-2.7E+09	0.0007	0.0222

Table 4.9: Performance results of machine learning models across different groups in the 2021 testing dataset for CLWC prediction.

Model Type	RMSE (kg/m^3) (Test)	MSE $(kg/m^3)^2$ (Test)	R^2 (Test)	MAE (kg/m^3) (Test)	Corr (Test)
Group 1 BaggedTree	4.4232E-10	1.9565E-19	0.9257	2.3492E-10	0.9624
Group 1 CoarseTree	5.7609E-10	3.3188E-19	0.8739	3.0211E-10	0.9349
Group 1 FineTree	4.8980E-10	2.3991E-19	0.9089	2.1636E-10	0.9538
Group 1 MTree	5.1583E-10	2.6608E-19	0.8989	2.5191E-10	0.9483
Group 1 WNN	0.0010	1.0331E-06	-3.9E+11	0.0007	-0.0107
Group 2 BaggedTree	4.7743E-08	2.2794E-15	0.9493	2.7651E-08	0.9745
Group 2 CoarseTree	6.3104E-08	3.9821E-15	0.9113	3.5952E-08	0.9547
Group 2 FineTree	5.3840E-08	2.8988E-15	0.9355	2.6836E-08	0.9674
Group 2 MTree	5.5793E-08	3.1128E-15	0.9307	2.9933E-08	0.9648
Group 2 WNN	0.0010	1.0192E-06	-2.3E+07	0.0006	0.0212
Group 3 BaggedTree	1.6658E-07	2.7749E-14	0.9188	1.1128E-07	0.9588
Group 3 CoarseTree	2.0082E-07	4.0330E-14	0.8819	1.3069E-07	0.9391
Group 3 FineTree	1.6676E-07	2.7808E-14	0.9186	9.8072E-08	0.9587
Group 3 MTree	1.7551E-07	3.0804E-14	0.9098	1.0907E-07	0.9539
Group 3 WNN	0.0010	9.7717E-07	-2.86E6	0.0006	-0.0011
Group 4 BaggedTree	1.1017E-07	1.2138E-14	0.9509	6.5126E-08	0.9754
Group 4 CoarseTree	1.5279E-07	2.3343E-14	0.9057	8.8267E-08	0.9517
Group 4 FineTree	1.2018E-07	1.4443E-14	0.9416	6.1437E-08	0.9706
Group 4 MTree	1.2948E-07	1.6765E-14	0.9323	7.0355E-08	0.9656
Group 4 WNN	0.0010	1.0013E-06	-4.05E6	0.0006	0.0028
Group 5 BaggedTree	6.1384E-09	3.7680E-17	0.8988	3.4144E-09	0.9481
Group 5 CoarseTree	7.0506E-09	4.9711E-17	0.8665	3.9368E-09	0.9309
Group 5 FineTree	7.2182E-09	5.2102E-17	0.8601	3.6885E-09	0.9286
Group 5 MTree	6.8916E-09	4.7494E-17	0.8724	3.7042E-09	0.9343
Group 5 WNN	0.0010	1.0099E-06	-2.7E+09	0.0007	-0.0277

The bar plots in Figure 4.17 summarize the comparative study of the performance measures for the machine learning models, as discussed before in the discussion on grouped datasets. The visual representation of each statistic through different colored bars emphasizes the Bagged Tree model constantly presents the lowest RMSE and MSE values, aligning with the findings based on Tables 4.8 and 4.9.



(a) (b)

Figure 4.17: Performance metrics comparison across machine learning models for grouped data, shown for validation and test datasets. Metrics are presented on a logarithmic decibel (dB) scale, where lower dB values signify reduced errors and improved model performance.

Next, Figures 4.18 and 4.19 show the scatter density plots for Groups 1 to 5 in both the training and testing phases, respectively. The Fine Tree and Bagged Tree models in Group 1 align well with the actual CLWC values throughout the training period. This alignment is most noticeable in the Bagged Tree model, especially at lower CLWC values. In contrast, the WNN model has a widely dispersed distribution of predictions that deviates significantly from the diagonal. In Group 2, the Bagged Tree model maintains better predictive symmetry and clear clusters along the diagonal, strengthening its reliability and consistency. The Fine Tree model, while still closely matching the actual data, has a broader range. The WNN model, continuing the pattern from Group 1, shows extensive dispersion. These observed behaviors are consistent in the remaining groups and the testing phase results.

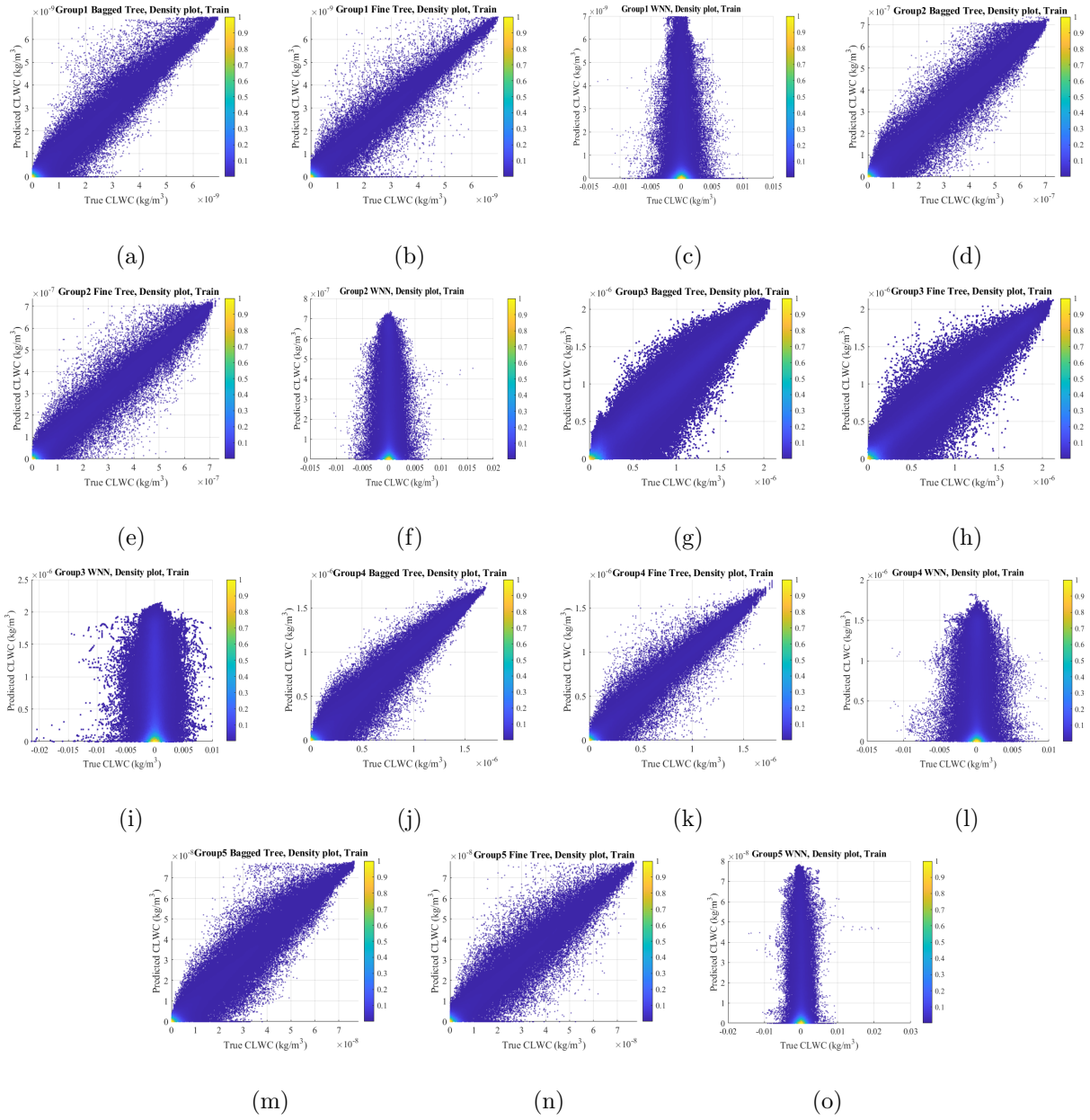


Figure 4.18: Scatter density plots with color maps within of expected vs. actual CLWC of the groups' training dataset using different ML algorithms; (a) Group 1, Bagged tree, (b) Group 1, Fine tree, (c) Group 1, Wide Neural network, (d) Group 2, Bagged tree, (e) Group 2, Fine tree, (f) Group 2, Wide Neural network, (g) Group 3, Bagged tree, (h) Group 3, Fine tree, (i) Group 3, Wide Neural network, (j) Group 4, Bagged tree, (k) Group 4, Fine tree, (l) Group 4, Wide Neural network, (m) Group 5, Bagged tree, (n) Group 5, Fine tree, (o) Group 5, Wide Neural network.

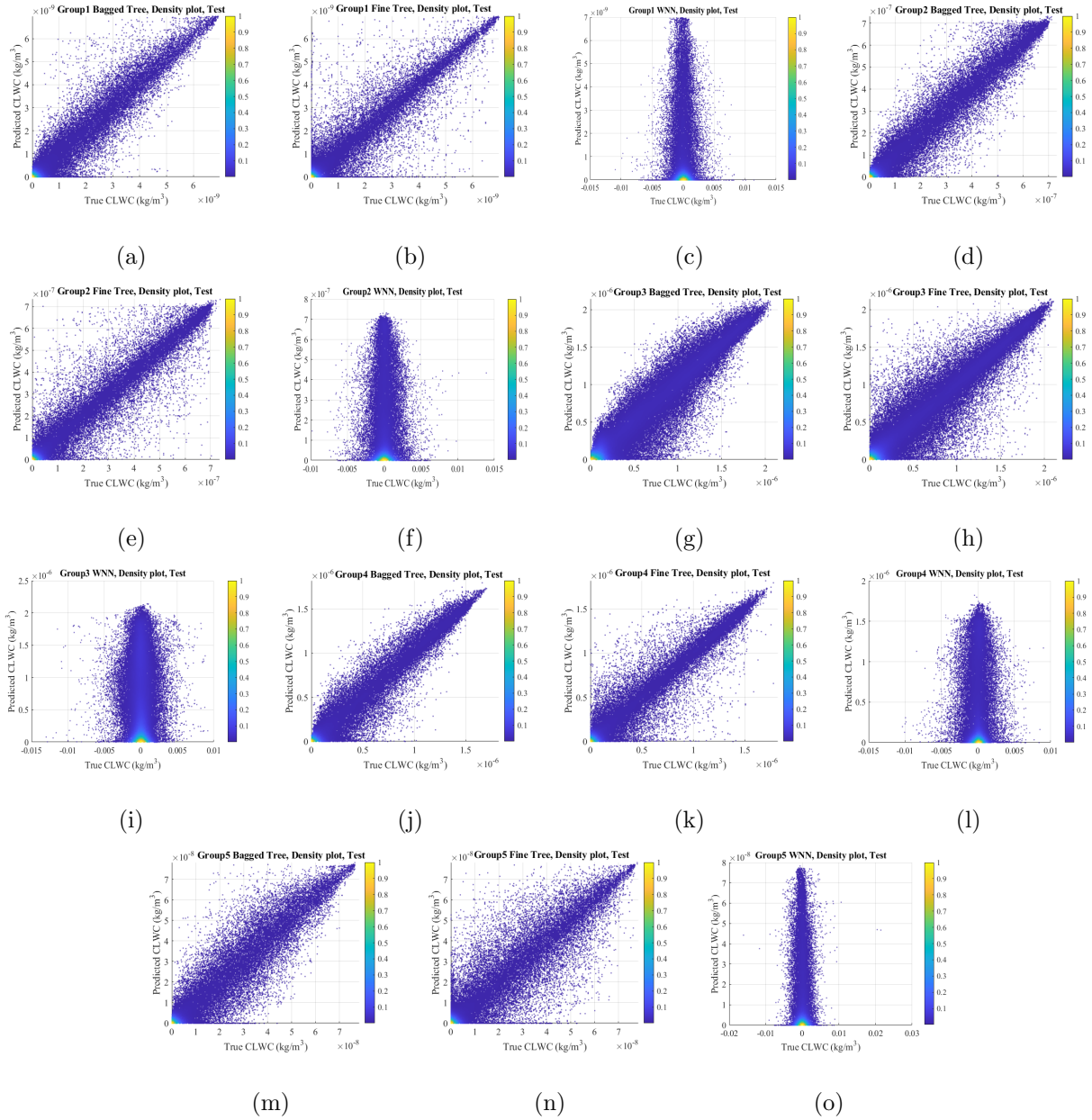


Figure 4.19: Scatter density plots with color maps within of expected vs. actual CLWC of the groups' testing dataset using different ML algorithms; (a) Group 1, Bagged tree, (b) Group 1, Fine tree, (c) Group 1, Wide Neural network, (d) Group 2, Bagged tree, (e) Group 2, Fine tree, (f) Group 2, Wide Neural network, (g) Group 3, Bagged tree, (h) Group 3, Fine tree, (i) Group 3, Wide Neural network, (j) Group 4, Bagged tree, (k) Group 4, Fine tree, (l) Group 4, Wide Neural network, (m) Group 5, Bagged tree, (n) Group 5, Fine tree, (o) Group 5, Wide Neural network.

Furthermore, in examining the CLWC predictions, Figures 4.20 and 4.21 display the 2D histogram plots for the Fine Tree, Bagged Tree, and WNN models. Figure 4.20 illustrates the data from the training dataset, while Figure 4.21 focuses on the testing dataset data.

During the training phase, Groups 1–3 illustrate the Fine Tree and Bagged Tree models' ability to predict CLWC values, as evidenced by the intense colors around the middle diagonal line. The Bagged Tree model, in particular, presents a constant and uniform distribution over the CLWC range, implying a reliable performance regardless of climatic variability. This robust predictive capability is reinforced in Groups 4 and 5, where the Bagged Tree model maintains its broad and uniform distribution pattern. The WNN model, on the other hand, continues to deviate significantly from expected predictive behaviors in each group. Its density plots consistently show a clustering of predictions away from the diagonal line, proving this model's inaccuracy in CLWC predictions. Similar results are also demonstrated in the testing phase results.

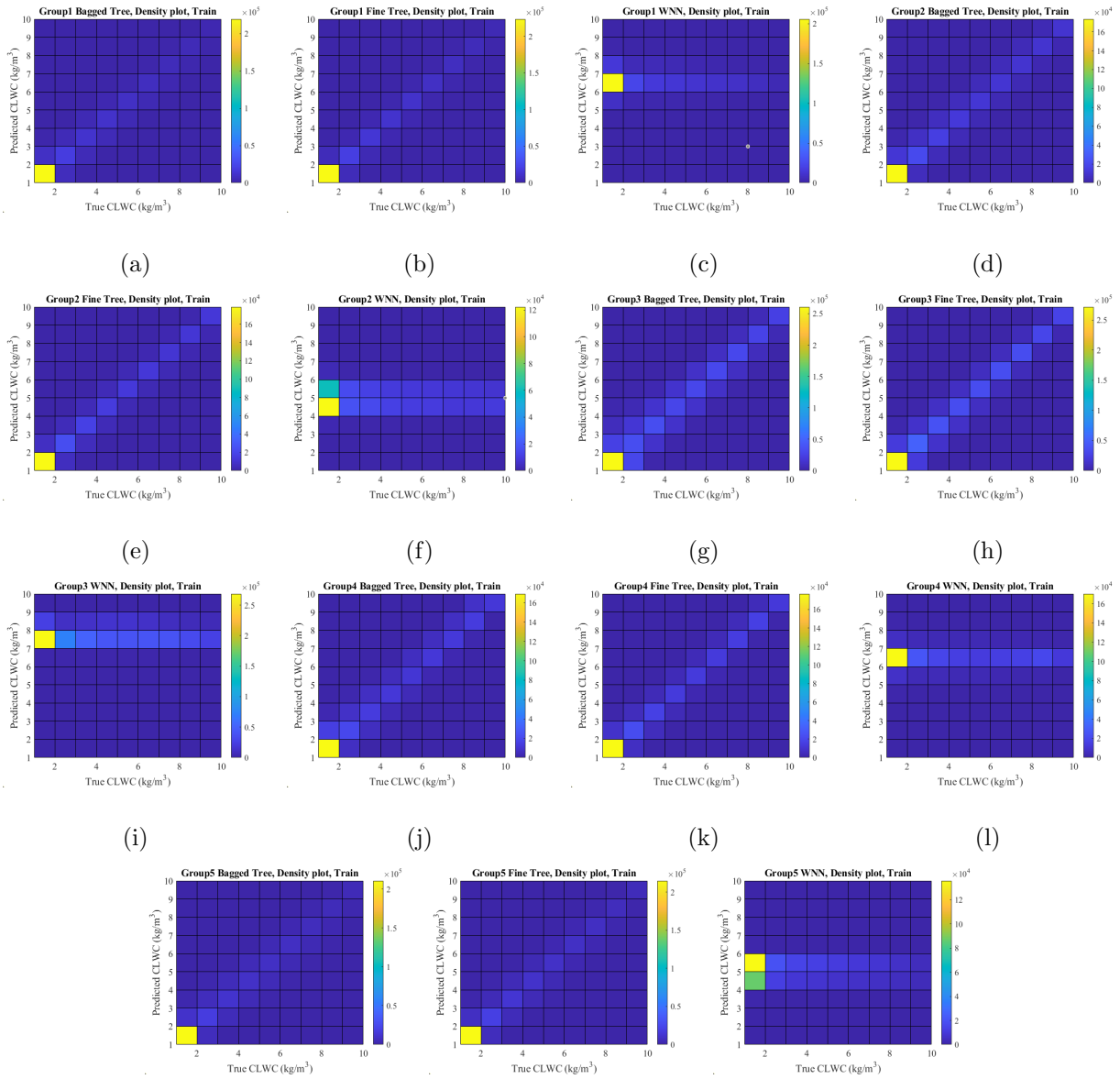


Figure 4.20: 2D histogram plots with color maps within of expected vs. actual CLWC of the groups' training dataset using different ML algorithms; (a) Group 1, Bagged tree (BT), (b) Group 1, Fine tree (FT), (c) Group 1, Wide Neural network (WNN), (d) Group 2, BT, (e) Group 2, FT, (f) Group 2, WNN, (g) Group 3, BT, (h) Group 3, FT, (i) Group 3, WNN, (j) Group 4, BT, (k) Group 4, FT, (l) Group 4, WNN, (m) Group 5, BT, (n) Group 5, FT, (o) Group 5, WNN. The color map indicates the frequency of data points, with different colors showing varying densities.

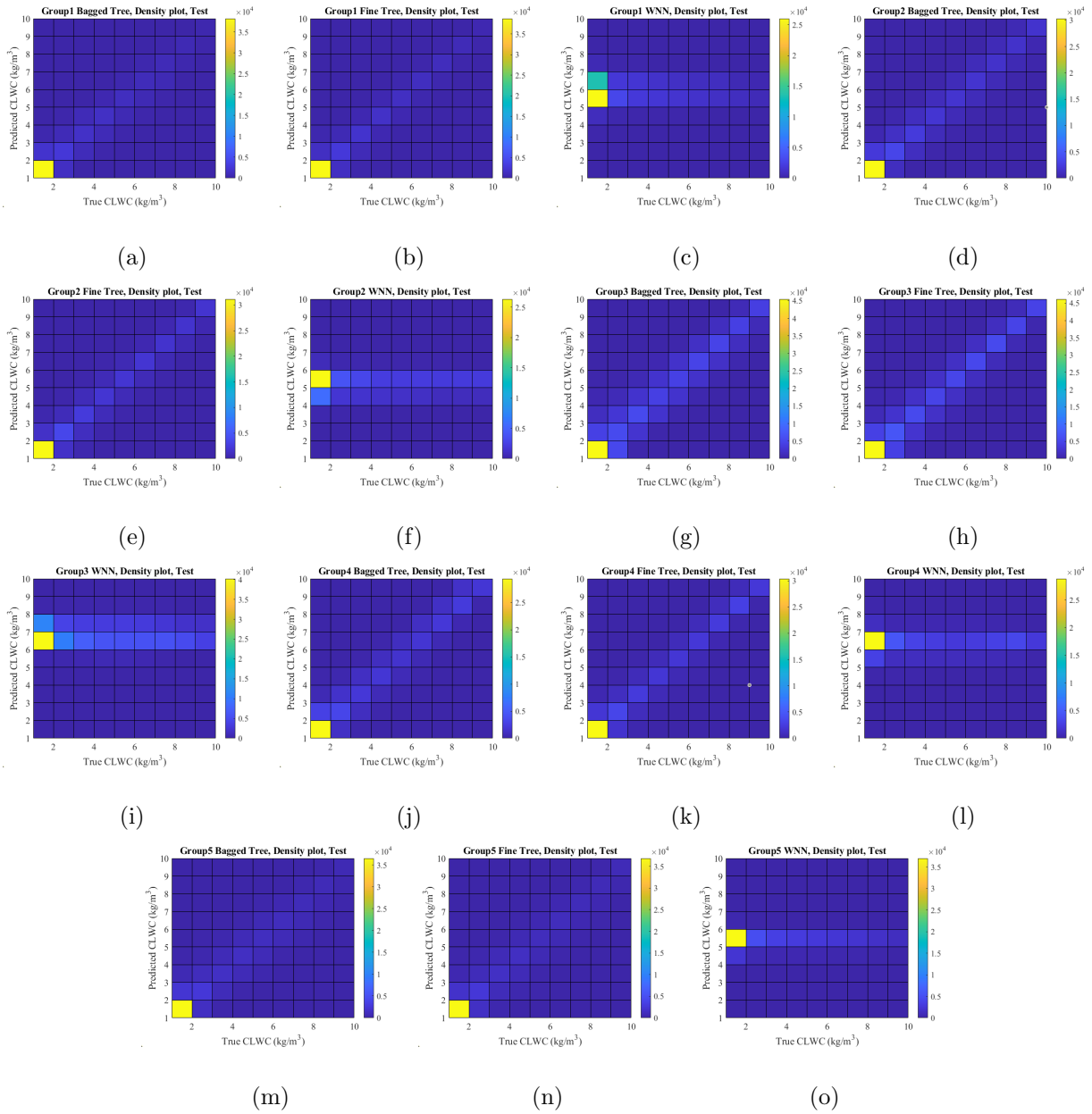


Figure 4.21: 2D histogram plots with color maps within of expected vs. actual CLWC of the groups' testing dataset using different ML algorithms; (a) Group 1, Bagged tree (BT), (b) Group 1, Fine tree (FT), (c) Group 1, Wide Neural network (WNN), (d) Group 2, BT, (e) Group 2, FT, (f) Group 2, WNN, (g) Group 3, BT, (h) Group 3, FT, (i) Group 3, WNN, (j) Group 4, BT, (k) Group 4, FT, (l) Group 4, WNN, (m) Group 5, BT, (n) Group 5, FT, (o) Group 5, WNN. The color map indicates the frequency of data points, with different colors showing varying densities.

