WEATHER FORECASTING USING MACHINE LEARNING

Mini Project Report

Submitted to the APJ Abdul Kalam Technological University in

partial fulfillment of requirements for the award of degree

Bachelor of Technology

in

Computer Science and Engineering

By

NAKSHATHRA S NAIR (LBT22CS076)

NANDANA N (LBT22CS077)

REKHA B (LBT22CS095)

VARADA A S (LBT22CS127)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LBS INSTITUTE OF TECHNOLOGY FOR WOMEN

TRIVANDRUM, KERALA

APRIL 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LBS INSTITUTE OF TECHNOLOGY FOR WOMEN

TRIVANDRUM, KERALA

CERTIFICATE

This is to certify that the report entitled **'WEATHER FORECASTING USING MACHINE LEARNING'** submitted by **NAKSHATHRA S NAIR(LBT22CS076)** , **NANDANA N(LBT22CS077)** , **REKHA B(LBT22CS095)** , **VARADA A S(LBT22CS127)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering is a bonafide record of the mini project work carried out by them under our guidance and supervision.

Prof. Smitha E.S	**Dr. Sumithra M.D**	**Dr. Sumithra M.D**
Associate Professor	Associate Professor	Head of Department
Department of CSE	Department of CSE	Department of CSE
LBSITW, TVM	LBSITW, TVM	LBSITW, TVM
(Mini Project Guide)	(Mini Project Coordinator)	

DECLARATION

We, the undersigned, hereby declare that the project report titled **'WEATHER FORECASTING USING MACHINE LEARNING'** submitted in partial fulfillment of the requirements for the Bachelor of Technology degree at APJ Abdul Kalam TechnologicalUniversity, Kerala, represents our original work conducted under the supervision of **Prof. Smitha E.S** Associate Professor, Department of Computer Science and Engineering, LBS Institute of Technology for Women, Poojappura. We affirm that this submission reflects our own ideas and that any contributions from external sources have been accurately cited and referenced. We attest to our adherence to the principles of academic honesty and integrity, ensuring that all data, ideas,facts, and sources have been presented truthfully and ethically. We acknowledge that any breach of academic integrity or misrepresentation of data may result in disciplinary action by the institute and/or the University. Furthermore, we confirm that this report has not been previously used to obtain any degree, diploma, or similar title from any other academic institution.

Place: Thiruvananthapuram

Date: April 2025

NAKSHATHRA S NAIR

NANDANA N

REKHA B

VARADA A S

ACKNOWLEDGEMENT

We extend our heartfelt gratitude to **Dr.Smithamol M.B** , our Principal, for the provision of essential facilities and infrastructure vital for the completion of this project. Our sincere appreciation also goes to **Dr.Sumithra M.D,** Head of the Department of Computer Science and Engineering, for facilitating the necessary resources throughout this endeavour. Special thanks are due to **Dr.Sumithra M.D**, our project coordinator and Associate Professor in the Department of Computer Science and Engineering, for her invaluable guidance and assistance throughout the project's duration. We are equally grateful to **Prof. Smitha E.S**, our project guide and Associate Professor in the Department of Computer Science and Engineering, for offering us the opportunity and expert guidance to undertake our third-year project. We would also like to recognize the collective efforts of all staff members in the department for their continuous support and encouragement. Finally, we extend our gratitude to our friends and well-wishers for their cooperation, support, and insightful suggestions, which have contributed significantly to the advancement of this project.