Figure 4.22 contains 15 plots comparing the predicted CLWC profiles against the actual CLWC profiles. This figure illustrates the Bagged Tree model's ability to predict CLWC across various atmospheric pressures for five distinct climatological groups. Figure 4.22 starts with the first three subfigures showing the comparison of CLWC profiles for group 1, followed by the next three subfigures for group 2, and so on for the remaining groups. The y-axis in each subfigure has been reversed to represent the actual altitudes accurately. For each climatological group, the predicted CLWC profiles are presented alongside the actual CLWC values at different times. The Bagged Tree model consistently aligns its predictions with the actual CLWC profiles across all examples, evidenced by the high correlation coefficients ranging from 0.983 to 0.996. This indicates the model's robust performance across all atmospheric pressure levels and adaptability in varying climatic conditions.

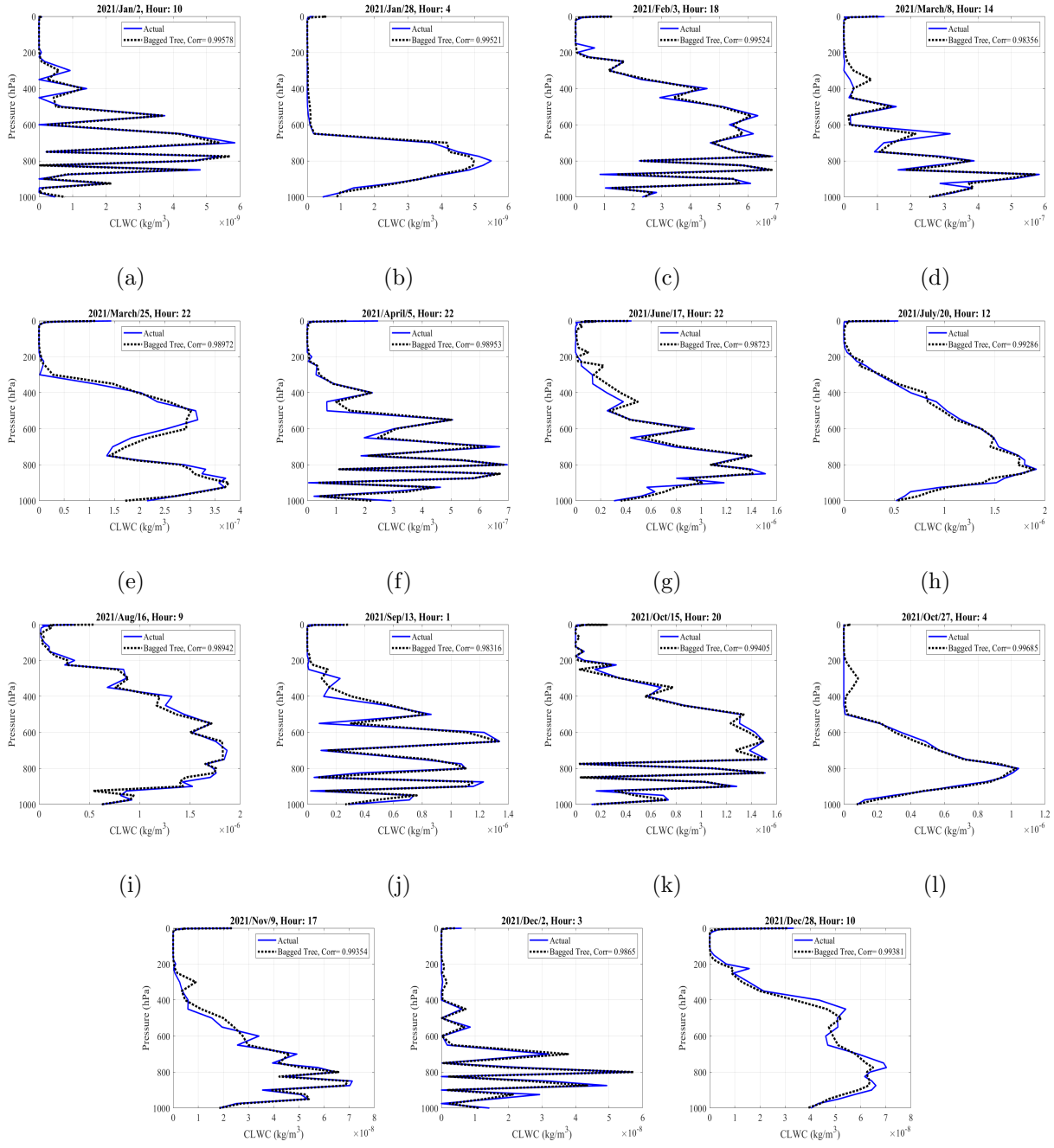


Figure 4.22: Vertical profiles of actual vs predicted CLWC values for the groups' data using the Bagged tree.

4.7 Conclusions

This chapter developed a machine learning (ML)-based methodology for estimating Cloud Liquid Water Content (CLWC) profiles by integrating ERA5 climatic data from the European Centre for Medium-Range Weather Forecasts (ECMWF) with detailed atmospheric observations from Hong Kong throughout 2021. One important aspect of this work was the use of ML-driven interpolation methods to align ERA5 data with radiosonde measurements. By effectively handling discrepancies in altitude, date, and time, these interpolation models ensured the reliable integration of both datasets. This method allowed for accurate temperature and relative humidity predictions, making it a vital part of the data validation process. The ability of the interpolation models to bridge data gaps contributed significantly to the overall accuracy of CLWC predictions, demonstrating their effectiveness in real-world applications. A key novel contribution of this study was the development of a metaheuristic algorithm for preprocessing the ERA5 data, which significantly enhanced the correlation between input features such as temperature, pressure levels, relative humidity, and specific humidity, and the target CLWC. This data-cleaning approach was designed to remove anomalies and noise, ensuring that the patterns modeled by the ML algorithms reflected true atmospheric phenomena rather than data artifacts. The novelty of this approach lies in its ability to preserve both linear and non-linear relationships between the observations and CLWC, which is an improvement over traditional methods that may oversimplify these complex relationships. As a result, this preprocessing step ensured that the ML models could predict CLWC more accurately. Furthermore, the dataset was segmented into groups based on similar atmospheric features, allowing for the use of specific ML models tailored to different climatic conditions. A comprehensive evaluation of the 2021 data revealed that decision tree models, particularly the Fine Tree, exhibited high accuracy, with the lowest RMSE and MSE values, and showed a strong fit to the data through high R-squared values. Among the ensemble models, the Bagged Tree model demonstrated the best predictive accuracy across both full-year and grouped monthly datasets. These results indicate the

superiority of tree-based models for handling the complex, dynamic nature of CLWC data. In contrast, shallow neural network (NN) architectures, particularly those using the ReLU activation function, underperformed due to inactive neurons, leading to degraded results. This highlights the limitations of certain neural network architectures for this particular problem and underscores the strength of ensemble models in this study. Looking ahead, the methods developed in this study will be extended to test their effectiveness in diverse environmental scenarios and to explore their integration into operational forecasting systems. Future efforts will involve refining the models with more advanced data preprocessing techniques and incorporating additional atmospheric parameters to improve the accuracy of CLWC predictions further. An important future step will be comparing ML predictions of CLWC with data from ground-based instruments and satellites, such as microwave radiometers and MODIS, to validate and enhance the models. Additionally, this work will explore real-time applications of these models, with the aim of incorporating them into disaster management systems and climate monitoring frameworks. Finally, exploring adaptive ML models that can be updated in real-time as new data becomes available will help improve the models' robustness and adaptability to dynamic atmospheric conditions. These adaptive models would offer more scalable solutions and could further extend the operational use of ML in atmospheric prediction.

Table 4.4: Enhanced CLWC Correlations Post-Cleaning for the Full-Year Data. ρ is the Correlation Coefficient.

	CLWC Old ρ	**CLWC New ρ**	**Parameters**
Pressure	0.21	0.68	[next, percentiles, movmean, 29.87, 6.63, 91.37, 1.32]; [center, percentiles, movmean, 92.41, 0.43, 82.82, 1.17]
Temperature	0.20	0.69	[center, percentiles, gaussian, 166.88, 17.80, 90.84, 1.20]; [center, percentiles, movmean, 92.41, 0.43, 82.82, 1.171]
Relative Humidity	0.27	0.76	[next, percentiles, gaussian, 169.08, 4.68, 96.75, 1.62]; [center, percentiles, movmean, 92.41, 0.43, 82.82, 1.17]
Specific Humidity	0.22	0.73	[linear, percentiles, movmedian, 157.10, 1.64, 91.09, 1.18]; [center, percentiles, movmean, 92.41, 0.43, 82.82, 1.171]

Table 4.5: CLWC Correlation Improvements by Climatic Group Post-Optimization.

	CLWC Old ρ	CLWC New ρ
Pressure (Group 1)	0.17138	0.61926
Temperature (Group 1)	0.15363	0.63219
RH (Group 1)	0.32388	0.76979
SH (Group 1)	0.27575	0.62938
Pressure (Group 2)	0.27649	0.7771
Temperature (Group 2)	0.24738	0.74314
RH (Group 2)	0.34682	0.79723
SH (Group 2)	0.33006	0.77009
Pressure (Group 3)	0.21278	0.81672
Temperature (Group 3)	0.21516	0.63519
RH (Group 3)	0.21088	0.71134
SH (Group 3)	0.18394	0.72645
Pressure (Group 4)	0.21288	0.77817
Temperature (Group 4)	0.20554	0.60512
RH (Group 4)	0.25367	0.74477
SH (Group 4)	0.20274	0.66778
Pressure (Group 5)	0.1898	0.63489
Temperature (Group 5)	0.17305	0.66359
RH (Group 5)	0.30275	0.79155
SH (Group 5)	0.27845	0.72957

Chapter 5

Identification and Mitigation of 5G Interference for Radar Altimeters

5.1 INTRODUCTION

Radar Altimeters (RAs), also known as radio altimeters, are critical commercial and civil aircraft sensors that improve safety and navigation at all phases of flight (ITU 2003). These devices enhance situational awareness by measuring clearance height above terrain and potential obstacles, assisting Automatic Flight Guidance and Control Systems (AFGCS) during instrument approaches, and contributing to systems such as the Engine-Indicating and Crew-Alerting System (EICAS), Predictive Wind Shear (PWS), and Electronic Centralized Aircraft Monitoring (ECAM) (Benveniste 2011). In civil and commercial aircraft, up to three radar altimeters can be employed simultaneously to guarantee accurate altitude measurements. These altimeters, which include pulsed and frequency-modulated continuous-wave (FMCW) radars, measure the aircraft's above-ground level (AGL) height by emitting radio frequency (RF) energy to the ground and measuring the round-trip propagation time of the reflected energy (Coleman 2001).

The introduction of 5G technology marks an advancement in communication capabilities, providing higher data speeds and connectivity for diverse applications worldwide (Shafi et al. 2017). As countries rapidly deploy these networks, using the C-band spectrum for 5G systems

has not fully considered the electromagnetic compatibility issues with other systems using the C-band spectrum. For example, agencies are reallocating a portion of the 3.7–4.2 GHz frequency band for 5G use, with the 3.7–3.98 GHz spectrum being assigned to new licensees since December 2020. This reallocation places the 5G frequencies close to those used by radar altimeters, raising concerns about potential signal interference (Amaireh and Zhang 2023). Such interference could degrade altimeter functionality, significantly endangering aviation safety. Two types of 5G fundamental radiations might cause interference to the radar altimeter: emissions within the source's operating frequency band (in-band) and spurious emissions falling inside the 4.2-4.4 GHz band (out-of-band) (RTCA 2020).

As part of the existing solutions, band-pass filters (BPF) were incorporated to control the interference in the 3.7 to 3.98 GHz range. These filters are suitable for direct integration with radar altimeter systems. However, while these static filters provide an immediate solution, they cannot adapt to the rapidly evolving 5G spectrum and unexpected signal variations. Also, another study in (Electronics 2023) discusses using BPFs optimized for minimal propagation delays and sharp transitions, which are effective in controlled scenarios but require precise engineering to match specific interference profiles, making them less adaptable in variable conditions. Furthermore, the study in (Kyriakos 2023) employs a more dynamic approach by adjusting its frequency response based on altitude measurements. While this is an improvement over static filters, it still cannot handle unexpected conditions. In contrast, the digital-signal-processing method offers significant advantages by continually adapting and learning from data, making it highly effective in environments with variable 5G interference (Jiang et al. 2023). ML models effectively handle complex, non-linear data relationships, making them suitable for aviation environments where traditional filters may fail (Cai et al. 2021). ML enhances radar system efficiency by automating adjustments and leveraging real-time data, which reduces costs and minimizes errors (Lang et al. 2020). The initial data and computing infrastructure investment pays off by decreasing the need for manual re-calibrations and lowering maintenance costs. ML's scalability allows for ongoing

updates without hardware modifications, offering cost-effectiveness over time (Lang et al. 2020).

This chapter uses Machine Learning (ML) models and algorithms to classify radar altimeter signals subjected to 5G NR interference and improve the accuracy of altitude measurements in the presence of potential 5G interference in radar altimeters. Figure 5.1 shows a flow chart detailing all of the steps included in this chapter. The technique begins with careful signal collection and validation using a low-cost Software-Defined Radio (SDR), resulting in a dataset that supports examining interference risks such as spurious emissions, receiver front-end overloading, receiver desensitization, and inter-modulation products. The study simulates radar altimeter signals and refines the dataset to include around 20,000 signals across a 1.676-kilometer range to ensure a wide variety of data for training and testing. The preprocessing step begins with segmenting the 5G base station signals and re-scaling them to emulate various interference levels, followed by mitigating outliers from the data before employing them in the ML processes. Subsequently, ML classification models are implemented to differentiate between the radar altimeter signals and 5G NR interference while also considering the particular challenges presented by different waveform characteristics. After detecting interference, two ML models are used to estimate flight altitudes for radar waveforms using down and up-frequency sweeps. By averaging the altitude predictions from these two models, a more accurate and consistent measurement is achieved. The chapter concludes with a detailed comparison of the ML models' performance, illustrating their effectiveness and the effect of frequency modulation sweep direction on altitude prediction in the presence of 5G interference.

The key novelty in this work is the use of machine learning to effectively classify radar altimeter signals and mitigate interference caused by 5G networks. Unlike previous studies that focused on static or hardware-based filtering solutions, this study demonstrates that ML models can successfully distinguish radar signals from interference while also providing accurate altitude estimations under varying interference conditions. To the best of our

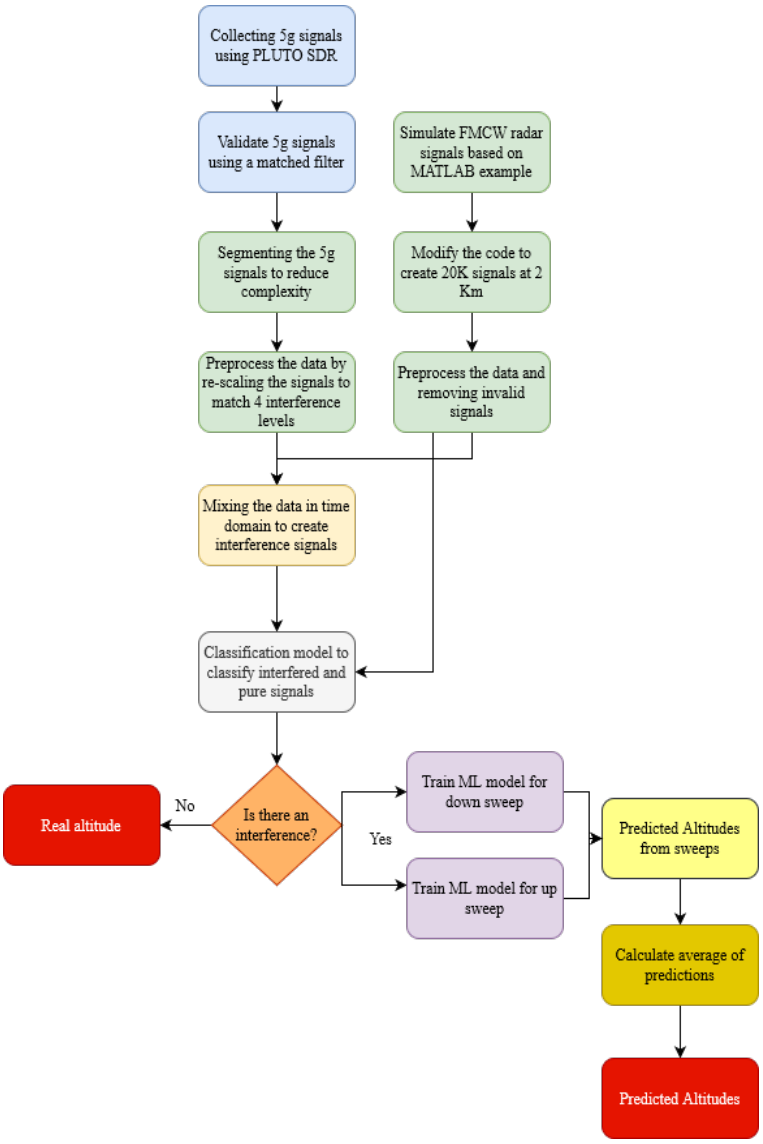


Figure 5.1: Complete processing flow for classifying and mitigating 5G interference in radar altimeter signals using a machine learning framework.

knowledge, this is the first work to address the issue of 5G interference in radar altimeters and resolve it using ML. While the real-time application was not implemented in this research, the ML models trained and tested here show potential for integration into real-time systems, offering more flexible and adaptive solutions than conventional methods. By leveraging the strengths of ML, this work presents a promising solution for handling high-interference environments, which could improve aviation safety in future applications.

The rest of this chapter is organized as follows: Section 5.2 describes the data collection process, including the methods for 5G signal collection, verification, and using a radar altimeter simulator to emulate 5G interference. Section 5.3 details the methodology, covering feature extraction from time-domain and frequency-domain data, preprocessing steps, performance evaluation metrics, and the machine learning algorithms and methods utilized in this study. Section 5.4 presents the results and discussion, including classification and altitude prediction results, with specific cases for up-sweep and down-sweep scenarios. Finally, Section 5.5 concludes the chapter and suggests future work.

5.2 Data Collection

5.2.1 5G Signal Sampling and Data Collection

The ADALM-PLUTO Software-Defined Radio (SDR) was used to collect 5G signals at a center frequency of 3.7 GHz, which is adjacent to the aviation radar altimeter range of 4.2-4.4 GHz. The ADALM-PLUTO, which has an operating range of up to 3.8 GHz, was tuned to a central frequency of 3.7 GHz and configured for high-precision capture at 2 million samples per frame. This configuration enabled us to collect a diverse dataset of 1000 signal frames across various times, locations, and weather conditions, providing insights into 5G signal behavior and its potential for interference with adjacent bands. Focusing on the 3.7 GHz band, particularly the n77/n78 5G bands, allowed us to explore possible interference with radar altimeters.

5.2.1.1 Verification of 5G Signal

To confirm that the collected signals are from actual 5G base stations, the cross-correlation with key 5G synchronization signals: the Primary Synchronization Signal (PSS) and the Secondary Synchronization Signal (SSS) was investigated. These signals, broadcast by base stations, are critical for mobile devices to identify and synchronize with the network, hence

initiating the cell search process. The PSS, with its high autocorrelation and specific structure, aids in accurate time synchronization and identifies the cell's Physical Layer Cell Identity Group (PCI Group). Following PSS detection, the SSS refines the cell identification and is required to determine the frame boundary and access broadcast channel information. Cross-correlating the received signal with PSS and SSS sequences effectively confirms their presence. Distinct peaks in the correlation output signal the alignment of the received signal with these sequences, indicating genuine 5G NR signals.

In addition, the power spectrum density plots shown in Figure 5.2 display a prominent peak at the center, characteristic of the carrier frequency used in 5G communications. The shape of the power spectrum, with its distinct roll-off, further supports the spectral profiles expected of 5G NR transmissions. The cross-correlation analysis using the PSS, as shown in Figure 5.3 (a), produces noticeable peaks that rise above the noise level. These peaks indicate the PSS within the received signal. Similarly, Figure 5.3 (b) shows that the cross-correlation with the SSS. The presence of significant peaks at specified lags verifies the detection of the SSS.

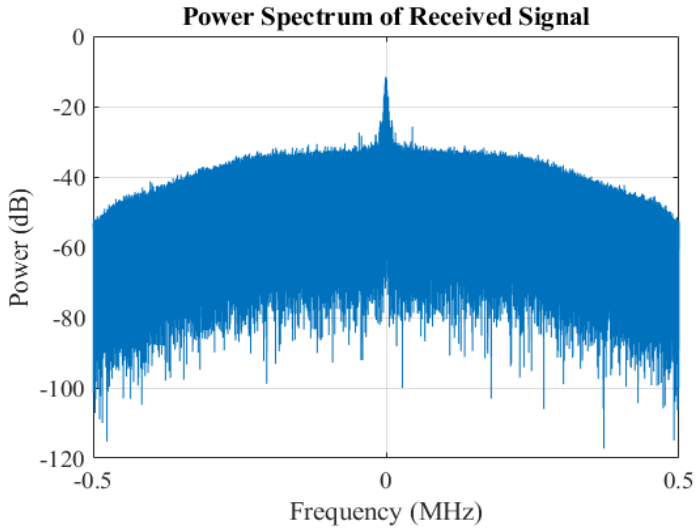
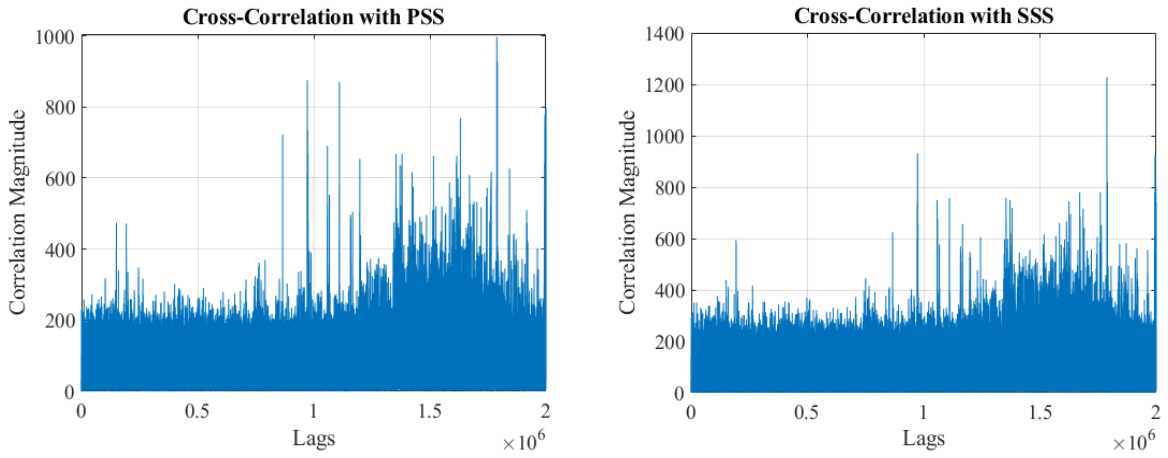


Figure 5.2: Power Spectrum of the Received 5G Signal.



(a) (b)

Figure 5.3: Cross-Correlation Analysis with (a) Primary Synchronization Signal (PSS), and (b) Secondary Synchronization Signal (SSS).

5.2.2 Simulation of Radar Altimeters

A system simulation of an FMCW radar altimeter was developed and used to generate radar I/Q data. The radar parameters are detailed in Table 5.1. Waveform and sensor parameters are set to simulate the airplane landing process beginning at an altitude of 1,676 meters. The simulation includes the ability to adjust different losses, terrain types, etc. Around 20,000 radar signal samples are simulated for various aircraft altitudes. Standard FMCW radar signal processing (Boris Atayants 2014) is applied for the altitude estimations, which are updated at a 10 Hz rate.

Figure 5.4 displays spectrograms of the received radar altimeter signals at altitudes of 1493.29 m, 969.68 m, 371.51 m, and 76.57 m, respectively. Signal strength visibly increases as altitude decreases, marked by the values changing in warmer colors. The pattern structure in the spectrograms at higher altitudes is potentially caused by the radar's antenna beam coverage at higher altitudes. This spreads across frequencies, particularly when the radar approaches its maximum altitude of 1,676 meters. As the altitude decreases, the radar's footprint narrows, and the spectrum spreading decreases, leading to "straight line" type

Table 5.1: Parameters of the FMCW radar altimeter used in the simulation.

Parameter	Value
Center Frequency	4.3 GHz
Bandwidth	150 MHz
Chirp Repetition Frequency (CRF)	22 KHz
Antenna Beamwidth	40 degrees
Transmitter Power	0.4 W
Maximum Range	1,676 m
Sweep Direction	Triangle
Noise Figure	8 dB

of spectral patterns. This characteristic is one of the possible features used in the ML algorithms to discriminate radar signals from 5G-base station signals from the ground.

5.2.3 Emulating Radar Altimeter Signals and 5G RFI

Integrating real 5G and simulated radar signals through time-domain superposition of In-phase and Quadrature (IQ) data was performed. A segmentation approach was applied to handle the challenge of time sampling in the captured 5G signals. This technique maintained the integrity of 5G interference characteristics while simplifying the number of input features for effective machine learning (ML) training. Following segmentation, the airplane landing process was simulated with two critical factors: the increase in radar altimeter signal power as the aircraft descends and the rise in 5G interference levels caused by the aircraft's proximity to the 5G base stations. For the detection and 5G-RFI classification, the interference was classified into four level categories during descent: low (for altitudes between 1,676 and 1,138 meters), moderate (for altitudes between 1,138 and 782.6154 meters), high (for altitudes between 782.6154 and 427.5399 meters), and extreme (for altitudes between 427.5399

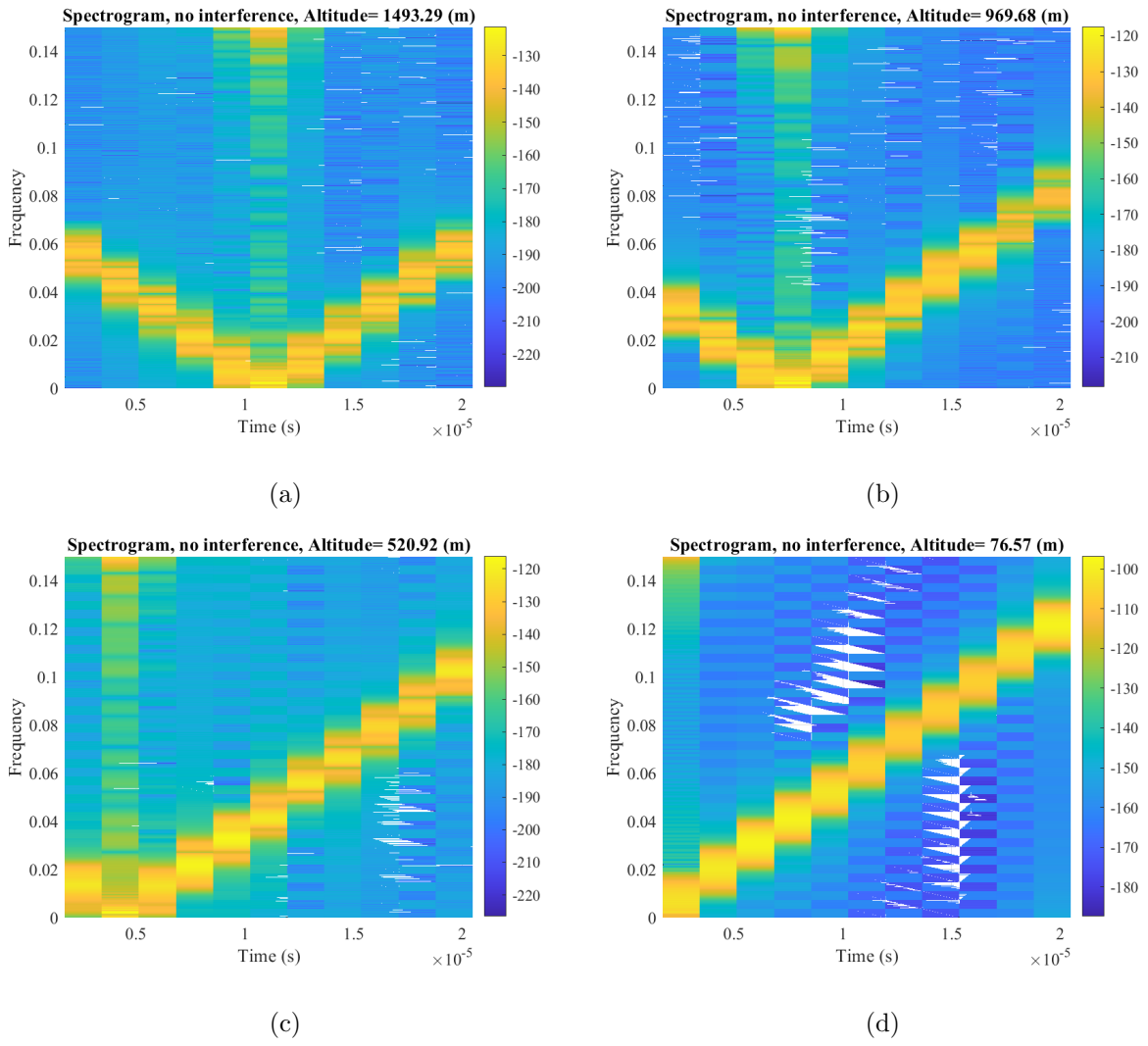


Figure 5.4: Spectrogram of the sample FMCW radar altimeter signals at the following altitudes: (a) 1493.29 meters, (b) 969.68 meters, (c) 520.92 meters, and (d) 76.57 meters.

meters and the ground). The raw 5G signal segments were rescaled to match the simulated aircraft altitudes, using statistical power values from radar altimeter signals at each altitude. Furthermore, the re-scaled 5G and radar altimeter signals were randomly combined for each interference category. Mixing the adjusted 5G interference signals with the simulated radar altimeter signals produced a dataset reflecting the full range of interference levels correlated with the aircraft's altitude. Figure 5.5 illustrates spectrograms for different signals at different altitudes before ("pure" signal) and after adding the interference signals to the pure

radar altimeter signals. As the simulated spectrum shows, the Radio Frequency Interference (RFI) from the 5G signals mainly affects the radar altimeter through the out-of-band (OOB) spectrum. It can be more significant at some time due to its noise-like nature. At a lower altitude of 371.51 meters with high interference, there is a noticeable distortion in the signal's spectral components, illustrating the interference's significant impact. The effect of interference is less significant at higher altitudes. Figure 5.6 shows an example of signal segments in the time domain before and after adding the interference. Notably, at the lower altitude of 371.51 meters, the interfered radar signal displays significant fluctuations in amplitudes.

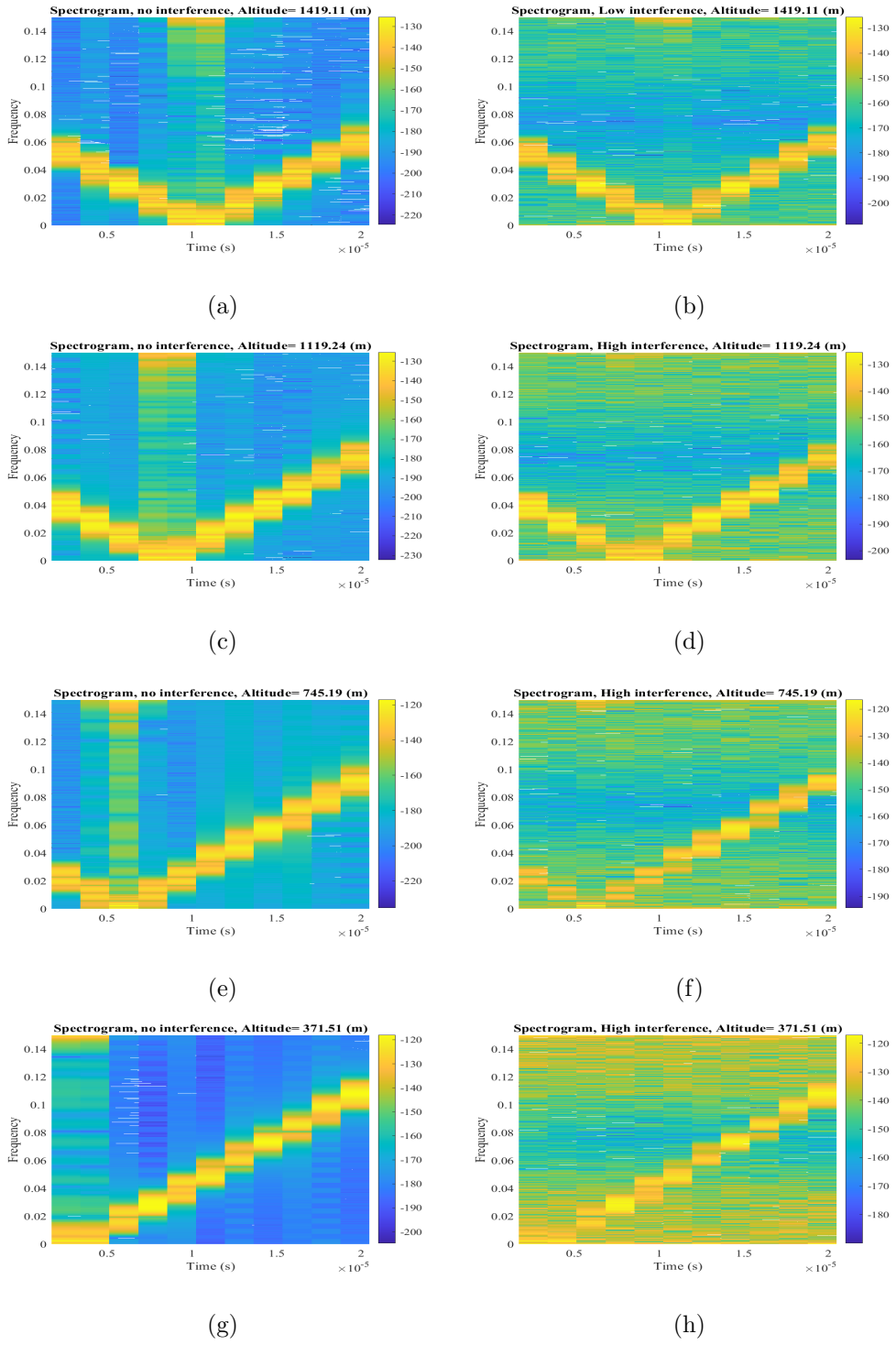


Figure 5.5: Spectrograms Showing the Effect of 5G Interference on Radar Altimeter Signals at Various Altitudes: (a) 1419.11 meters, (b) 1119.24 meters, (c) 745.19 meters, and (d) 371.51 meters.

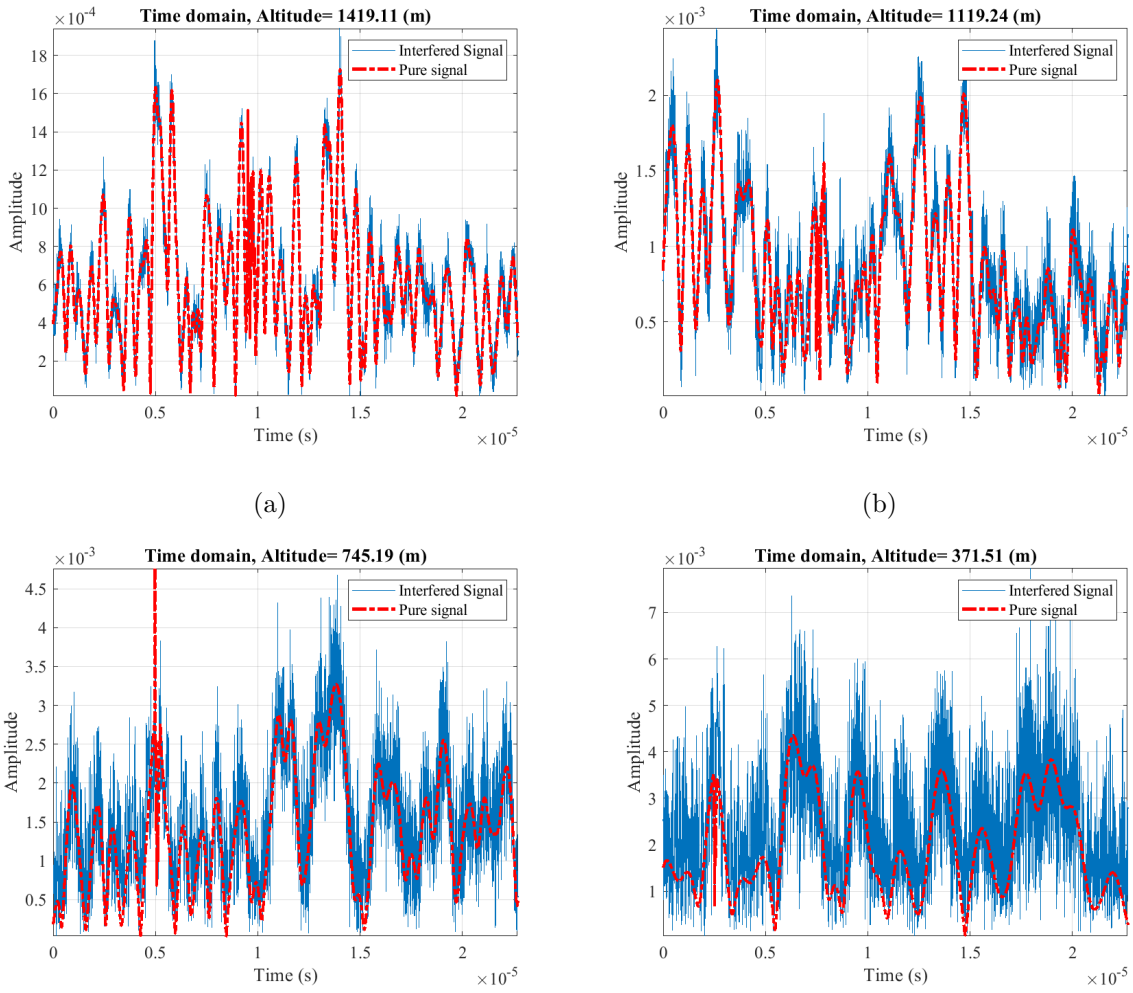


Figure 5.6: Time-Domain Analysis of Radar Altimeter Signals with and without 5G Interference: (a) 1419.11 meters, (b) 1119.24 meters, (c) 745.19 meters, and (d) 371.51 meters.

5.3 Detailed Methodology

5.3.1 Feature Extraction

5.3.1.1 Time-domain features

Based on typical signal feature analysis (Houidi et al. 2020), Time-domain signal features include peak amplitude, Root Mean Square (RMS) amplitude, kurtosis, skewness, standard

deviation, variance, signal power, and Signal-to-Noise Ratio (SNR). Peak amplitude detects important signal reflections, and RMS amplitude, which distinguishes signals from noise, is critical for altitude determination. Kurtosis and skewness provide insights into the signal's amplitude distribution, showing potential outliers and asymmetry that could be introduced by interference (Blanca et al. 2013). Standard deviation and variance measure amplitude variations and signal power, together with SNR, measuring the signal's overall strength and prominence over noise. The statistical distributions of the "pure" and interfered signals for down and up sweeps are shown in Figure 5.7, in form of the boxplots for peak amplitude, Root Mean Square (RMS) amplitude, signal kurtosis, skewness, standard deviation (STD), and signal power, which collectively present the effects of 5G interference on radar altimeter signals with clear attenuation in features during down sweeps. The peak amplitude boxplots show attenuation in the presence of interference, especially during down sweeps, indicating possible difficulty in identifying ground reflections at lower altitudes, especially important during takeoff and landing. Kurtosis and skewness variations in the interfered radar signals show clear non-normal distributions with heavier tails and asymmetric amplitude values.

Figure 5.8 illustrates the impacts 5G-RFI on the radar signal amplitude and power features at different altitudes. The power profile does not show a consistent reduction across all altitudes. Instead, there is a substantial difference between the original and interfered signals at lower altitudes, which narrows as altitude increases. This pattern indicates a complex relationship between interference strength and altitude. Furthermore, the peak amplitude profile shows that interference has the greatest influence in the critical lower altitude zones, affecting important aircraft operations such as takeoff and landing. Integrating these time-domain features into machine learning models improves prediction accuracy, allowing for a more accurate estimate in the presence of interference. This approach assures that machine learning algorithms can handle the complexity of signal variance, keeping the integrity and reliability of radar altimetry in aircraft.