NANDANA N

REKHA B

NAKSHATHRA S NAIR

VARADA A S

ABSTRACT

Accurate weather forecasting plays a vital role in various sectors such as agriculture, transportation, disaster management, and daily life planning. Traditional forecasting methods, while effective to an extent, often struggle to capture complex climatic patterns, especially at local scales. This project aims to develop a weather forecasting system for Thiruvananthapuram, leveraging the XGBoost gradient boosting algorithm, which is renowned for its efficiency and predictive accuracy. The proposed approach involves preprocessing historical weather data, extracting key meteorological features, and training the model to learn intricate dependencies among climatic variables. The model utilizes five years of historical weather data, incorporating key parameters such as temperature (maximum, minimum, and average), pressure, and wind speed to generate highly accurate 5-day weather forecasts. By applying machine learning techniques, particularly regression modelling, the system provides more reliable real-time weather predictions compared to conventional methods. Thiruvananthapuram, being a coastal city with a tropical climate, experiences significant variations in temperature, humidity, and monsoon patterns. Machine learning-based forecasting models can help mitigate the impact of extreme weather events by improving the accuracy of short-term and long-term predictions. By integrating advanced ML techniques with historical weather data, this project contributes to more precise weather predictions, ultimately benefiting local residents, businesses, disaster response teams, and government agencies in making well-informed decisions based on reliable weather insights.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	DESCRIPTION
ML	Machine Learning
XGBOOST	Extreme Gradient Boosting
NWP	Numerical Weather Prediction
Tmax	Maximum temperature
Tmin	Minimum temperature
Tavg	Average temperature
Wspd	Windspeed
Pres	Pressure

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO WEATHER FORECASTING SYSTEM

Weather plays a fundamental role in shaping human activities, influencing everything from agricultural productivity to transportation systems, and even urban planning. Accurate weather forecasts can significantly enhance decision-making processes across various sectors, helping individuals and organizations to prepare for adverse weather conditions and optimize their strategies. For instance, farmers can adjust their planting schedules, businesses can plan for energy consumption based on temperature forecasts, and transportation networks can adjust operations in anticipation of storms or fog. Despite the importance of weather predictions, forecasting, especially in tropical and coastal regions like Thiruvananthapuram, remains a complex and challenging task. The rapidly changing atmospheric conditions and the highly dynamic interactions between various climatic variables often complicate predictions, making them less reliable.

Traditional weather forecasting systems primarily rely on physical models and numerical simulations based on complex mathematical equations that simulate atmospheric behavior. These models, while effective to some degree, often require immense computational resources and can struggle to deliver precise local forecasts. Numerical Weather Prediction (NWP) models generate predictions based on large-scale data such as temperature, pressure, humidity, and wind patterns observed from satellites and weather stations. However, these models are often limited when it comes to accurately predicting weather at smaller, localized scales, such as specific regions within a city or areas affected by unique local factors like coastal proximity or mountainous terrain.

As the world generates more and more weather-related data, the need for more efficient and accurate forecasting techniques has become apparent. The availability of vast historical weather datasets, combined with advancements in computational intelligence, has made machine learning (ML) a promising alternative to traditional forecasting methods. Unlike traditional systems that rely heavily on predefined physical equations, machine learning algorithms can learn from vast amounts of historical data, adapting over time and improving their predictions.

These algorithms are particularly useful in handling complex, nonlinear relationships between weather variables, something that traditional models struggle to do. Machine learning can efficiently process large datasets, capture intricate patterns, and generate reliable forecasts without the need for explicit programming of every possible weather scenario.

Machine learning models continuously refine their predictions by learning from new data. This means that over time, they can become more accurate as they adapt to changing weather patterns and trends that may not be immediately apparent to traditional forecasting models. This adaptability is particularly valuable for regions like Thiruvananthapuram, which are characterized by unique and rapidly changing climatic conditions, including the influence of the monsoon season, coastal winds, and other regional factors. Machine learning models can adjust more dynamically to these changes, making them highly effective for short-term, localized weather predictions.

Among the various machine learning techniques, **XGBoost (Extreme Gradient Boosting)** stands out as an excellent choice for weather forecasting. XGBoost is a powerful ensemble learning algorithm based on decision trees, known for its accuracy, scalability, and ability to handle large, complex datasets. XGBoost works by building an ensemble of decision trees where each new tree corrects the errors of the previous one, an approach known as gradient boosting. This method helps the model learn from its mistakes and refine its predictions iteratively. The algorithm's ability to handle non-linear relationships, combined with regularization techniques that reduce overfitting, makes it particularly well-suited for the variable and often noisy data encountered in weather forecasting. Moreover, XGBoost is highly efficient, capable of processing large datasets quickly, which is crucial when dealing with real-time weather predictions.