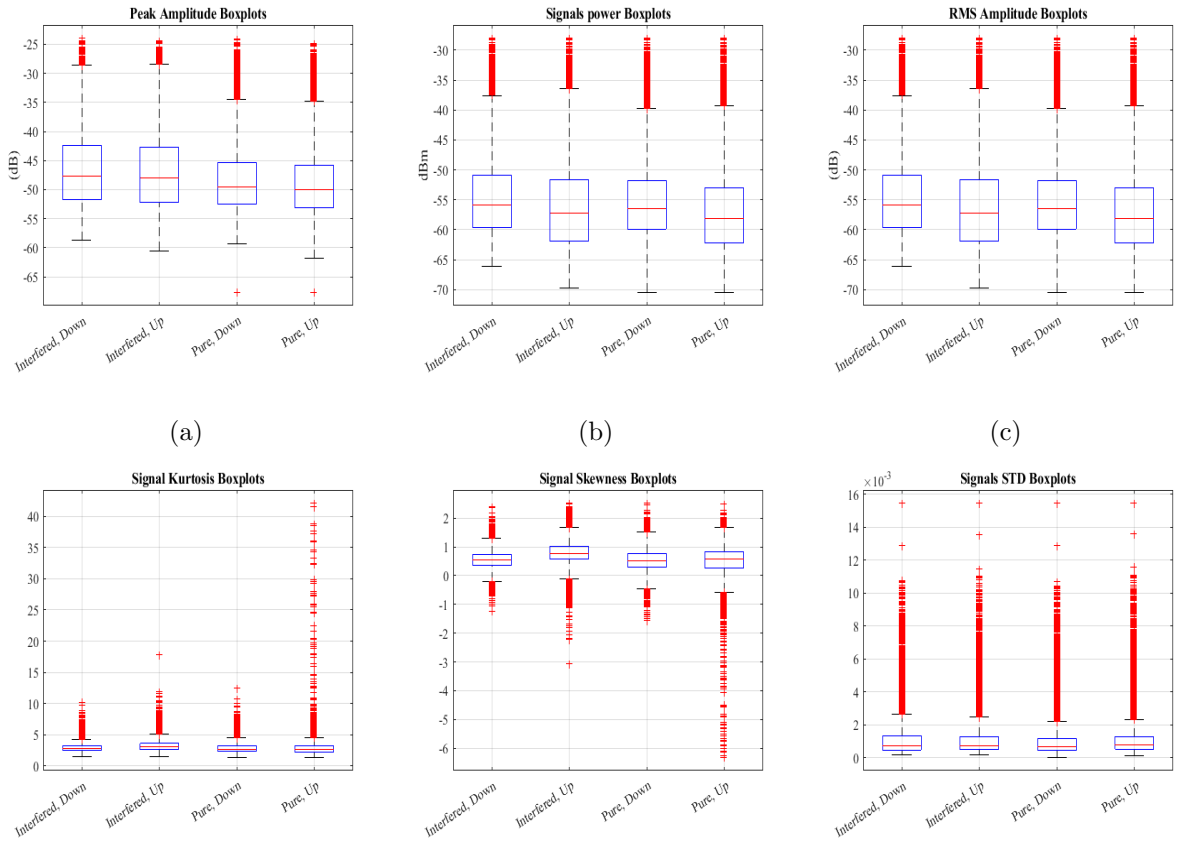
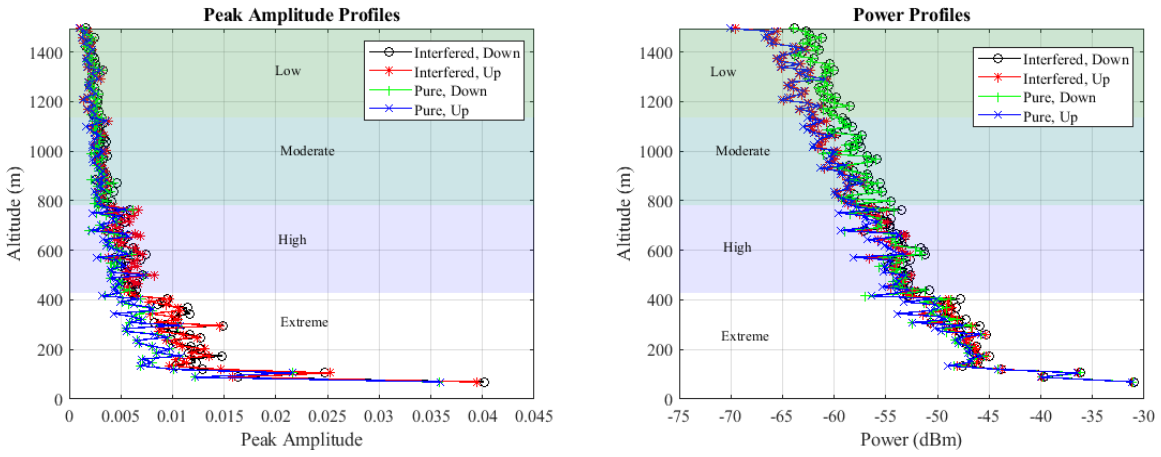


Figure 5.7: Boxplots of the time-Domain Features used by the ML processing for the original and interfered radar altimeter signal for altitude = 317.5 meters. All the features are based on the amplitudes of one time-frame capture. (a) Peak amplitudes, (b) Signals power values, (c) RMS amplitude values, (d) Signal Kurtosis, (e) Signals' skewness, (f) Signal amplitudes' standard deviation (STD).

5.3.1.2 Frequency domain features

Spectral features have been computed for the original (or "pure") and interfered radar altimeter signals across the up and down sweeps. The calculated features include the Spectral Centroid, Bandwidth, Flatness, Rolloff, Slope, Crest, Decrease, Entropy, Energy, Edge Frequency, and Zero Crossings. These features are essential to machine learning (ML) frameworks that aim to improve the accuracy of altitude predictions and interference detection. Spectral features have been computed for the pure and interfered radar altimeter signals



(a) (b)

Figure 5.8: Altitude-Dependent Profiling of (a) Peak Amplitudes, and (b) Power levels for the Original Radar Altimeter Signals and the Signals with the 5G-RFI Effects.

across up and down sweeps. The calculated features include the Spectral Centroid (Schubert et al. 2004), Bandwidth, Flatness (Johnston 1988), Rolloff (Scheirer and Slaney 1997), Slope, Crest, Decrease, Entropy, Energy, Edge Frequency, and Zero Crossings. Among these features, Spectral Decrease and Entropy are critical to evaluate amplitude decline and randomness in the spectral distribution, respectively (Misra et al. 2004; Peeters 2004). Spectral energy and spectral edge frequency further enrich the analysis by representing the total signal power and defining the upper-frequency boundary of the signal's energy, respectively (Stoica et al. 2005). Finally, spectral zero crossings indicate frequency stability or instability (Stoica et al. 2005). Figure 5.9 illustrates the spectral centroid, bandwidth, and entropy features. Interference usually increases spectral bandwidth, as seen by the broadened frequency spread, and introduces variability in the spectral centroid, indicating variations in the energy distribution within the spectrum. Spectral entropy increases to imply that interference causes more uncertainties. This study used time-domain and frequency-domain signal features for RFI detection and possible future mitigation. However, more weighting was placed on the time-domain features based on computational efficiency considerations.

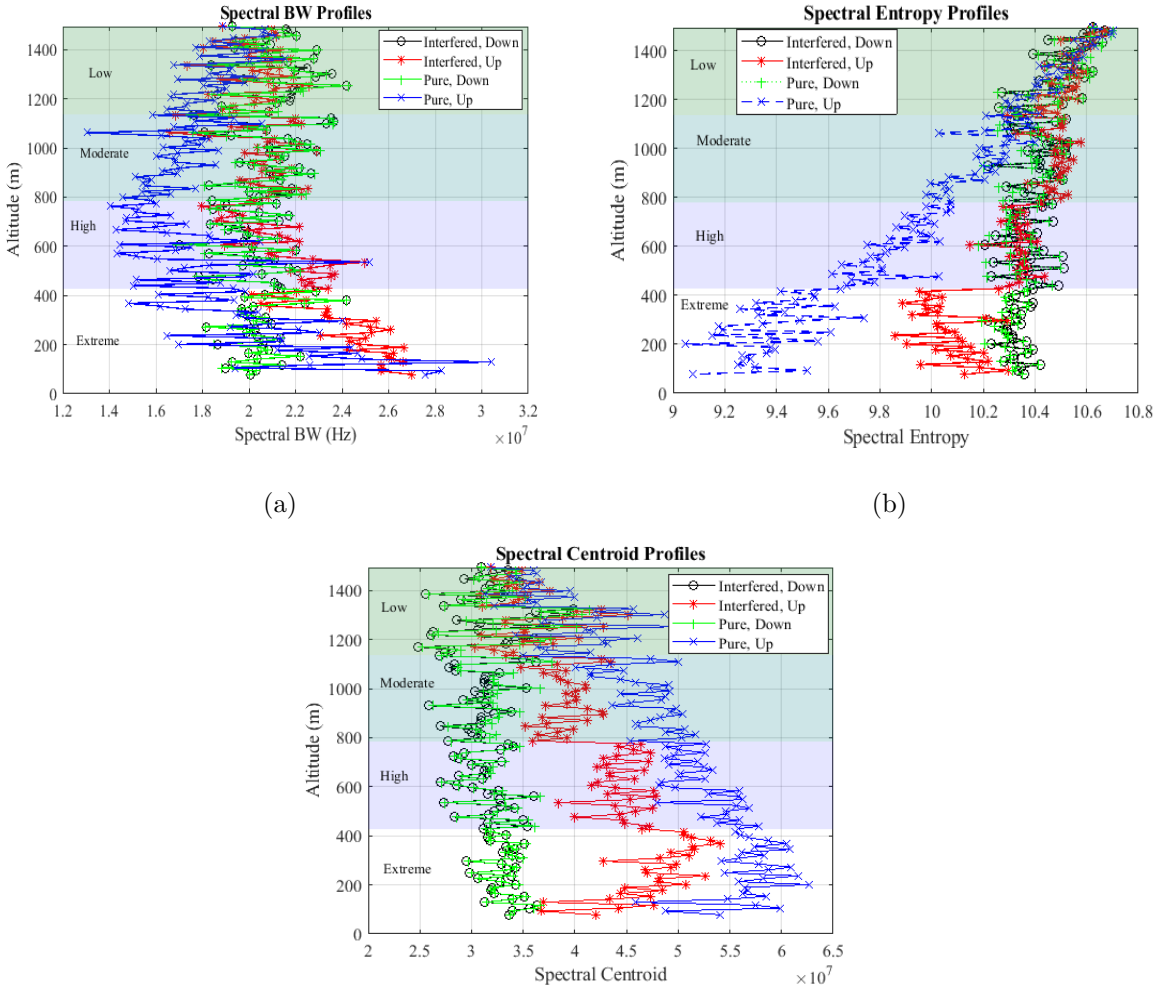


Figure 5.9: Altitude-Dependent Profiling of (a) Spectral BW, (b) Spectral Entropy, and (c) Spectral Centroid for Up and Down Sweeps for the Original Radar Altimeter Signals and the Signals with the 5G-RFI Effects.

5.3.2 Prepossessing

The preprocessing step extracts and combines both time-and-frequency domain features into training and testing datasets. A signal classification approach based on time-domain samples and features is proposed by maintaining the original radar receiver sampling frequency of 149.996 MHz within a 150 MHz bandwidth. The dataset was segmented for altitude estimations based on sweep direction (up or down) before ML model training. This technique

makes creating specific ML models for each sweep direction easier, using the unique altitude information within each. Training separate models for up and down sweeps also allowed for capturing distinct altitude aspects more effectively. Next, a five-fold cross-validation across classification and altitude prediction phases was implemented for enhanced model reliability and generalizability. This technique, critical for assessing model performance consistently across various scenarios, was complemented by outlier detection and removal to refine data quality.

5.3.3 ML Algorithms and Methods Used in This Study

This study applies various ML algorithms to classification (detecting the presence of 5G-RFI) and regression (estimating the approaching altitudes in the presence of 5G-RFI). Table 5.2 details the machine learning models applied for classifying pure and interfered signals, including their hyperparameters. These ML models were selected to balance classification accuracy and computational efficiency. Tree-based models like Fine, Medium, and Coarse Trees were chosen for their interpretability and effectiveness in managing complex, high-dimensional data, making them ideal for distinguishing between pure and interfered signals. Ensemble trees, such as Bagged Trees, were favored for their robustness and ability to prevent overfitting by combining the predictions of multiple trees. Neural Networks (NN) were included due to their ability to capture non-linear relationships, though simpler NN architectures, such as Narrow and Medium NN, were preferred to avoid overfitting in this relatively smaller dataset. The Tri-layer Neural Network (TriNN), a more complex model, was tested for its ability to manage intricate data relationships. To prevent overfitting, the model was carefully tuned, adjusting parameters like the number of neurons and layers to maintain optimal performance.

For regression altitude prediction, the models employed and their respective hyperparameters are detailed in Table 3.1. The methodology was executed on a computer with an

i7-2600K CPU and 24 GB of RAM. The integration of these strategies highlights a sophisticated balance between maintaining signal integrity and computational practicality.

Table 5.2: List of the hyperparameters of the classification models.

Model	**Parameters**
Fine Tree	Splits: 100, Criterion: Gini, Surrogate: Off
Medium Tree	Splits: 20, Criterion: Gini, Surrogate: Off
Coarse Tree	Splits: 4, Criterion: Gini, Surrogate: Off
Narrow NN	Layers: 1 (10), ReLU, Iter: 1000, λ: 0, Std: Yes
Medium NN	Layers: 1 (25), ReLU, Iter: 1000, λ: 0, Std: Yes
Wide NN	Layers: 1 (100), ReLU, Iter: 1000, λ: 0, Std: Yes
Bi NN	Layers: 2 (10, 10), ReLU, Iter: 1000, λ: 0, Std: Yes
Tri NN	Layers: 3 (10, 10, 10), ReLU, Iter: 1000, λ: 0, Std: Yes
Fine KNN	Neighbors: 1, Metric: Euclidean, Weight: Equal, Std: Yes
Gaussian NB	Distribution: Gaussian
Logistic Reg	None

5.4 Results and Discussion

5.4.1 5G-RFI Detection Results

The signal samples with various levels of RFI and without RFI are divided into testing (20%) and training datasets (80%). The first objective is to discriminate the radar signal samples containing any levels (low, moderate, high, and extreme, called Class 1) of RFI from

the "pure" radar altimeter signals (called Class 2). Table. 5.3 compares machine learning models' performance in radar signal classification under interference. The Fine Tree model has remarkable accuracy both in validation (98.24%) and testing (98.36%), as well as the lowest total cost in testing (187), making it the most effective model for exact classifications with few misclassification errors. Its prediction speed (11618.28 observations per second) and comparatively low training time (369.22 seconds) highlight its effectiveness. The Medium Tree and Coarse Tree models exhibit a decreasing pattern in accuracy and an increasing pattern in total cost over the validation and testing phases. Despite its lower accuracy, the Coarse Tree model has the fastest prediction speed (13788.99 obs/sec). Neural Network models perform differently, with the Narrow, Medium, and Wide configurations achieving test accuracies of 93.44%, 92.59%, and 92.85%, respectively. Notably, increasing network size does not directly correspond with higher accuracy, as seen by the Wide Neural Network's lengthy training period (21754.53 seconds), but only modestly better test accuracy than its narrower counterparts. The Bilayered and Trilayered networks underscore the declining results on increasing complexity, with test accuracies of 91.95% and 92.21%, respectively, with the Trilayered Neural Network having the longest training time (28930.28 seconds). The Fine KNN model significantly fails in accuracy and cost parameters; In contrast, the Gaussian Naive Bayes and Logistic Regression models provide feasible alternatives. However, they also demonstrate trade-offs between accuracy and computational demands.

Table 5.3: Performance of Radar Signal RFI Level Classification for Different ML Models.

Model	**Accuracy (%)**		**Total Cost**		**Speed**	**Time**
	Val	**Test**	**Val**	**Test**	**(obs/sec)**	**(sec)**
F. Tree	98.2	98.4	1138	187	11618	369
M. Tree	97.0	97.3	1926	313	12587	287
C. Tree	95.2	95.7	3096	490	13789	223
N. NN	91.4	93.4	5535	748	9651	9910
M. NN	91.5	92.6	5516	845	9375	8397
W. NN	91.7	92.9	5387	815	4283	21755
BiNN	91.2	91.9	5689	918	9554	10210
TriNN	91.0	92.2	5831	888	3834	28930
F. KNN	21.4	11.0	50755	10148	14	29474
G. NB	79.4	89.1	13303	1244	1120	29685
L. Reg.	90.6	87.2	6061	1459	2987	40387

Figure 5.10 depicts the confusion matrices of the training dataset for several ML models. The Fine Tree model's confusion matrix shows good accuracy, with true positives for Class 1 and Class 2 stated at 31325 and 32137, respectively, compared to very low false negatives and false positives (248 and 890). In contrast, the KNN model does much worse, with true positives of 4756 for Class 1 and 9089 for Class 2, but misclassification is significantly greater, particularly for false positives in Class 1 (27459). The Gaussian Naive Bayes model performs well, with true positives of 30913 for Class 1 and 20384 for Class 2. Given its performance, the model shows limits with 1,302 false negatives for Class 1 and 12,001 false positives for Class 2. Logistic regression performs effectively, with true positives of 28697 for Class 1 and 29842 for Class 2, indicating that it can classify accurately. The false positives and false negatives for Class 1 are 3518 and 2543, respectively, showing an acceptable degree of

accuracy. The Narrow Neural Network has true positives of 29329 for Class 1 and 29736 for Class 2, and there are false negatives (2649) and false positives (2886). The Tri-layered Neural Network's confusion matrix shows true positives 29320 for Class 1 and 29449 for Class 2, with fewer false negatives (2936) and false positives (2895). These results demonstrate the advantages of neural network architecture for certain classification tasks and the capability to leverage deeper networks to attain greater classification accuracy.

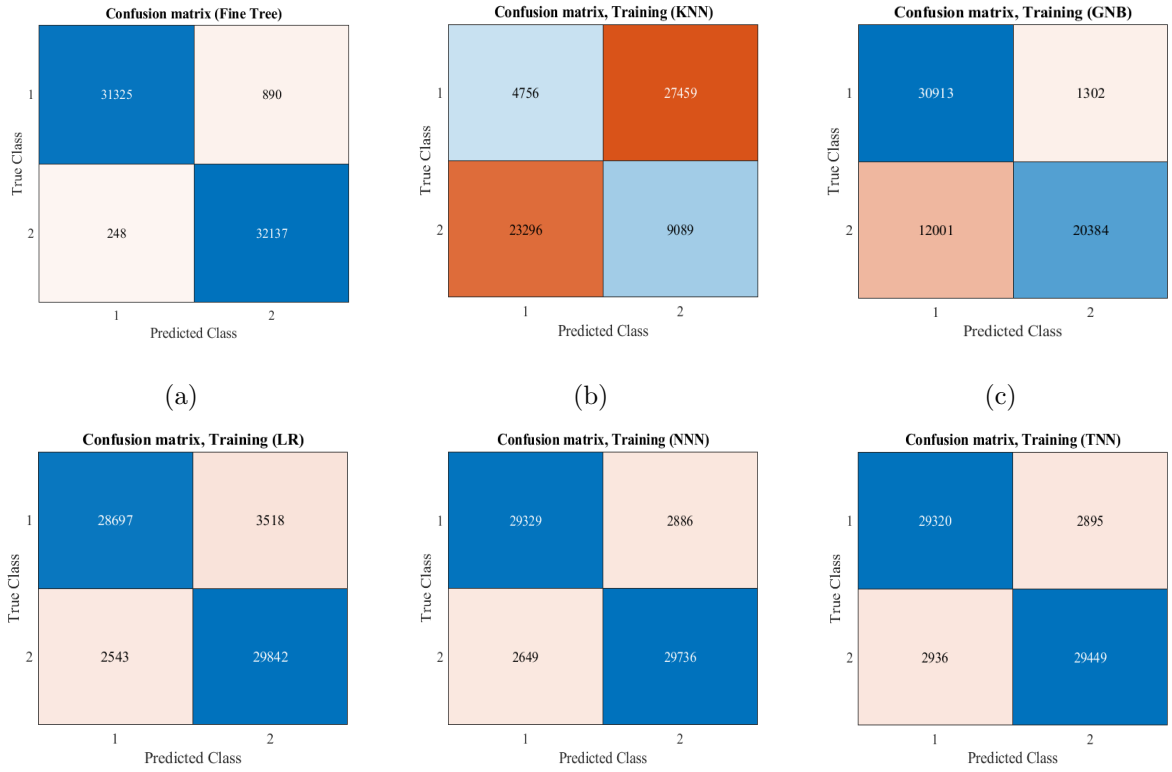


Figure 5.10: Training dataset confusion matrices for (a) Fine Tree, (b) Fine KNN, (c) Gaussian Naive Bayes, (d) Logistic Regression, (e) Narrow Neural Network, and (f) Trilayered Neural Network.

The confusion matrices in Figure 5.11 summarize the ML algorithm performance for the test data sets. The Fine Tree model demonstrates a good classification ability, with 5638 true positives for Class 1 and 5575 true positives for Class 2, suggesting a high accuracy in detecting accurate classifications. The misclassifications are 147 for Class 1 and 40 for

Class 2, indicating a low error rate, making the Fine Tree model very dependable for both classes. KNN performed significantly lower, with 391 true positives for Class 1 and 861 for Class 2. The higher misclassification, 5394 false positives for Class 1, emphasizes the model's difficulty in classifying classes. The GNB model has a better result, with 5223 true positives in Class 1 and 4933 in Class 2, with false positives and negatives of 562 and 682, respectively. Although not as good as the FT, the GNB achieves an acceptable performance between accurate classifications and errors. The LR model reveals 4967 true positives for Class 1 and 4974 for Class 2, with a larger number of false positives and negatives, specifically 818 false positives for Class 1. Furthermore, the NNN works well, with 5387 true positives for Class 1 and 5265 for Class 2, with slightly higher but still low misclassification counts of 398 and 350 for Class 1 and Class 2, respectively. This illustrates that the model takes a balanced approach to both classes. The TNN has 5311 true positives for Class 1 and 5201 for Class 2, with 474 and 414 mis-classifications, respectively. These results show that the TNN maintains strong predictive performance but slightly higher error rates than the Fine Tree model.

Figures 5.12 and 5.13 display confusion matrices for the training and test datasets, presenting essential metrics such as True Positive Rate (TPR) and False Negative Rate (FNR) to illustrate classification performance across various models. The Fine Tree model shows excellent accuracy in both datasets, with TPRs of 97.2% and 97.5% for Class 1 and 99.2% and 99.3% for Class 2 in training and testing datasets, respectively. Corresponding FNRs remain low at around 2.4% and 0.8% in training, with a slight increase to 2.5% and 0.7% in testing. In stark contrast, the KNN model struggles significantly, demonstrating extremely low TPRs of 14.8% and 6.8% for Class 1 and 28.1% and 15.3% for Class 2, alongside high FNRs, highlighting its poor performance in both datasets. The Gaussian Naive Bayes (GNB) model performs well for Class 1 with TPRs of 96.0% in training and 90.3% in testing but shows a preference for Class 1 as indicated by lower TPRs for Class 2. Logistic Regression (LR) exhibits strong predictive abilities, especially for Class 2 with

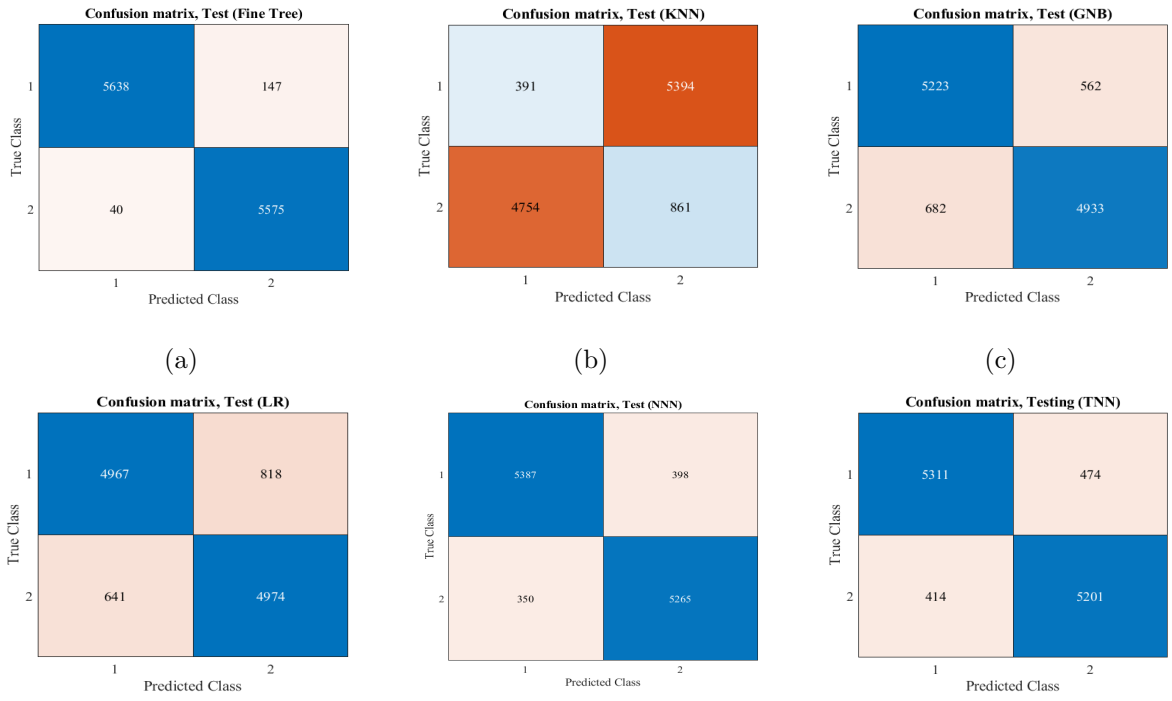


Figure 5.11: Testing dataset confusion matrices for (a) Fine Tree, (b) Fine KNN, (c) Gaussian Naive Bayes, (d) Logistic Regression, (e) Narrow Neural Network, and (f) Trilayered Neural Network.

TPRs of 88.6% in testing, despite a higher tendency to miss Class 1 instances. The Narrow Network model (NNN) maintains a solid balance with consistent TPRs of approximately 93% for both classes in testing, indicating effective generalization across datasets. Similarly, the TriLayered Neural Network (TNN) shows high and balanced TPRs, maintaining performance with minimal bias between the classes across the training and testing phases.

Figure 5.14 shows confusion matrices in a percentage format, including Positive Predictive Value (PPV) and False Discovery Rate (FDR) for the training dataset. The FT has an outstanding PPV of 99.2% for Class 1 and 97.3% for Class 2, with extremely low FDRs of 0.8% and 2.7%, respectively. The KNN model has significantly worse performance. The GNB model has a PPV of 72.0% for Class 1 and 94.0% for Class 2, indicating better performance in correctly recognizing Class 2 cases. However, the FDR of 28.0% for Class 1 suggests a high

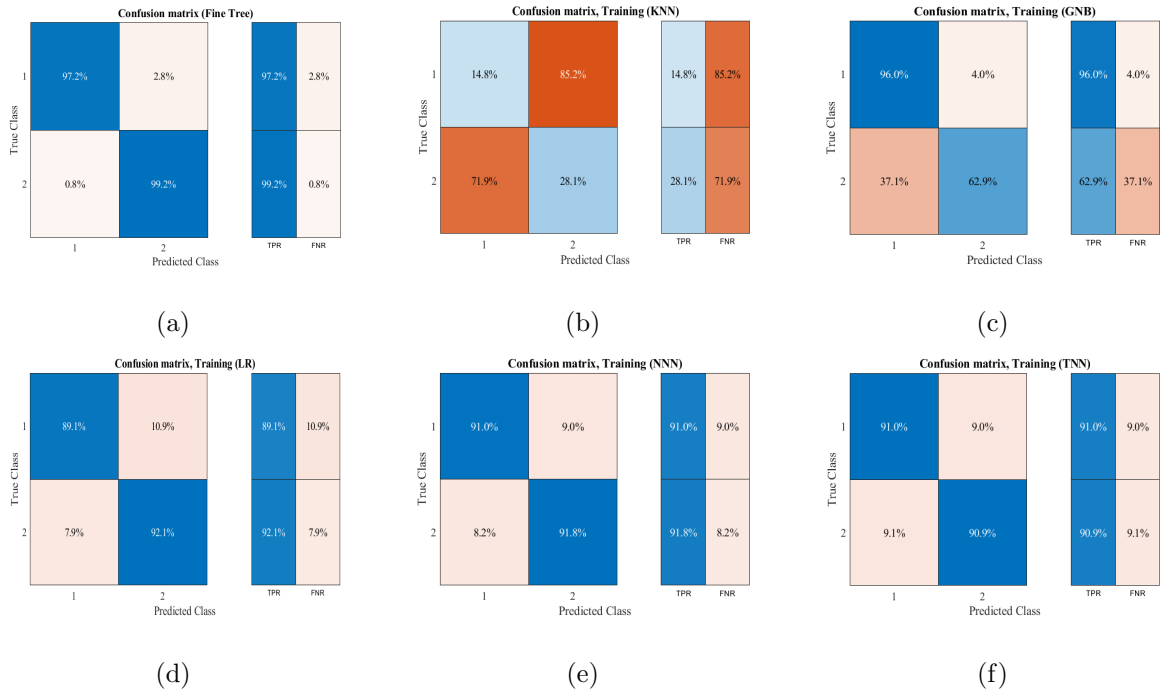
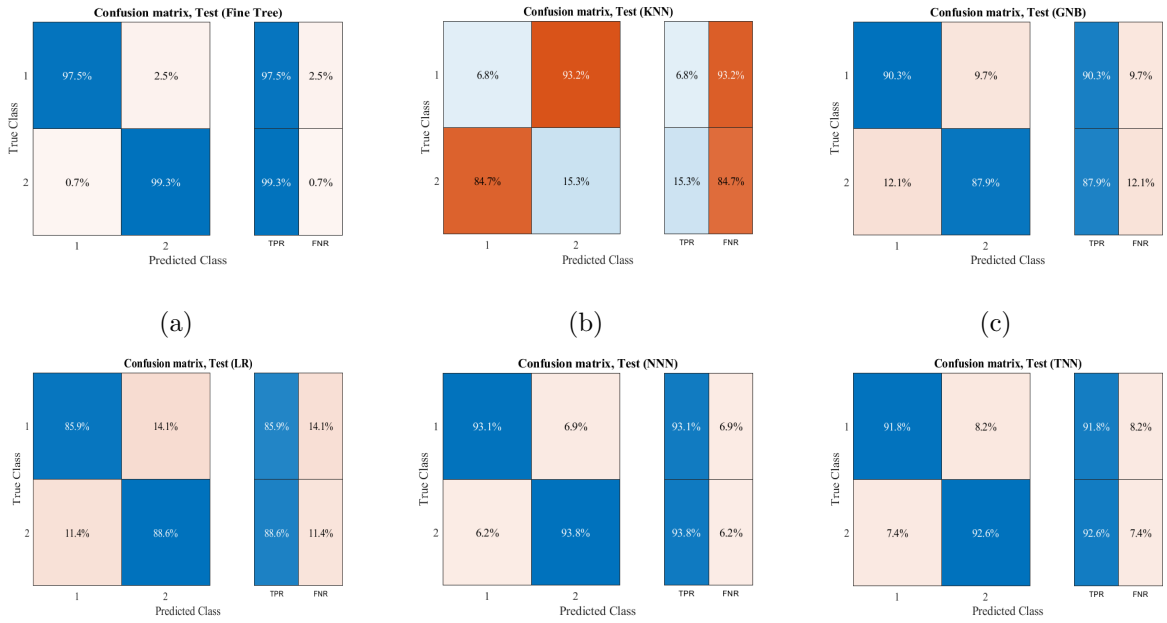


Figure 5.12: Confusion matrices representing the classification performance of machine learning models on the training dataset, showing the True Positive Rate (TPR) and False Negative Rate (FNR) for each class. Models include the (a) Fine Tree, (b) Fine KNN, (c) Gaussian Naive Bayes, (d) Logistic Regression, (e) Narrow Neural Network, and (f) Trilayered Neural Network.

percentage of false detections. Furthermore, the LR model has a high PPV value of 91.9% for Class 1 and 89.5% for Class 2, indicating that it can predict positive cases accurately. Nevertheless, the FDR for Class 2 is 10.5%, implying a significantly greater probability of false positives in Class 2 predictions compared to 8.1% for Class 1. The NNN model has a slightly better PPV for Class 1. The FDR rates are somewhat lower for Class 1. The results from TNN show that the model is equally capable of predicting both classes with the same degree of precision and misclassification rates.

Figure 5.15 shows the similar confusion matrices as Figure 5.14 for the testing dataset. The FT model performs well in the test dataset as well, with a PPV of 99.3% for Class 1 and 97.4% for Class 2 and very low FDR rates of 0.7% for Class 1 and 2.6% for Class 2. KNN, like earlier results, receives a significantly lower PPV of 7.6% for Class 1 and 13.8% for Class



(a) (b) (c)

(d) (e) (f)

Figure 5.13: Confusion matrices representing the classification performance of machine learning models on the testing dataset, showing the True Positive Rate (TPR) and False Negative Rate (FNR) for each class. Models include the (a) Fine Tree, (b) Fine KNN, (c) Gaussian Naive Bayes, (d) Logistic Regression, (e) Narrow Neural Network, and (f) Trilayered Neural Network.

2 and high FDRs of 92.4% and 86.2%, respectively. Additionally, the GNB model shows a higher PPV for Class 2 at 89.3% compared to 85.5% for Class 1. The FDR for GNB is 11.5% for Class 1 and 10.2% for Class 2, showing that it performs better at Class 2 predictions while producing a moderate number of false detections. LR has a lower PPV of 88.6% for Class 1 and 85.9% for Class 2, with FDRs of 11.4% for Class 1 and 14.1% for Class 2. This suggests that, while LR is effective, it is more likely to produce false positives, especially for Class 2 predictions. The confusion matrix of NNN demonstrates a PPV of 93.9% for Class 1 and 93.0% for Class 2, demonstrating an ability for RFI detection but with a little higher FDR for Class 2 (7%) than Class 1 (6.1%). TNN results exhibit comparable performance and indicate a slightly higher misclassification rate for predicting Class 2.

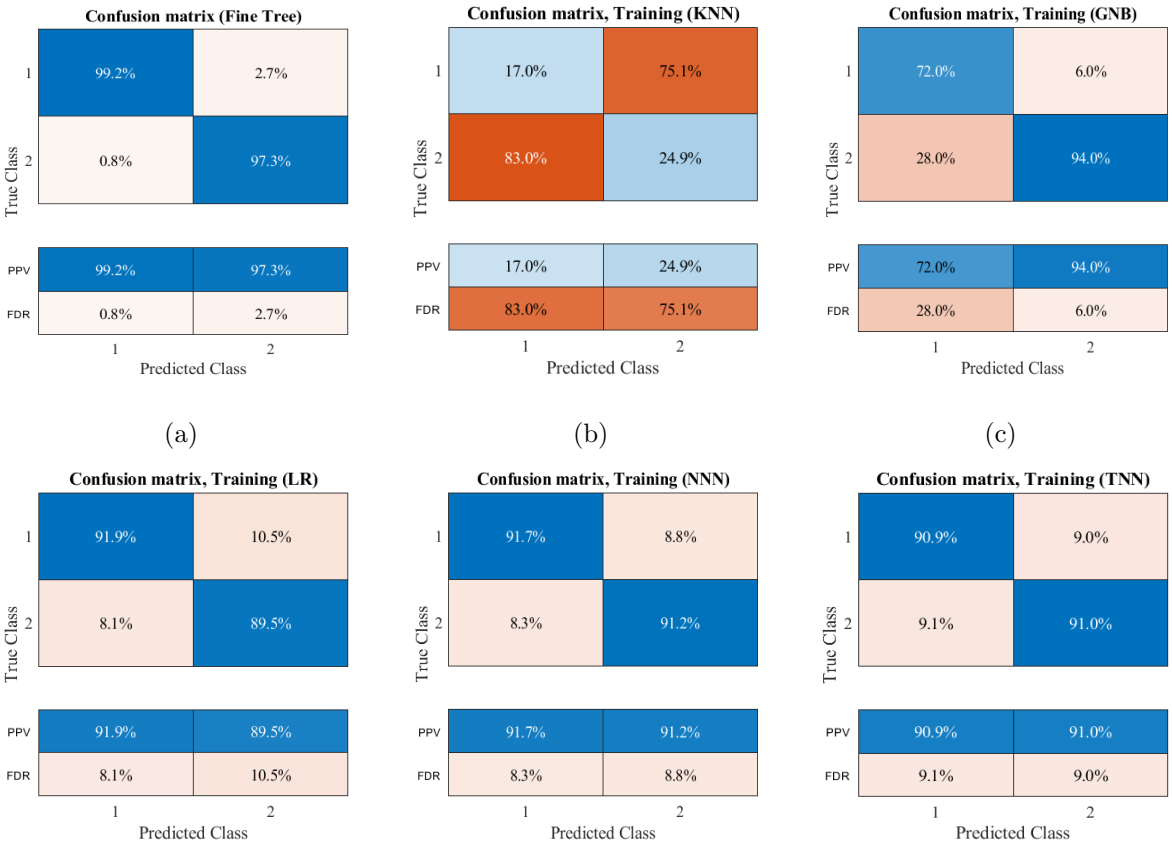


Figure 5.14: Confusion matrices representing the classification performance of machine learning models on the training dataset, showing the Positive Predictive Value (PPV) and False Discovery Rate (FDR) for each class. Models include the (a) Fine Tree, (b) Fine KNN, (c) Gaussian Naive Bayes, (d) Logistic Regression, (e) Narrow Neural Network, and (f) Trilayered Neural Network.

Figure 5.16 displays the Receiver Operating Characteristic (ROC) curves for the models in the test dataset. The Fine Tree has a remarkable Area Under the Curve (AUC) of 0.9923, indicating excellent classification performance. KNN has a lower AUC of 0.1105 than the other models, implying this dataset's worst classification performance. GNB achieves an AUC of 0.947, showing that it is a competent classifier. The LR model, with an AUC of 0.955, shows strong predictive performance despite being slightly lower than NN. The NNN also has a high AUC of 0.973, the TNN has a robust AUC of 0.959.

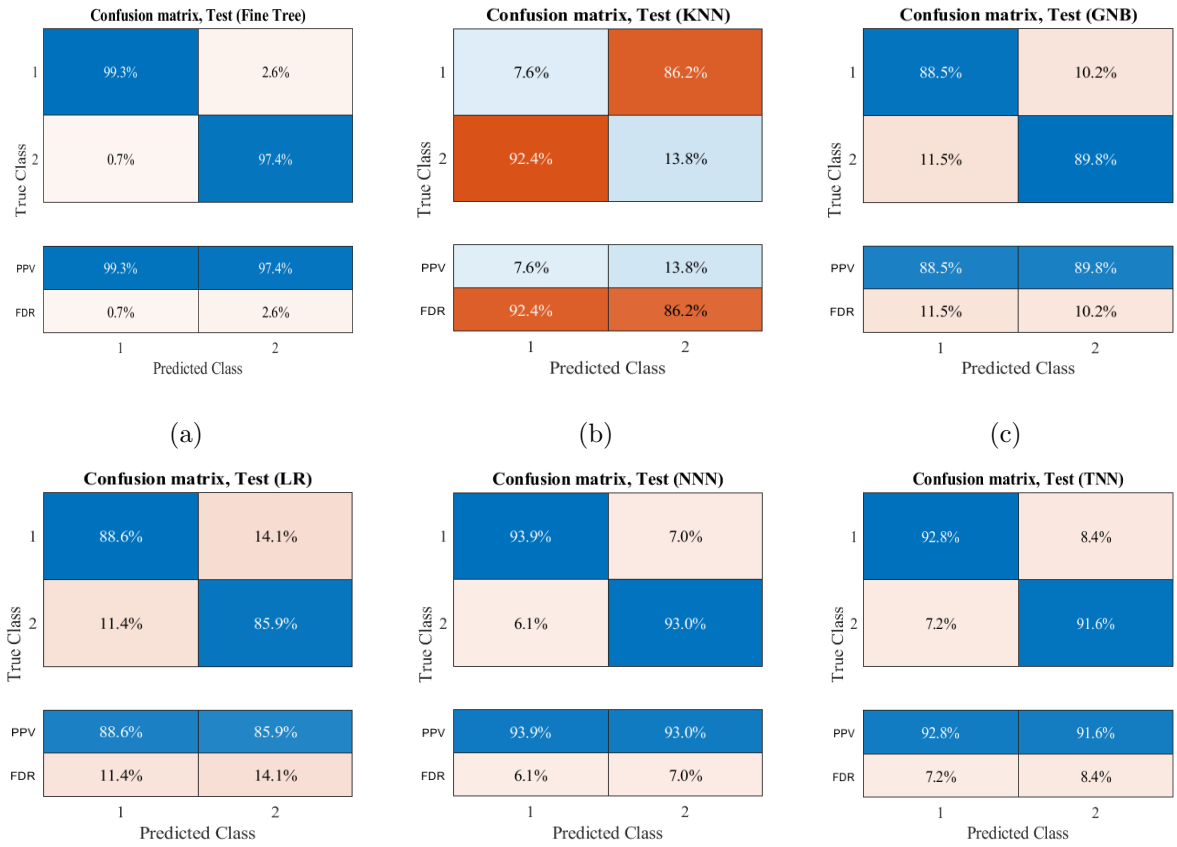


Figure 5.15: Confusion matrices representing the classification performance of machine learning models on the testing dataset, showing the Positive Predictive Value (PPV) and False Discovery Rate (FDR) for each class. Models include the (a) Fine Tree, (b) Fine KNN, (c) Gaussian Naive Bayes, (d) Logistic Regression, (e) Narrow Neural Network, and (f) Trilayered Neural Network.