This project aims to apply machine learning, specifically XGBoost, to develop a weather forecasting system for Thiruvananthapuram. By leveraging historical weather data collected over several years, the project seeks to build a model that can predict essential weather parameters such as temperature, atmospheric pressure, and wind speed for the upcoming days. Thiruvananthapuram's unique climatic features, influenced by its coastal location and the seasonal monsoons, present distinct challenges that make precise forecasting critical. The goal is to create a robust, adaptive forecasting model that can provide reliable short-term predictions, which are vital for agricultural planning, transportation, and local governance.

The project will involve gathering and preprocessing historical weather data for Thiruvananthapuram, cleaning and structuring the data to ensure that it is suitable for machine learning applications. The XGBoost algorithm will then be applied to this dataset to train a model capable of identifying patterns and making predictions based on the historical trends found in the data. In addition to developing the predictive model, the project will also involve creating a user-friendly interface that allows users to input a date and receive accurate forecasts for key weather parameters. This interface will make the model accessible to a wide range of users, from everyday individuals to local authorities who need to make weather-sensitive decisions.

By focusing specifically on Thiruvananthapuram, this project aims to address the need for regional weather forecasting, an area that is often overlooked in global climate models. Traditional forecasting methods are generally designed for large-scale predictions, which may not always be relevant for specific localities that experience unique climatic conditions. By providing a localized weather forecasting system, this project will help local stakeholders—such as farmers, urban planners, and transportation authorities—make more informed decisions based on accurate, real-time weather predictions. This localized approach will not only improve the precision of weather forecasts but also highlight the potential for machine learning to contribute to more customized and adaptive forecasting systems.

Through this initiative, the project demonstrates how machine learning can be leveraged to overcome some of the limitations of traditional weather forecasting. By applying a data-driven approach using XGBoost, the project aims to enhance the accuracy and reliability of short-term weather predictions in Thiruvananthapuram, ultimately contributing to the broader field of regional forecasting. The ultimate goal is to provide a tool that can support better decision-making across a variety of sectors, thereby improving the resilience of the community to weather-related challenges and enhancing the overall quality of life for its residents.

1.2 OBJECTIVE

To develop a machine learning model that predicts continuous weather conditions (temperature, pressure and windspeed) for the next 5 days in Trivandrum district using historical weather data,integrating a web interface for user-friendly access.

CHAPTER 2

LITERATURE REVIEW

These six papers explore the integration of machine learning (ML) and traditional weather prediction methods to improve forecasting accuracy. The studies present a range of ML techniques, including hybrid models combining NWP and ML, Random Forests, SVM, ANNs, and deep learning methods like LSTMs and CNNs. Several papers focus on real-time forecasting using satellite and sensor data, while others examine the advantages of ML in specific applications like rainfall prediction, temperature, humidity, and solar forecasting. The key benefits across these studies include enhanced accuracy, real-time updates, and better applicability in remote areas. However, challenges such as high computational costs, data dependency, and limited long-term accuracy are common.

1. **A Hybrid Machine Learning Numerical Weather Prediction Approach for Rainfall Prediction - Amol Ashok Patil, Kedar Kulkarni,2023**

This journal presents an advanced technique to improve rainfall forecasting by integrating Numerical Weather Prediction (NWP) models with Machine Learning (ML). Traditional NWP models rely on mathematical equations and physical principles to simulate atmospheric conditions. However, these models often struggle with local variations and prediction errors, especially in regions with dynamic weather patterns. To overcome these limitations, the authors propose a hybrid approach that enhances NWP predictions using machine learning techniques.

In this study, the researchers first collect historical meteorological data, including rainfall intensity, temperature, humidity, and atmospheric pressure. This dataset is used to train an ML model, which learns patterns and relationships among various climatic parameters. The ML model is then combined with the outputs of an NWP model, refining the predictions by identifying biases and correcting inaccuracies. This fusion of physics-based and data-driven approaches results in a more precise forecasting system.

Experimental results demonstrate that the hybrid model outperforms conventional NWP methods, reducing common prediction errors and improving the accuracy of rainfall forecasting. The study finds that integrating ML allows the model to adapt to non-linear and region-specific weather behaviours, making it more reliable for real-world applications. This approach is particularly useful in areas where monsoonal variations and sudden weather changes affect daily life, agriculture, and disaster preparedness.