5.4.2 Results of Altitude Estimation

Once the 5G-RFI is detected in the radar signals, the interfered signals proceed to a second stage involving regression ML models, which use the I/Q signals from radar and the additional features to estimate the radar altitudes. Altitudes are calculated by dividing the sampled FMCW signal recording into down and up frequency sweeps, creating unique models for each direction, and then averaging the results of the radar signals' up and down sweeps. This approach ensures models are finely tuned to the specific characteristics of each

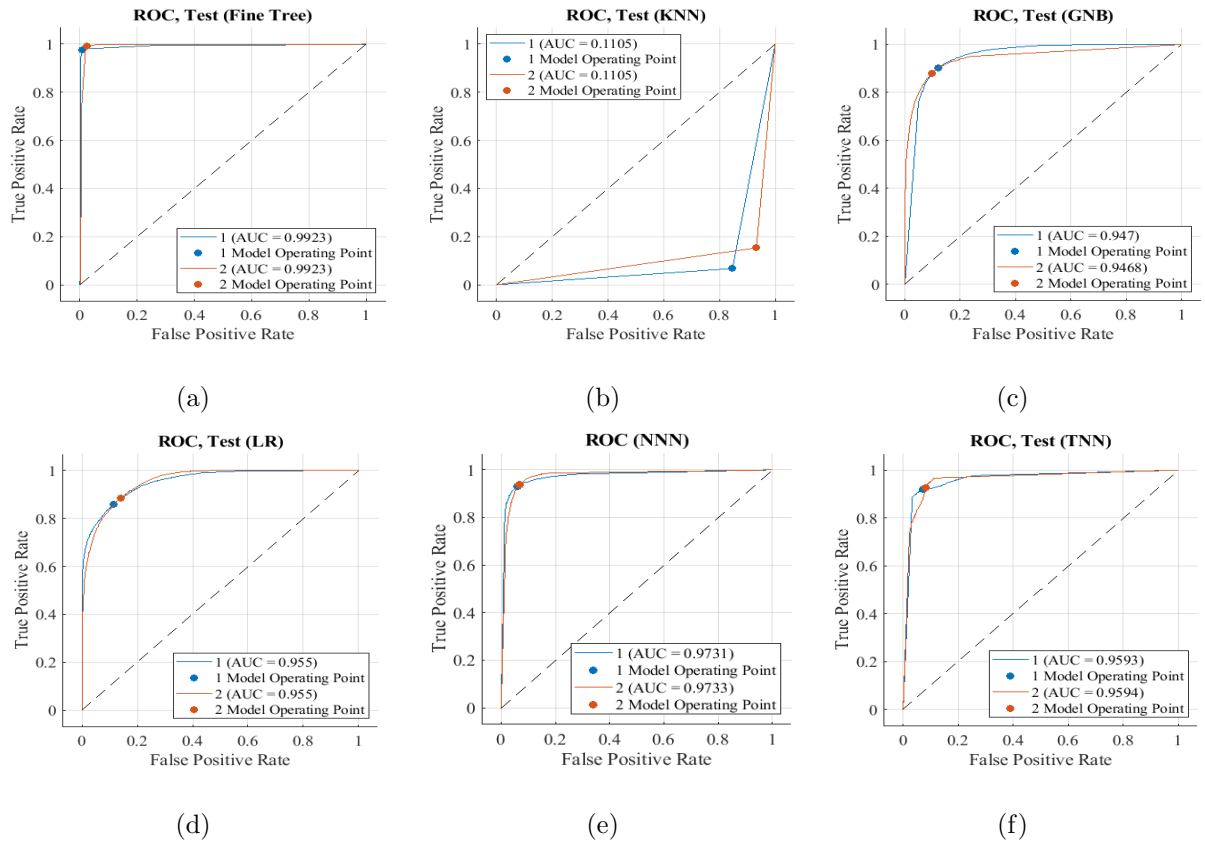


Figure 5.16: Receiver Operating Characteristic (ROC) curves for various ML algorithms evaluated with the testing dataset, with Area Under the Curve (AUC) metrics. Models displayed include the (a) Fine Tree, (b) Fine KNN, (c) Gaussian Naive Bayes, (d) Logistic Regression, (e) Narrow Neural Network, and (f) Trilayered Neural Network.

sweep direction. Note that the results for not using ML algorithms for mitigation are only briefly listed in the following sections since they are unacceptable for navigational functions ($>$ 500 meters in MAE). The purpose is to examine the improvements in altitude estimations through ML processing. Again, the training and testing data sets comprise RFI levels from low to extreme that are randomly mixed.

5.4.2.1 Up-Sweep Cases

Table 5.4 compares the performance of altitude estimations for training and testing datasets. The units for RMSE and MAE are in meters; the units for MSE are m^2, and R^2 and Corr

are unitless metrics. The Fine Tree (FT) model has a significant advantage with a validation RMSE of 39.0623 m and a slightly higher R-squared of 0.9909 over the Medium Tree (MT). The Coarse Tree (CT) model's validation RMSE of 40.2454 shows a balance between model simplicity and accuracy in variance capture. The Bagged Tree model shows how ensemble methods may improve prediction, as indicated by its excellent stability (validation RMSE = 24.6151 m, test RMSE = 24.1991 m). It shows the benefit of aggregating multiple models to minimize variance and increase generalization. Linear Regression (LR)'s advantage of having a clear model structure is offset by its relatively high RMSE. The Wide Neural Network (WNN), with lower RMSE scores than the Narrow Neural Network (NNN), demonstrates the benefits of employing many neurons in the network for better variance modeling. The Medium Neural Network (MNN) achieves an excellent mix of depth and variance, while the Tri-layered Neural Network (TriNN) outperforms the Bi-layered Neural Network (BiNN). The Boosted Tree, with a larger RMSE of 49.9384 m, still presents a solid R-squared of 0.9851and shows its ability to represent variance at the expense of more complexity.

Table 5.4: Comparative Performance of Machine Learning Models for Altitude Prediction from Up Sweep Radar Signals.

Model Type	RMSE (Train) (m)	MSE (Train) (m^2)	R^2 (Train)	MAE (Train) (m)	Corr (Train)	MAE (Test) (m)	MSE (Test) (m^2)	RMSE (Test) (m)	R^2 (Test)	Corr (Test)
Fine Tree	39.1	1525.86	0.9909	18.0	0.9996	18.1349	1635.4	40.4	0.9905	0.9953
Medium Tree	39.4	1550.753	0.9908	18.9	0.9989	17.5095	1386.3	37.3	0.992	0.996
Coarse Tree	40.2	1619.69	0.9903	20.2	0.9976	19.6519	1595.8	39.4	0.9907	0.9954
Boosted tree	49.9	2493.848	0.9851	42.6	0.9975	42.3625	2530.1	50.3	0.9853	0.9968
Bagged tree	24.6	605.9048	0.9964	12.6	0.9996	12.1279	585.6	24.2	0.9966	0.9983
LR	104.8	10982.02	0.9345	74.7	0.9889	70.3131	9348.6	96.7	0.9458	0.9728
NNN	54.4	2955.964	0.9824	40.7	0.9954	38.3673	2559.1	50.6	0.9852	0.9926
MNN	47.0	2210.816	0.9868	35.3	0.9989	31.289	1809.7	42.5	0.9895	0.9947
WNN	43.9	1934.249	0.9885	32.2	0.9982	30.9583	1712.4	41.4	0.9901	0.995
BiNN	52.3	2738.892	0.9837	39.2	0.9973	35.497	2265.7	47.6	0.9869	0.9934
TriNN	43.6	1900.082	0.9887	32.5	0.9984	29.1337	1613.4	40.2	0.9906	0.9953

Figure 5.17 shows scatter density plots with color gradients for the testing dataset based on up-sweep signals. The Bagged Tree (BT) model's graphic displays dense clusters along the line of equality, with minor deviations reflected by cooler colors. Again, the FT plot shows a similar behavior. The NNN and TNN scatter plots show larger color density distributions. While the major concentration remains along the line of equality.

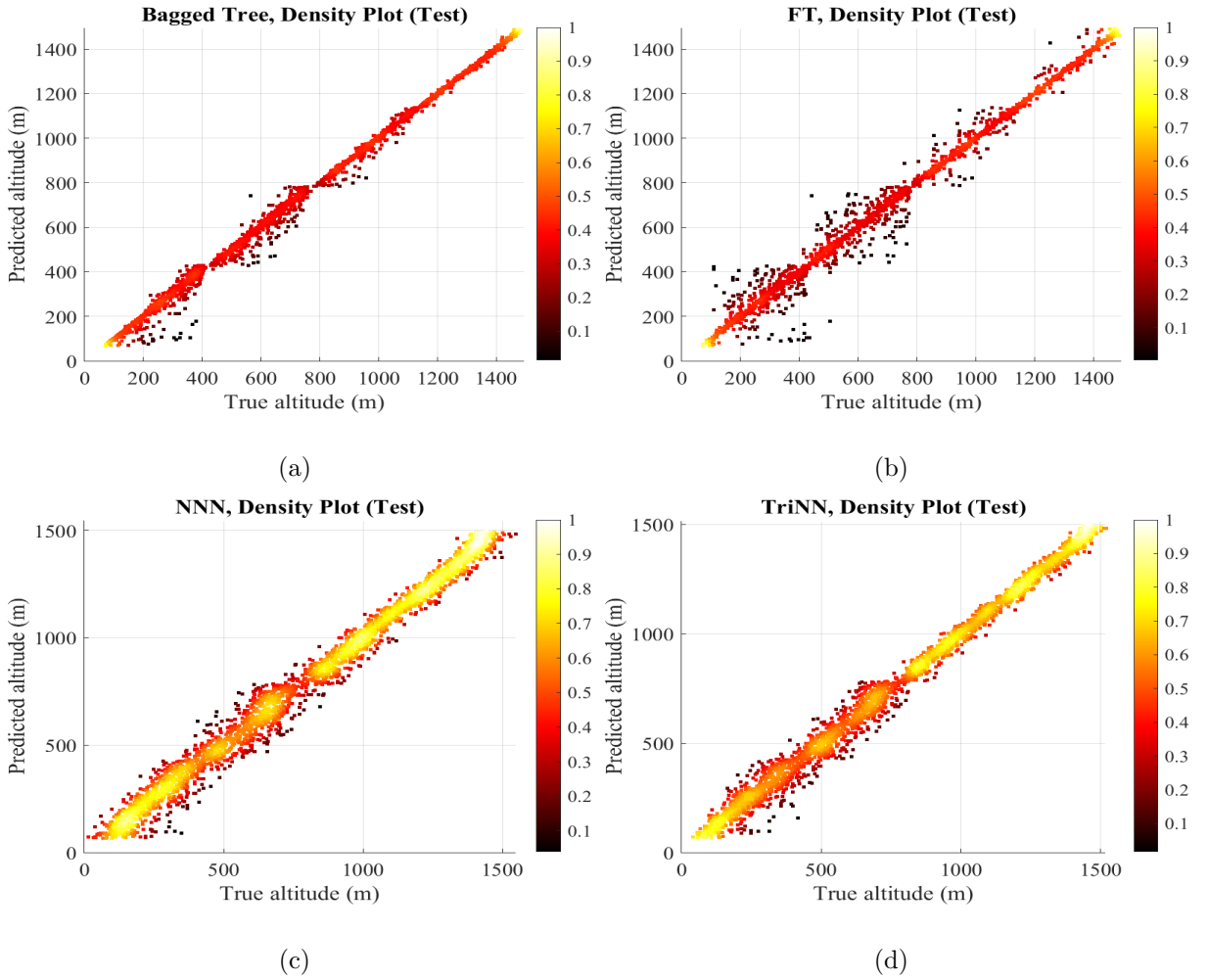


Figure 5.17: Comparative scatter plots with color gradients for Altitude Prediction (in meters) on up-sweep Test data of the following models: (a) Bagged Tree, (b) Fine Tree, (c) Narrow neural network, and (d) TriLayred neural network.

5.4.2.2 Down Sweep Case

Table 5.5 evaluates the altitude estimation performance based on down-sweep radar signals. The Fine Tree (FT) model has a notable prediction accuracy with an RMSE of 94.3654 m during validation, suggesting a small decrease in accuracy compared to its up-sweep performance. This trend is also reflected in its test results, with an RMSE of 90.2243 m, which, though greater than the up-sweep analysis, still demonstrates significant improvement from non-mitigation results. The Medium and Coarse Tree models support the trend that greater simplicity in the tree model's structure leads to improved management of the down sweep signal variance. This is especially obvious in the Coarse Tree's higher performance on test data, which has an RMSE of 79.4652 m. The Bagged Tree (BT) model outperforms its peers, and Linear Regression (LR) reveals its decreased effectiveness. This issue becomes more apparent when comparing its performance on down and up sweeps. The NNN achieved an RMSE of 83.9502 on validation, indicating a noticeable but insignificant decrease from its up-sweep analysis. The WNN and the TriNN both perform excellently consistently.

Figure 5.18 depicts scatter plots with color gradients from the testing dataset, demonstrating the predictive ability of four models when applied to down-sweep radar signals. The Bagged Tree model has a dense region of warm colors that matches the line of perfect prediction, indicating a good connection between actual and predicted responses. This plot shows that the model is good at generalizing to new, unseen data. The FT model produces a slightly more dispersed pattern, particularly at higher response values. Nevertheless, most data points are near the ideal prediction line, consistent with their high-performance scores. Similar to the previous figures, the LR plot exhibits a broader dispersion of data points along the response values. The spreading of warmer colors into cooler ones, particularly in the lower and higher ends of the response spectrum, indicates the model's poorer accuracy, consistent with the larger RMSE and lower R-squared values from the results table. Finally,

Table 5.5: Comparative Performance of Machine Learning Models for Altitude Prediction from Down Sweep Radar Signals.

Model Type	RMSE (Train) (m)	MSE (Train) (m^2)	R^2 (Train)	MAE (Train) (m)	Corr (Train)	MAE (Test) (m)	MSE (Test) (m^2)	RMSE (Test) (m)	R^2 (Test)	Corr (Test)
Fine Tree	94.4	8904.8	0.9469	70.5	0.9575	66.9	8140.4	90.2	0.9528	0.9596
Medium Tree	87.8	7715.3	0.954	66.1521	0.9673	64.7	7426.5	86.2	0.9569	0.9692
Coarse Tree	81.6	6654.6	0.9603	61.8912	0.9691	60.4	6314.7	79.5	0.9634	0.9703
Boosted tree	83.1	6911.4	0.9588	65.2022	0.976	65.5	7050.0	83.9	0.9591	0.9766
Bagged tree	68.4	4674.7	0.9721	52.3225	0.9764	51.3	4509.5	67.2	0.9739	0.9765
LR	161.9	26237.5	0.8435	114.5206	0.9162	109.6	23357.2	152.8	0.8646	0.9196
NNN	83.9	7047.6	0.958	64.1611	0.943	56.8	5658.2	75.2	0.9672	0.945
MNN	75.4	5685.9	0.9661	57.2915	0.9545	54.7	5050.0	71.1	0.9707	0.9558
WNN	72.9	5312.6	0.9683	54.328	0.9541	52.7	4754.7	68.9	0.9724	0.9553
BiNN	81.2	6593.8	0.9607	61.8341	0.9669	76.1	9655.9	98.3	0.944	0.9664
TriNN	63.5	4031.7	0.976	47.5646	0.957	45.8	3658.3	60.5	0.9788	0.9576

the TriNN produces a dense line of high-density predictions nearly matching the ideal diagonal. This pattern demonstrates the model's ability to capture the complex relationships in the data, resulting in a strong predictive performance as supported by its test metrics.

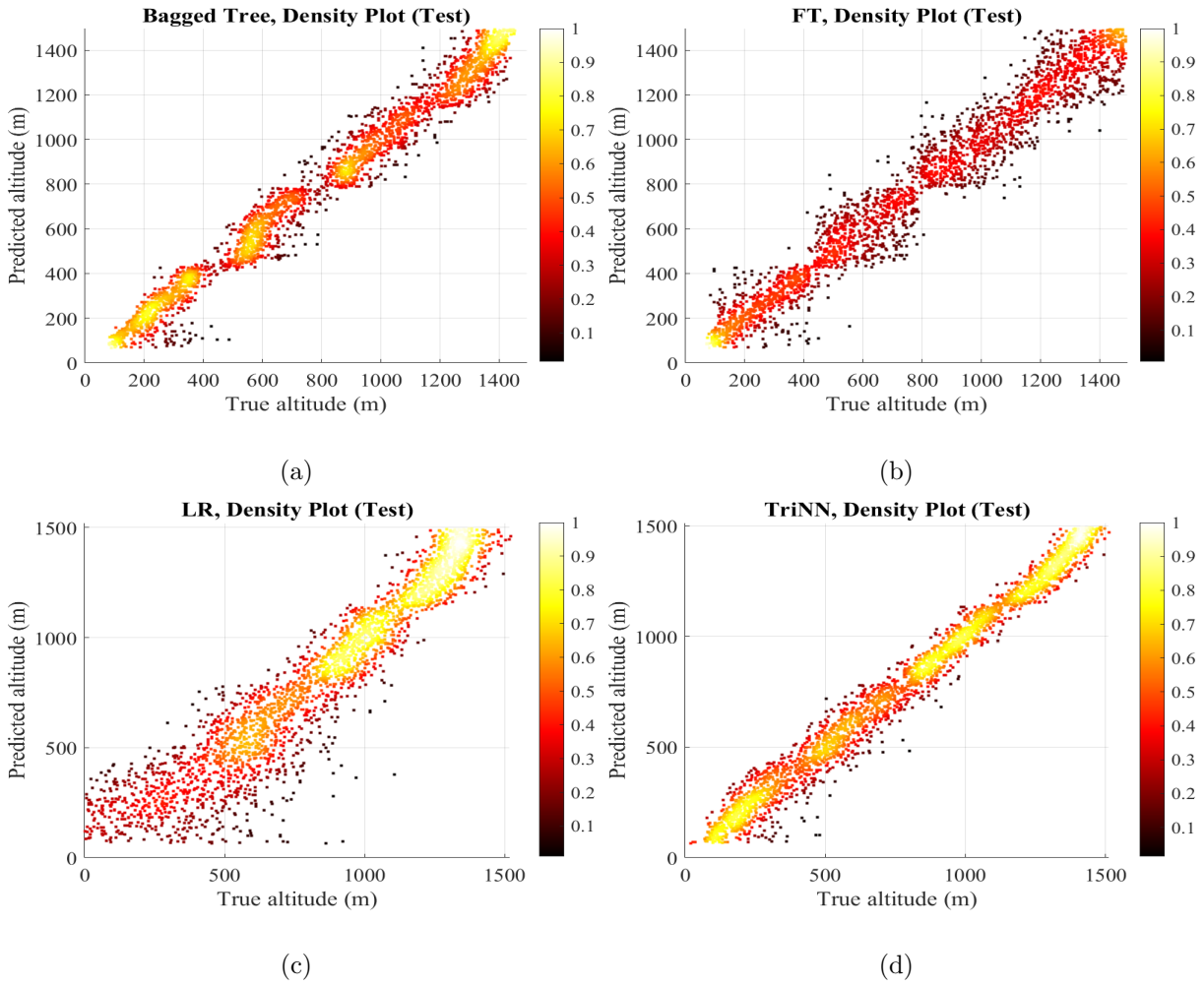


Figure 5.18: Comparative scatter plots with color gradients for Altitude Prediction (in meters) on down-sweep Test data of the following models: (a) Bagged Tree, (b) Fine Tree, (c) Linear regression, and (d) TriLayred neural network.

The summary of altitude estimation performance is provided in Table 5.6. It shows the performance of altitude estimation utilizing down-sweep and up-sweep and their average estimation compared to not applying any ML-based RFI mitigation techniques. For down sweep predictions using the TriNN model, validation metrics show a robust R-squared value of 0.976 and an MSE of 4031.706, indicating a strong model fit to the data without overfitting. This performance carries into the testing phase with an R-squared of 0.9788. In contrast, the best up-sweep model (Bagged Trees) outperforms the down-sweep with a notably higher

R-squared of 0.9964 in validation and 0.9966 in testing, along with lower RMSE values. The averaged altitude estimations based on optimal up and down sweep estimations produce the most balanced results during validation, with an R-squared of 0.99762 and the lowest MSE (398.2082) among the setups. In the testing phase, the average approach maintains high accuracy with an R-squared of 0.9907. However, it does not reach the accuracy of the best up-sweep predictions alone, evidenced by an RMSE of 40.0318. While this is an improvement over the down sweep's RMSE of 60.4839, averaging the results of multiple models may slightly reduce some of the model's strengths. The last row of the table, which represents altitude estimations derived directly from signals with 5G interference without applying any ML methods, shows severe performance degradation. The RMSE increases to 808.1 while the R-squared falls to -2.8, emphasizing the significant impact of 5G interference on measurement accuracy.

Table 5.6: Summary of Performance of Aircraft Altitude Estimation Under 5G RFI.

Model	**Training Results**					**Testing Results**				
	RMSE (m)	**MSE (m^2)**	**R^2**	**MAE (m)**	**Corr**	**RMSE (m)**	**MSE (m^2)**	**R^2**	**MAE (m)**	**Corr**
Best Down (TriNN)	63.5	4032	0.976	47.6	0.957	60.5	3658	0.979	45.8	0.958
Best Up (Bagged)	24.6	606	0.996	12.5	1.000	24.2	586	0.997	12.1	0.998
Mean	19.9	398	0.998	14.2	0.999	40.0	1603	0.991	29.5	0.995
With Interf.	821.3	674606	-3.02	553.8	0.440	808.1	653004	-2.80	552.2	0.445

Figure 5.19 presents altitude profiles for both the training and testing datasets, offering a detailed examination of model performance in altitude estimation using the altimeter signals with 5G RFI, up sweep signals, down sweep signals, and their average. The plots show that the up-sweep predictions align best with the truth data, as indicated by the red line, especially at higher altitudes. However, some variability in predictions emerges at lower altitudes. On the other hand, the down-sweep predictions display a slightly reduced accuracy, with notable variations across different altitudes and some peaks at lower altitudes. The averaged predictions further improved the accuracy, suggesting that integrating up and down-sweep predictions leads to a more precise estimation. For the result plots based on

testing datasets, although there is a slight increase in the variation of altitude predictions across all methods compared to the training dataset, the overall performance closely aligns with the training results. The up-sweep method achieves the best performance with the least variation in altitude predictions, while the down-sweep shows increased variations across different altitudes. The averaged predictions maintain a balance between the down-sweep and the up-sweep estimations. These observations suggest that while the down-sweep and average predictions offer a smoother predictive trend, they might not entirely address the overfitting or specific biases in the testing data environment. On the other hand, 5G interference significantly impacts altitude measurements, as seen in the pink-colored traces in training and testing datasets that consistently deviate from the true data.

5.5 Conclusions

This study addresses the issue of 5G base station interference with radar altimeters during approaching flights. A two-step machine learning (ML) approach is developed to detect RF interference and mitigate the impact of interference on altitude estimations. The method begins by classifying altimeter signals to distinguish between the "pure" and interfered states using computationally effective ML algorithms, then uses various regression models for altitude estimation. The study utilizes three types of data: real 5G signals from a Norman, Oklahoma base station, emulated FMCW radar altimeter signals during an airplane's landing process, and a combination of them to simulate radar altimeter signals with 5G interference. A novel approach estimates altitudes by models trained separately from up-sweep and down-sweep radar signals and then averages the estimations from these models. The performance results validate the two-step and dual-model approach: The Fine Tree (FT) model achieved 98.3596% accuracy in classification. In altitude prediction, the Bagged Tree (BT) model applied to up-sweep signals demonstrated a correlation coefficient of 0.9983. The best results for down-sweep signals were obtained with TriLayred Neural Network (TriNN), achieving a

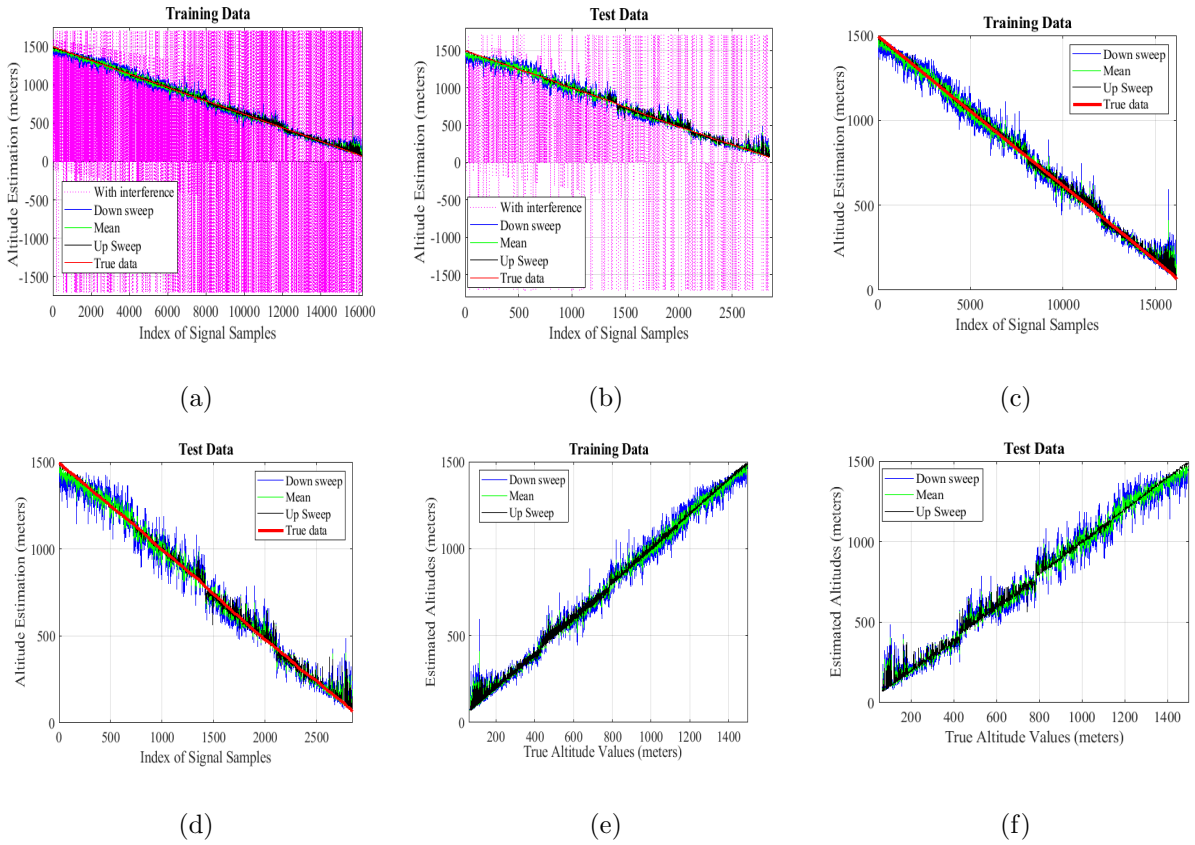


Figure 5.19: Comparative analysis of altitude estimations and actual vs. predicted altitudes: (a) Training dataset including the impact of 5G interference on altitude predictions, (b) Testing dataset including the impact of 5G interference on altitude predictions, (c) Training dataset without the impact of 5G interference, (d) Testing dataset without the impact of 5G interference, (e) Actual vs. predicted altitudes from downs-weep, upsweep, and their mean for the training dataset without the impact of 5G interference, and (f) Actual vs. predicted altitudes from down-sweep, upsweep, and their mean for the testing dataset without the impact of 5G interference.

correlation coefficient of 0.9576. Future work should focus on incorporating real-time ML processing to allow for dynamic interference mitigation during flight. Additionally, expanding the dataset to include more realistic 5G signal collections and radar measurements across a wider range of flight conditions would further strengthen the robustness of the models. Future studies should also explore advanced ML algorithms and hybrid methods to handle

more complex interference patterns, potentially improving both the accuracy and efficiency of interference mitigation in radar systems. By doing so, this approach holds promise for broader adoption in aviation safety systems amid the global expansion of 5G networks.

Chapter 6

Conclusions and Future Work

This dissertation explored integrating machine learning (ML) techniques with radar applications to address atmospheric sciences and aviation challenges. Through a series of studies, this research demonstrated the potential of ML to enhance atmospheric measurements, improve cloud physics profiling, and mitigate 5G interference in radar sensing. In conclusion, this dissertation has demonstrated the substantial benefits of integrating machine learning with radar technology as a foundation for exploring atmospheric science and aviation applications.

6.1 Key Findings and Contributions

6.1.1 Chapter 2: Theoretical Foundations of Machine Learning Methods

This chapter provided an overview of the evolution of ML and its role in scientific research. It discussed various types of ML, including supervised, unsupervised, and reinforcement learning, and examined common ML models like regression and classification. The chapter also introduced the evaluation metrics for regression and classification used throughout the dissertation, such as RMSE, MSE, and R-squared for regression and accuracy and precision

for classification. Additionally, the chapter provided an overview of metaheuristic algorithms in optimizing objective functions.

6.1.2 Chapter 3: Atmospheric Humidity Estimation from Wind Profiler Radar

Chapter 3 developed and validated a novel ML approach for estimating atmospheric relative humidity (RH) profiles using wind profiler radar data. The study introduced a cascaded ML framework that used only Doppler moment data, avoiding reliance on temperature. An intermediate pressure estimation stage improved RH prediction accuracy across different seasons. This chapter demonstrated the significant potential of ML techniques to refine atmospheric measurements.

The findings of this work have been published in the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS) under the title "Atmospheric Humidity Estimation From Wind Profiler Radar Using a Cascaded Machine Learning Approach" (Amaireh et al. 2023a).

6.1.3 Chapter 4: Improving Cloud Liquid Water Content Profiling

Chapter 4 enhanced Cloud Liquid Water Content (CLWC) profiling using novel ML techniques. By cross-validating space-based remote sensing (ERA5) data with high-precision radiosonde observations and applying a metaheuristic algorithm for data cleansing, the study improved the correlation between input features (such as pressure, temperature, relative humidity, and specific humidity) and CLWC. Various ML models, including decision trees, ensemble trees, and neural networks, were used to predict the CLWC profiles, with decision tree and ensemble models outperforming the others. This chapter demonstrated the potential for global improvements in weather prediction models.

The findings of this work have been published as a preprint and are currently under review in the Journal of Atmospheric and Oceanic Technology (JTECH) under the title "A Novel Approach for Improving Cloud Liquid Water Content Profiling with Machine Learning" (Amaireh et al. 2023b).

6.1.4 Chapter 5: Identification and Mitigation of 5G Interference for Radar Altimeters

As an application for aviation safety, Chapter 5 addressed the issue of interference from 5G base stations with radar altimeter signals. The study developed and applied an ML framework to identify and mitigate the effects of 5G interference by categorizing signals as pure or interfered and employing regression models to predict altitudes when interference was present. The study demonstrated the framework's robustness in real-world conditions using real 5G signals collected from base stations with Software-Defined Radio (SDR) and emulated radar signals. The findings of this chapter are crucial for ensuring aviation safety amid the rise of 5G technology.

The research has been published in IEEE Access under the title "Novel Machine Learning-Based Identification and Mitigation of 5G Interference for Radar Altimeters" (Amaireh and Zhang 2024), as well as in two conference papers presented at SPIE conferences: "Investigation of potential 5G RF interference with C-band radar operations and mitigation solutions", and "Improved investigation of electromagnetic compatibility between radar sensors and 5G-NR radios" (Amaireh et al. 2024; Amaireh and Zhang 2023).

6.2 Future works

Future research would explore more advanced ML models and techniques, integrating real-time data processing and applying these methodologies to other atmospheric and environmental measurements. For example, as telecommunications technology evolves, further investigation into the long-term impacts of 5G interference on radar systems and the development of more sophisticated mitigation strategies will be necessary. Enhanced data integration, combining data from radar systems and other sensors such as satellite and ground-based observations, could create more comprehensive datasets for training ML models. This could improve the accuracy and reliability of predictions in diverse atmospheric conditions. Developing adaptive and online learning algorithms that update ML models in real-time as new data becomes available would allow for continuous improvement in prediction accuracy and adaptability to changing conditions. Extending the ML and radar integration framework to interdisciplinary applications, such as environmental monitoring, disaster management, and climate change studies, could leverage improved data accuracy and predictive capabilities. Exploring advanced ML techniques, such as reinforcement learning, generative adversarial networks (GANs), and transfer learning, could further enhance radar systems' capabilities, addressing complex problems and improving model performance.

Bibliography

Adab, H., K. D. Kanniah, K. Solaimani, and K. P. Tan, 2013: Estimating atmospheric humidity using modis cloud-free data in a temperate humid region. *2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS*, IEEE, 1827–1830.

Albergel, C., E. Dutra, S. Munier, J.-C. Calvet, J. Munoz-Sabater, P. de Rosnay, and G. Balsamo, 2018: Era-5 and era-interim driven isba land surface model simulations: which one performs better? *Hydrology and Earth System Sciences*, **22 (6)**, 3515–3532.

Alishouse, J. C., J. B. Snider, E. R. Westwater, C. T. Swift, C. S. Ruf, S. A. Snyder, J. Vongsathorn, and R. R. Ferraro, 1990: Determination of cloud liquid water content using the ssm/i. *IEEE transactions on geoscience and remote sensing*, **28 (5)**, 817–822.

Alpaydin, E., 2020: *Introduction to machine learning*. MIT press.

Amaireh, A., A. S. Al-Zoubi, and N. I. Dib, 2019: Sidelobe-level suppression for circular antenna array via new hybrid optimization algorithm based on antlion and grasshopper optimization algorithms. *Progress In Electromagnetics Research C*, **93**, 49–63.

Amaireh, A. and Y. Zhang, 2024: Novel machine learning-based identification and mitigation of 5g interference for radar altimeters. *IEEE Access*, **12**, 102 425–102 439, doi: 10.1109/ACCESS.2024.3432833.

Amaireh, A., Y. Zhang, and P. Chan, 2023a: Atmospheric humidity estimation from wind profiler radar using a cascaded machine learning approach. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Amaireh, A., Y. R. Zhang, P. W. Chan, and D. Zrnic, 2023b: A novel approach for improving cloud liquid water content profiling with machine learning. *SSRN Electronic Journal*, doi: 10.2139/ssrn.4744644, URL `https://ssrn.com/abstract=4744644`.

Amaireh, A., Y. R. Zhang, D. J. Xu, and D. Bate, 2024: Improved investigation of electromagnetic compatibility between radar sensors and 5g-nr radios. *Radar Sensor Technology XXVIII*, SPIE, Vol. 13048, 85–98.

Amaireh, A. A., A. S. Al-Zoubi, and N. I. Dib, 2020a: The optimal synthesis of scanned linear antenna arrays. *International Journal of Electrical and Computer Engineering (IJECE)*, **10 (2)**, 1477–1484.

Amaireh, A. A., A. S. Al-Zoubi, and N. I. Dib, 2023c: A new hybrid optimization technique based on antlion and grasshopper optimization algorithms. *Evolutionary Intelligence*, **16 (4)**, 1383–1422.

Amaireh, A. A., N. I. Dib, and A. S. Al-Zoubi, 2020b: The optimal synthesis of concentric elliptical antenna arrays. *International Journal of Electronics*, **107 (3)**, 461–479.

Amaireh, A. A., N. I. Dib, and A. S. Al-Zoubi, 2022: Synthesis of new antenna arrays with arbitrary geometries based on the superformula. *International Journal of Electrical and Computer Engineering (IJECE)*, **12 (6)**, 6228–6238.

Amaireh, A. A. and Y. R. Zhang, 2023: Investigation of potential 5g rf interference with c-band radar operations and mitigation solutions. *Radar Sensor Technology XXVII*, SPIE, Vol. 12535, 206–217.

Amar, M. N., A. J. Ghahfarokhi, and C. S. W. Ng, 2022a: Predicting wax deposition using robust machine learning techniques. *Petroleum*, **8 (2)**, 167–173.

Amar, M. N., A. J. Ghahfarokhi, C. S. W. Ng, and N. Zeraibi, 2021: Optimization of wag in real geological field using rigorous soft computing techniques and nature-inspired algorithms. *Journal of Petroleum Science and Engineering*, **206**, 109 038.

Amar, M. N., H. Ouaer, and M. A. Ghriga, 2022b: Robust smart schemes for modeling carbon dioxide uptake in metal- organic frameworks. *Fuel*, **311**, 122 545.

Amar, M. N., M. Shateri, A. Hemmati-Sarapardeh, and A. Alamatsaz, 2019: Modeling oil-brine interfacial tension at high pressure and high salinity conditions. *Journal of Petroleum Science and Engineering*, **183**, 106 413.

Arce, G. R., 2005: *Nonlinear signal processing: a statistical approach*. John Wiley & Sons.

Arulkumaran, K., M. P. Deisenroth, M. Brundage, and A. A. Bharath, 2017: Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, **34 (6)**, 26–38.

Barreca, A. I., 2012: Climate change, humidity, and mortality in the united states. *Journal of Environmental Economics and Management*, **63 (1)**, 19–34.

Benveniste, J., 2011: Radar altimetry: past, present and future. *Coastal Altimetry*, 1–17.

Besic, N., J. Figueras i Ventura, J. Grazioli, M. Gabella, U. Germann, and A. Berne, 2016: Hydrometeor classification through statistical clustering of polarimetric radar measurements: A semi-supervised approach. *Atmospheric Measurement Techniques*, **9 (9)**, 4425–4445.

Bianco, L., D. Cimini, F. S. Marzano, and R. Ware, 2005: Combining microwave radiometer and wind profiler radar measurements for high-resolution atmospheric humidity profiling. *Journal of Atmospheric and Oceanic Technology*, **22 (7)**, 949–965.

Bishop, C. M. and N. M. Nasrabadi, 2006: *Pattern recognition and machine learning*, Vol. 4. Springer.

Blanca, M. J., J. Arnau, D. López-Montiel, R. Bono, and R. Bendayan, 2013: Skewness and kurtosis in real data samples. *Methodology*.

Boris Atayants, V. E. V. P. S. S., Viacheslav Davydochkin, 2014: *Precision FMCW Short-Range Radar for Industrial Applications*. Artech House, Boston, MA.

Brajard, J., A. Carrassi, M. Bocquet, and L. Bertino, 2021: Combining data assimilation and machine learning to infer unresolved scale parametrization. *Philosophical Transactions of the Royal Society A*, **379 (2194)**, 20200 086.

Breiman, L., 1996: Bagging predictors. *Machine learning*, **24**, 123–140.

Breiman, L., 2001: Random forests. *Machine learning*, **45**, 5–32.

Brun, E., 1989: Investigation on wet-snow metamorphism in respect of liquid-water content. *Annals of glaciology*, **13**, 22–26.

Cai, X., M. Giallorenzo, and K. Sarabandi, 2021: Machine learning-based target classification for mmw radar in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, **6 (4)**, 678–689.

Chen, L.-C. and A. A. Bradley, 2006: Adequacy of using surface humidity to estimate atmospheric moisture availability for probable maximum precipitation. *Water resources research*, **42 (9)**.

Chen, T. and C. Guestrin, 2016: Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 785–794.

Chowdhury, M., S. Kiraga, M. N. Islam, M. Ali, M. N. Reza, W.-H. Lee, and S.-O. Chung, 2021: Effects of temperature, relative humidity, and carbon dioxide concentration on growth and glucosinolate content of kale grown in a plant factory. *Foods*, **10 (7)**, 1524.

Chu, X., W. Bai, Y. Sun, W. Li, C. Liu, and H. Song, 2022: A machine learning-based method for wind fields forecasting utilizing gnss radio occultation data. *IEEE Access*, **10**, 30 258–30 273.

Coleman, R., 2001: Satellite altimetry and earth sciences: A handbook of techniques and applications. Wiley Online Library.

Cortes, C. and V. Vapnik, 1995: Support-vector networks. *Machine Learning*, **20 (3)**, 273–297.

Costa, V. G. and C. E. Pedreira, 2023: Recent advances in decision trees: An updated survey. *Artificial Intelligence Review*, **56 (5)**, 4765–4800.

Davis, R. E., G. R. McGregor, and K. B. Enfield, 2016: Humidity: A review and primer on atmospheric moisture and human health. *Environmental research*, **144**, 106–116.

Deb, K., 2011: Multi-objective optimisation using evolutionary algorithms: an introduction. *Multi-objective evolutionary optimisation for product design and manufacturing*, Springer, 3–34.

Domingos, P., 2015: *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. Basic Books.

Dong, C., F. Weng, and J. Yang, 2022: Assessments of cloud liquid water and total precipitable water derived from fy-3e mwts-iii and noaa-20 atms. *Remote Sensing*, **14 (8)**, 1853.

Easterling, D. R., et al., 2017: Precipitation change in the united states.

Eccel, E., 2012: Estimating air humidity from temperature and precipitation measures for modelling applications. *Meteorological Applications*, **19 (1)**, 118–128.

Electronics, B., 2023: Filtering the way to safer 5g near airports. `https://www.bench.com/setting-the-benchmark/filtering-the-way-to-safer-5g-near-airports`.

Fracastoro, G., E. Magli, G. Poggi, G. Scarpa, D. Valsesia, and L. Verdoliva, 2021: Deep learning methods for synthetic aperture radar image despeckling: An overview of trends and perspectives. *IEEE Geoscience and Remote Sensing Magazine*, **9 (2)**, 29–51.

Freund, Y. and R. E. Schapire, 1997: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, **55 (1)**, 119–139.

Friedman, J. H., 2001: Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Gagne II, D. J., S. E. Haupt, D. W. Nychka, and G. Thompson, 2019: Interpretable deep learning for spatial analysis of severe hailstorms. *Monthly Weather Review*, **147 (8)**, 2827–2845.

Ge, J., et al., 2023: A novel liquid water content retrieval method based on mass absorption for single-wavelength cloud radar. *IEEE Transactions on Geoscience and Remote Sensing*.

Gini, F. and M. Rangaswamy, 2008: Knowledge-based radar detection, tracking, and classification.

Goodfellow, I., Y. Bengio, and A. Courville, 2016: *Deep learning*. MIT press.

Greenwald, T. J., G. L. Stephens, T. H. Vonder Haar, and D. L. Jackson, 1993: A physical retrieval of cloud liquid water over the global oceans using special sensor microwave/imager (ssm/i) observations. *Journal of Geophysical Research: Atmospheres*, **98 (D10)**, 18 471–18 488.

Grody, N., 1976: Remote sensing of atmospheric water content from satellites using microwave radiometry. *IEEE Transactions on Antennas and Propagation*, **24 (2)**, 155–162.

Grody, N., J. Zhao, R. Ferraro, F. Weng, and R. Boers, 2001: Determination of precipitable water and cloud liquid water over oceans from the noaa 15 advanced microwave sounding unit. *Journal of Geophysical Research: Atmospheres*, **106 (D3)**, 2943–2953.

Guinn, T. A. and R. J. Barry, 2016: Quantifying the effects of humidity on density altitude calculations for professional aviation education. *International Journal of Aviation, Aeronautics, and Aerospace*, **3 (3)**, 2.

Gultepe, I., et al., 2019: A review of high impact weather for aviation meteorology. *Pure and applied geophysics*, **176**, 1869–1921.

Guo, X., Z. Wang, R. Zhao, Y. Wu, X. Wu, and X. Yi, 2023: Liquid water content measurement with sea multi-element sensor in cardc icing wind tunnel: Calibration and performance. *Applied Thermal Engineering*, **235**, 121 255.

Harel, D., H. Fadida, A. Slepoy, S. Gantz, and K. Shilo, 2014: The effect of mean daily temperature and relative humidity on pollen, fruit set and yield of tomato grown in commercial protected cultivation. *Agronomy*, **4 (1)**, 167–177.

Hartmann, D. L., 2015: *Global physical climatology*, Vol. 103. Newnes.

Hassan, D., G. A. Isaac, P. A. Taylor, and D. Michelson, 2022: Optimizing radar-based rainfall estimation using machine learning models. *Remote Sensing*, **14 (20)**, 5188.