By leveraging artificial intelligence and meteorological modelling, the paper highlights the potential of hybrid learning in weather prediction. This research is valuable for scientists, meteorologists, and policymakers who seek better forecasting tools to mitigate the risks associated with extreme rainfall events. The findings of this study reinforce the growing trend of using ML techniques like gradient boosting, deep learning, and neural networks in meteorological research.[1]

2. **Weather forecasting using Machine Learning - Amar Deep Gupta, Devansh Katoch, Smita Sharma, Shipra Ravi Kumar, Aditya Rana,2023**

This journal explores the application of machine learning techniques to enhance the accuracy of weather predictions. The study highlights the limitations of traditional forecasting methods, which rely on numerical weather prediction (NWP) models and statistical approaches that often fail to capture the complexities of local climatic conditions. To address these challenges, the researchers propose a machine learning-based forecasting model that utilizes historical weather data to predict key parameters such as temperature, humidity, wind speed, and precipitation.

The authors employ various machine learning algorithms, including decision trees, random forests, and deep learning models, to analyse large datasets and identify patterns that influence weather conditions. The study evaluates different models based on their accuracy, efficiency, and ability to handle complex meteorological variables. The findings indicate that machine learning models, particularly ensemble methods like gradient boosting, outperform conventional approaches in terms of prediction accuracy.

This research emphasizes the advantages of using data-driven techniques for weather forecasting, as they can adapt to non-linear climate patterns and improve short-term predictions. The study also discusses potential applications in disaster management, agriculture, and urban planning, where accurate weather forecasts can help mitigate risks and optimize resource allocation. The results suggest that machine learning offers a scalable and efficient solution for improving weather prediction systems, making them more accessible and reliable for real-world use.[2]

3. **Real-Time Weather Forecasting Using Machine Learning - Snehjot Kaur, Ajay Pal Singh, Abhishek Pandey, Harshit Chauhan,2023**

This Journal presents an advanced technique to improve rainfall forecasting by integrating Numerical Weather Prediction (NWP) models with Machine Learning (ML). Traditional NWP models rely on mathematical equations and physical principles to simulate atmospheric conditions. However, these models often struggle with local variations and prediction errors, especially in regions with dynamic weather patterns. To overcome these limitations, the authors propose a hybrid approach that enhances NWP predictions using machine learning techniques.

In this study, the researchers first collect historical meteorological data, including rainfall intensity, temperature, humidity, and atmospheric pressure. This dataset is used to train an ML model, which learns patterns and relationships among various climatic parameters. The ML model is then combined with the outputs of an NWP model, refining the predictions by identifying biases and correcting inaccuracies. This fusion of physics-based and data-driven approaches results in a more precise forecasting system.

Experimental results demonstrate that the hybrid model outperforms conventional NWP methods, reducing common prediction errors and improving the accuracy of rainfall forecasting. The study finds that integrating ML allows the model to adapt to non-linear and region-specific weather behaviours, making it more reliable for real-world applications. This approach is particularly useful in areas where monsoonal variations and sudden weather changes affect daily life, agriculture, and disaster preparedness.

By leveraging artificial intelligence and meteorological modelling, the paper highlights the potential of hybrid learning in weather prediction. This research is valuable for scientists, meteorologists, and policymakers who seek better forecasting tools to mitigate the risks associated with extreme rainfall events. The findings of this study reinforce the growing trend of using ML techniques like gradient boosting, deep learning, and neural networks in meteorological research.[3]

4. **A Comprehensive Study on Weather Forecasting using Machine Learning - Deepti Mishra, Prathibha Joshi,2021**

This journal provides an in-depth analysis of how machine learning techniques can enhance weather prediction accuracy. Traditional weather forecasting methods, which rely on numerical and statistical models, often struggle with high computational costs and inaccuracies in short-term predictions. This study explores how machine learning (ML) algorithms can overcome these challenges by analysing historical weather data and identifying complex patterns in meteorological parameters such as temperature, humidity, wind speed, and rainfall.