Hassanien, A. E., M. Tolba, and A. T. Azar, 2014: Advanced machine learning technologies and applications. *Second International Conference, Egypt, AML, Springer*, Springer.

Hastie, T., R. Tibshirani, and J. H. Friedman, 2009: *The elements of statistical learning: data mining, inference, and prediction*, Vol. 2. Springer.

Holmstrom, M., D. Liu, and C. Vo, 2016: Machine learning applied to weather forecasting. *Meteorol. Appl*, **10**, 1–5.

Hong Kong Observatory, 2021: Highlight of hong kong climate. `https://www.hko.gov.hk/en/cis/climat.htm`.

Hosmer Jr, D. W., S. Lemeshow, and R. X. Sturdivant, 2013: *Applied logistic regression*. John Wiley Sons.

Houidi, S., D. Fourer, and F. Auger, 2020: On the use of concentrated time–frequency representations as input to a deep convolutional neural network: Application to non intrusive load monitoring. *Entropy*, **22 (9)**, 911.

Illingworth, A., et al., 2007: Cloudnet: Continuous evaluation of cloud profiles in seven operational models using ground-based observations, b. am. meteorol. soc., 88, 883–898. BAMS-88-6-883.

Imura, S., J.-i. Furumoto, T. Tsuda, and T. Nakamura, 2007: Estimation of humidity profiles by combining co-located vhf and uhf wind-profiling radar observation. *Journal of the Meteorological Society of Japan. Ser. II*, **85 (3)**, 301–319.

ITU, 2003: Rec. itu-r rs.1624: Sharing between the earth exploration-satellite (passive) and airborne altimeters in the aeronautical radionavigation service in the band 4 200-4 400 mhz. Tech. rep., ITU.

Jacob, D., 2001: The role of water vapour in the atmosphere. a short overview from a climate modeller's point of view. *Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy*, **26 (6-8)**, 523–527.

James, G., D. Witten, T. Hastie, and R. Tibshirani, 2013: *An Introduction to Statistical Learning: with Applications in R*. Springer.

Javeed, S., K. S. Alimgeer, W. Javed, M. Atif, and M. Uddin, 2018: A modified artificial neural network based prediction technique for tropospheric radio refractivity. *Plos one*, **13 (3)**, e0192 069.

Jiang, W., Y. Ren, Y. Liu, and J. Leng, 2021: A method of radar target detection based on convolutional neural network. *Neural Computing and Applications*, **33**, 9835–9847.

Jiang, W., Y. Wang, Y. Li, Y. Lin, and W. Shen, 2023: Radar target characterization and deep learning in radar automatic target recognition: A review. *Remote Sensing*, **15 (15)**, 3742.

Johnston, J. D., 1988: Transform coding of audio signals using perceptual noise criteria. *IEEE Journal on selected areas in communications*, **6 (2)**, 314–323.

Jordan, M. I. and T. M. Mitchell, 2015: Machine learning: Trends, perspectives, and prospects. *Science*, **349 (6245)**, 255–260.

Kaelbling, L. P., M. L. Littman, and A. W. Moore, 1996: Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, **4**, 237–285.

Kim, M., J. Cermak, H. Andersen, J. Fuchs, and R. Stirnberg, 2020: A new satellite-based retrieval of low-cloud liquid-water path using machine learning and meteosat seviri data. *Remote Sensing*, **12 (21)**, 3475.

Klaus, V., L. Bianco, C. Gaffard, M. Matabuena, and T. J. Hewison, 2006: Combining uhf radar wind profiler and microwave radiometer for the estimation of atmospheric humidity profiles. *Meteorologische Zeitschrift*, **15 (1)**, 87–98.

Korolev, A., G. Isaac, J. Strapp, S. Cober, and H. Barker, 2007: In situ measurements of liquid water content profiles in midlatitude stratiform clouds. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, **133 (628)**, 1693–1699.

Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012: Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, **25**.

Kumar, V. and C. Khosla, 2018: Data cleaning-a thorough analysis and survey on unstructured data. *2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, 305–309.

Kuo, C. C., D. H. Staelin, and P. W. Rosenkranz, 1994: Statistical iterative scheme for estimating atmospheric relative humidity profiles. *IEEE Transactions on Geoscience and Remote Sensing*, **32 (2)**, 254–260.

Kutner, M. H., C. J. Nachtsheim, J. Neter, and W. Li, 2005: *Applied linear statistical models*. McGraw-hill.

Kwak, S. K. and J. H. Kim, 2017: Statistical data preparation: management of missing values and outliers. *Korean journal of anesthesiology*, **70 (4)**, 407–411.

Kyriakos, C. S., 2023: Tracking filter for radio altimeter. `https://patents.google.com/patent/US4427981A`.

Lang, P., X. Fu, M. Martorella, J. Dong, R. Qin, X. Meng, and M. Xie, 2020: A comprehensive survey of machine learning applied to radar signal processing. *arXiv preprint arXiv:2009.13702*.

LeCun, Y., Y. Bengio, and G. Hinton, 2015: Deep learning. *Nature*, **521 (7553)**, 436–444.

Levanon, N. and E. Mozeson, 2004: *Radar signals*. John Wiley & Sons.

Li, W., H. Chen, and L. Han, 2023: Polarimetric radar quantitative precipitation estimation using deep convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*.

Lillicrap, T. P., J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, 2015: Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Liu, S., Y. Yin, Z. Chu, and S. An, 2020: Cdl: A cloud detection algorithm over land for mwhs-2 based on the gradient boosting decision tree. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **13**, 4542–4549.

Loh, W.-Y., 2011: Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, **1 (1)**, 14–23.

Malardel, S., N. Wedi, W. Deconinck, M. Diamantakis, C. Kühnlein, G. Mozdzynski, M. Hamrud, and P. Smolarkiewicz, 2016: A new grid for the ifs. *ECMWF newsletter*, **146 (23-28)**, 321.

McGovern, A., R. J. Chase, M. Flora, D. J. Gagne, R. Lagerquist, C. K. Potvin, N. Snook, and E. Loken, 2023: A review of machine learning for convective weather. *Artificial Intelligence for the Earth Systems*, 1–61.

McGovern, A., K. L. Elmore, D. J. Gagne, S. E. Haupt, C. D. Karstens, R. Lagerquist, T. Smith, and J. K. Williams, 2017: Using artificial intelligence to improve real-time decision-making for high-impact weather. *Bulletin of the American Meteorological Society*, **98 (10)**, 2073–2090.

Meischner, P., 2005: *Weather radar: principles and advanced applications*. Springer Science & Business Media.

Mirjalili, S., 2015: The ant lion optimizer. *Advances in engineering software*, **83**, 80–98.

Misra, H., S. Ikbal, H. Bourlard, and H. Hermansky, 2004: Spectral entropy based feature for robust asr. *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, IEEE, Vol. 1, I–193.

Mitchell, T., 1997: *Machine Learning*. McGraw Hill.

Mnih, V., et al., 2015: Human-level control through deep reinforcement learning. *Nature*, **518 (7540)**, 529–533.

Montgomery, D. C., E. A. Peck, and G. G. Vining, 2021: *Introduction to Linear Regression Analysis*. John Wiley Sons.

Morrison, H., et al., 2020: Confronting the challenge of modeling cloud and precipitation microphysics. *Journal of advances in modeling earth systems*, **12 (8)**, e2019MS001 689.

Murphy, K. P., 2012: *Machine learning: a probabilistic perspective*. MIT press.

Nandan, R., M. V. Ratnam, V. R. Kiran, and D. N. Naik, 2022: Retrieval of cloud liquid water path using radiosonde measurements: Comparison with modis and era5. *Journal of Atmospheric and Solar-Terrestrial Physics*, **227**, 105 799.

Ng, C. S. W., H. Djema, M. N. Amar, and A. J. Ghahfarokhi, 2022: Modeling interfacial tension of the hydrogen-brine system using robust machine learning techniques: Implication for underground hydrogen storage. *International Journal of Hydrogen Energy*, **47 (93)**, 39 595–39 605.

Nilsson, N. J., 1998: *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann.

Nimnuan, P., et al., 2017: Determination of effective droplet radius and optical depth of liquid water clouds over a tropical site in northern thailand using passive microwave soundings, aircraft measurements and spectral irradiance data. *Journal of Atmospheric and Solar-Terrestrial Physics*, **161**, 8–18.

Notaro, V., L. Liuzzo, G. Freni, and G. La Loggia, 2015: Uncertainty analysis in the evaluation of extreme rainfall trends and its implications on urban drainage system design. *Water*, **7 (12)**, 6931–6945.

Ogunjo, S., O. Ife-Adediran, E. Owoola, and I. Fuwape, 2019: Quantification of historical drought conditions over different climatic zones of nigeria. *Acta Geophysica*, **67**, 879–889.

Olafsson, H. and J.-W. Bao, 2020: *Uncertainties in Numerical Weather Prediction*. Elsevier.

O'Connor, J. C., M. J. Santos, S. C. Dekker, K. T. Rebel, and O. A. Tuinenburg, 2021: Atmospheric moisture contribution to the growing season in the amazon arc of deforestation. *Environmental Research Letters*, **16 (8)**, 084 026.

Park, H., J. Lee, C. Yoo, S. Sim, and J. Im, 2021: Estimation of spatially continuous near-surface relative humidity over japan and south korea. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **14**, 8614–8626.

Peeters, G., 2004: A large set of audio features for sound description (similarity and classification) in the cuidado project. *CUIDADO Ist Project Report*, **54 (0)**, 1–25.

Quinlan, J. R., 1986: Induction of decision trees. *Machine Learning*, **1 (1)**, 81–106.

RTCA, 2020: Assessment of c-band mobile telecommunications interference impact on low range radar altimeter operations. Tech. rep., RTCA, Washington, DC, USA.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams, 1986: Learning representations by back-propagating errors. *Nature*, **323 (6088)**, 533–536.

Russell, S. J. and P. Norvig, 2016: *Artificial intelligence: a modern approach*. Pearson.

Saïd, F., B. Campistron, and P. Di Girolamo, 2018: High-resolution humidity profiles retrieved from wind profiler radar measurements. *Atmospheric Measurement Techniques*, **11 (3)**, 1669–1688.

Samuel, A. L., 1959: Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, **3 (3)**, 210–229.

Scheirer, E. and M. Slaney, 1997: Construction and evaluation of a robust multifeature speech/music discriminator. *1997 IEEE international conference on acoustics, speech, and signal processing*, IEEE, Vol. 2, 1331–1334.

Schubert, E., J. Wolfe, A. Tarnopolsky, et al., 2004: Spectral centroid and timbre in complex, multiple instrumental textures. *Proceedings of the international conference on music perception and cognition, North Western University, Illinois*, sn, 112–116.

Senthil Kumar, P., 2019: Improved prediction of wind speed using machine learning. *EAI Endorsed Transactions on the Energy Web*, **6 (23)**.

Shafi, M., et al., 2017: 5g: A tutorial overview of standards, trials, challenges, deployment, and practice. *IEEE journal on selected areas in communications*, **35 (6)**, 1201–1221.

Sherwood, S., 2010: Direct versus indirect effects of tropospheric humidity changes on the hydrologic cycle. *Environmental Research Letters*, **5 (2)**, 025 206.

Shi, X., Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, 2015: Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, **28**.

Shukla, J. and Y. Sud, 1981: Effect of cloud-radiation feedback on the climate of a general circulation model. *Journal of the Atmospheric Sciences*, **38 (11)**, 2337–2353.

Silver, D., et al., 2016: Mastering the game of go with deep neural networks and tree search. *Nature*, **529 (7587)**, 484–489.

Silver, D., et al., 2017: Mastering the game of go without human knowledge. *Nature*, **550 (7676)**, 354–359.

Simonoff, J. S., 2012: *Smoothing methods in statistics*. Springer Science & Business Media.

Skolnik, M. I., 1980: Introduction to radar systems. *New York*.

SS, V. C. and A. HS, 2022: Nature inspired meta heuristic algorithms for optimization problems. *Computing*, **104 (2)**, 251–269.

Stankov, B. B., E. E. Gossard, B. L. Weber, R. J. Lataitis, A. B. White, D. E. Wolfe, D. C. Welsh, and R. G. Strauch, 2003: Humidity gradient profiles from wind profiling radars using the noaa/etl advanced signal processing system (sps). *Journal of Atmospheric and Oceanic Technology*, **20 (1)**, 3–22.

Stoica, P., R. L. Moses, et al., 2005: *Spectral analysis of signals*, Vol. 452. Pearson Prentice Hall Upper Saddle River, NJ.

Sutton, R. S. and A. G. Barto, 2018: *Reinforcement Learning: An Introduction*. MIT Press.

Talebkeikhah, M., M. N. Amar, A. Naseri, M. Humand, A. Hemmati-Sarapardeh, B. Dabir, and M. E. A. B. Seghier, 2020: Experimental measurement and compositional modeling of crude oil viscosity at reservoir conditions. *Journal of the Taiwan Institute of Chemical Engineers*, **109**, 35–50.

Techel, F. and C. Pielmeier, 2011: Point observations of liquid water content in wet snow–investigating methodical, spatial and temporal aspects. *The Cryosphere*, **5 (2)**, 405–418.

Teh, H. Y., A. W. Kempa-Liehr, and K. I.-K. Wang, 2020: Sensor data quality: A systematic review. *Journal of Big Data*, **7 (1)**, 1–49.

Thies, B., S. Egli, and J. Bendix, 2017: The influence of drop size distributions on the relationship between liquid water content and radar reflectivity in radiation fogs. *Atmosphere*, **8 (8)**, 142.

Trenberth, K. E. and C. J. Guillemot, 1998: Evaluation of the atmospheric moisture and hydrological cycle in the ncep/ncar reanalyses. *Climate Dynamics*, **14**, 213–231.

Tsuda, T., M. Miyamoto, and J.-i. Furumoto, 2001: Estimation of a humidity profile using turbulence echo characteristics. *Journal of Atmospheric and Oceanic Technology*, **18 (7)**, 1214–1222.

Vicen-Bueno, R., R. Carrasco-Alvarez, M. Rosa-Zurera, and J. C. Nieto-Borge, 2009: Sea clutter reduction and target enhancement by neural networks in a marine radar system. *Sensors*, **9 (3)**, 1913–1936.

Wallace, J. M. and P. V. Hobbs, 2006: *Atmospheric science: an introductory survey*, Vol. 92. Elsevier.

Wang, H., W. Wang, J. Wang, D. Gong, D. Zhang, L. Zhang, and Q. Zhang, 2021: Rainfall microphysical properties of landfalling typhoon yagi (201814) based on the observations of micro rain radar and cloud radar in shandong, china. *Advances in Atmospheric Sciences*, **38 (6)**, 994–1011.

Weng, F. and N. C. Grody, 1994: Retrieval of cloud liquid water using the special sensor microwave imager (ssm/i). *Journal of Geophysical Research: Atmospheres*, **99 (D12)**, 25 535–25 551.

Weng, F., N. C. Grody, R. Ferraro, A. Basist, and D. Forsyth, 1997: Cloud liquid water climatology from the special sensor microwave/imager. *Journal of climate*, **10 (5)**, 1086–1098.

Weng, F., L. Zhao, R. R. Ferraro, G. Poe, X. Li, and N. C. Grody, 2003: Advanced microwave sounding unit cloud and precipitation algorithms. *Radio Science*, **38 (4)**, 33–1.

Wuest, T., D. Weimer, C. Irgens, and K.-D. Thoben, 2016: Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research*, **4 (1)**, 23–45.

Zhao, Q., Y. Liu, W. Yao, and Y. Yao, 2021: Hourly rainfall forecast model using supervised learning algorithm. *IEEE Transactions on Geoscience and Remote Sensing*, **60**, 1–9.

Zhou, K., Y. Zheng, W. Dong, and T. Wang, 2020: A deep learning network for cloud-to-ground lightning nowcasting with multisource data. *Journal of Atmospheric and Oceanic Technology*, **37 (5)**, 927–942.

Zhu, L., J. Suomalainen, J. Liu, J. Hyyppä, H. Kaartinen, H. Haggren, et al., 2018: A review: Remote sensing sensors. *Multi-purposeful application of geospatial data*, 19–42.

Appendix A - List Of Symbols

y	Predicted output
β_0	Intercept
β_i	Coefficients or weights of the model
x_i	Input features
ε	Error term
$E(p)$	Prediction error
N	Number of samples
u_i	Learning sample input
v_i	Learning sample output
p	Predictor
k	Number of smaller samples for cross-validation
$E^{\mathrm{CV}}(p)$	Cross-validation error
N_k	Size of the k-th sample
X_k	k-th sample
$p^{(k)}(u_i)$	Predictor derived from the k-th sample
$E^{\mathrm{ts}}(p)$	Test sample error
$R(t)$	Node impurity
$N_w(t)$	Weighted number of instances in node t
w_i	Weight of the i-th instance
f_i	Frequency variable
z	Linear combination of input features in logistic regression

$\sigma(z)$	Sigmoid function
$d(p, q)$	Euclidean distance between points p and q
p	Data point in feature space
q	Data point in feature space
C	Number of classes
p_i	Proportion of instances of class i in the node
t	Parent node
t_k	Child nodes
K	Number of child nodes
z_i	Output score for class i
y_i	True label
p_i	Predicted probability
y_{ij}	Binary indicator for correct classification
p_{ij}	Predicted probability for class j for sample i
ρ	Correlation coefficient
X_{true}	True data points
$X_{\text{predicted}}$	Predicted data points
μ_{true}	Mean of the true data
$\mu_{\text{predicted}}$	Mean of the predicted data
σ_{true}	Standard deviation of the true data
$\sigma_{\text{predicted}}$	Standard deviation of the predicted data
R^2	Coefficient of determination
u	Horizontal wind vector component
v	Horizontal wind vector component
w	Vertical wind vector component
q_0	Humidity at level z_0
T	Temperature

P	Pressure
θ	Potential temperature
M	Vertical gradient of atmospheric refractivity
C_n^2	Turbulence structure parameter
α^2	Scalar constant dependent on specific regions
$\frac{dV}{dz}$	Vertical shear of the horizontal wind vector
ϵ	Turbulent kinetic energy
$q(z)$	Humidity profile
α	Scalar constant
β	Regression coefficient

Appendix B - List Of Acronyms and Abbreviations

ACO	Ant Colony Optimization
AFGCS	Automatic Flight Guidance and Control Systems
AI	Artificial Intelligence
AGL	Above Ground Level
ALO	Antlion Optimization
ANN	Artificial Neural Network
Bagged	Bootstrap Aggregating
BET	Bagged Ensemble Tree
BiNN	Bi-layered Neural Network
BoostedET	Boosted Ensemble Tree
BPF	Band-Pass Filter
CINRAD	China's new-generation weather radar
CLAAS-2	Cloud Archive User Service - 2
CLWC	Cloud Liquid Water Content
CNN	Convolutional Neural Network
CRF	Chirp Repetition Frequency
CT	Coarse Tree
DL	Deep Learning
DNN	Deep Neural Networks
ECAM	Electronic Centralized Aircraft Monitoring
ECMWF	European Centre for Medium-Range Weather Forecasts
EICAS	Engine-Indicating and Crew-Alerting System
FDR	False Discovery Rate
FFT	Fast Fourier Transform
FMCW	Frequency-Modulated Continuous-Wave
FN	False Negative
FNR	False Negative Rate
FP	False Positives
FSK	Frequency-Shift Keying
FT	Fine Tree
FAA	Federal Aviation Administration
GA	Genetic Algorithms
GBDT	Gradient Boosting Decision Tree
GNB	Gaussian Naive Bayes
GNSS-RO	Global Navigation Satellite System-Radio Occultation

GOA	Grasshopper Optimization Algorithm
GPR	Gaussian Process Regression
GPS	Global Positioning System
GPU	Graphics Processing Unit
HKO	Hong Kong Observatory
I/Q	In-phase and Quadrature components
ILR	Interactions Linear Regression
k-NN	k-Nearest Neighbors
LMA	Levenberg-Marquardt
LR	Linear Regression
LSTM	Long Short-Term Memory
LWC	Liquid Water Content
LWP	Liquid Water Path
MAE	Mean Absolute Error
ML	Machine Learning
MSE	Mean Squared Error
MT	Medium Tree
MLP	Multilayer Perceptron
MWHS-2	Microwave Humidity Sounder-2
NNN	Narrow Neural Network
OLS	Ordinary Least Squares
OOB	Out-of-Band
PCA	Principal Component Analysis
PCI	Physical Layer Cell Identity
PDR	Partial Dependency Plot
Pr	Pressure
PPV	Positive Predictive Value
PSK	Phase-Shift Keying
PSO	Particle Swarm Optimization
PSS	Primary Synchronization Signal
PWS	Predictive Wind Shear
RA	Radar Altimeter
RCS	Radar Cross Section
ReLU	Rectified Linear Unit
RF	Radio Frequency
RFI	Radio Frequency Interference
RH	Relative Humidity
RL	Reinforcement learning
RMSE	Root Mean Squared Error
SCAMS	scanning microwave spectrometer
SDR	Software-Defined Radio
SEVIRI	Spinning Enhanced Visible and Infrared Imager
SGD	stochastic gradient descent
SNR	Signal-to-Noise Ratio
SH	Specific Humidity

SSS	Secondary Synchronization Signal
STD	standard deviation
SW	Spectrum Width
SVM	Support Vector Machine
TC	Total Cost
TN	True Negatives
TP	True Positives
TPR	True Positive Rate
TriNN	Tri-layered Neural Network
5G-NR	Fifth Generation-New Radio