The authors review various supervised and unsupervised ML algorithms, including Linear Regression, Decision Trees, Support Vector Machines (SVM), Neural Networks, and Deep Learning models, to determine their effectiveness in weather forecasting. The study compares the strengths and weaknesses of these models, highlighting how deep learning techniques such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks can improve forecasting accuracy by capturing sequential dependencies in climate data.

The research also emphasizes the importance of data preprocessing, feature selection, and real-time data integration for improving model performance. The study concludes that machine learning offers a more flexible and adaptive approach to weather forecasting compared to traditional methods, making it particularly useful for regions experiencing unpredictable climate variations. The authors suggest that combining ML with existing meteorological models can further enhance the precision of forecasts, benefiting applications in agriculture, disaster management, and environmental monitoring.[4]

5. **Machine Learning for Applied Weather Prediction- Sue Ellen Haupt, Jim Cowie, Seth Linden, Tyler McCandles, Branko Kosovic, Stefano Alessandrini,2018**

This journal explores the integration of machine learning (ML) techniques with traditional weather forecasting methods to enhance prediction accuracy. Traditional numerical weather prediction (NWP) models rely on complex physics-based equations to simulate atmospheric conditions, but they often struggle with local variability, computational efficiency, and short-term forecasting errors. This study examines how ML can be applied to refine these models by learning from historical weather patterns and real-time data.

The authors discuss various ML algorithms, including Decision Trees, Random Forests, Neural Networks, and Support Vector Machines (SVM), emphasizing their ability to identify patterns and improve forecast precision. One key focus of the study is how ML can be used to correct biases in NWP models, improving their short-term accuracy, especially for parameters like temperature, precipitation, and wind speed. The research also highlights the potential of deep learning approaches, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), for modelling complex weather systems.

Another important aspect of this study is its emphasis on applied weather prediction, meaning the use of ML in real-world applications such as aviation, energy management, agriculture, and disaster preparedness. The authors conclude that while ML cannot completely replace traditional NWP models, it can significantly enhance their predictive capabilities, making forecasts more reliable, adaptive, and computationally efficient. The study advocates for a hybrid approach, where ML is used alongside meteorological models to provide more accurate and actionable weather forecasts.[5]

6. **Weather Prediction using Machine Learning Algorithms- Aishwarya Shaji, Amritha A R, Rajalakshmi V R,2022**

This journal explores the application of machine learning techniques in improving the accuracy of weather forecasting. Traditional weather prediction models rely on numerical methods and meteorological equations, which often struggle with real-time adaptability and precision. This study focuses on data-driven approaches that leverage machine learning algorithms to analyse past weather patterns and predict key climatic parameters such as temperature, humidity, wind speed, and rainfall.

The authors implement and compare various machine learning models, including Decision Trees, Random Forest, Support Vector Machines (SVM), and Neural Networks, to determine their effectiveness in weather forecasting. By training these models on historical datasets, the study evaluates their accuracy and ability to handle complex, non-linear relationships between different weather variables. The findings reveal that ensemble learning techniques, particularly Random Forest and Neural Networks, outperform conventional models in terms of predictive accuracy.

This research highlights the advantages of machine learning in weather forecasting, including faster processing, improved accuracy, and the ability to adapt to changing climate conditions. The study also discusses practical applications in agriculture, disaster preparedness, and urban planning, where precise weather predictions can aid in better decision-making. The authors conclude that integrating machine learning with meteorological systems can significantly enhance forecasting models, making them more reliable for real-world applications.[6]

Table 2.1 Summary of Literature Review

S.N	TITLE OF PAPER	SUMMARY	PROS	CONS
1.	A Hybrid Machine Learning Numerical Weather Prediction Approach for Rainfall Prediction - Amol Ashok Patil,Kedar Kulkarni,2023	1.Hybrid Model – Merges NWP and ML for better rainfall forecasts. 2.Higher Accuracy – Uses SVM, Neural Networks, and XGBoost for up to 25% improvement	1.Accuracy 2.Adaptability 3.Efficiency	1.Complexity 2.Data-dependence 3.Computation
2.	Weather forecasting using Machine Learning - Amar Deep Gupta, Devansh Katoch, Smita Sharma, Shipra Ravi Kumar, Aditya Rana,2023	1.Reviews Random Forests, SVM, ANN, Decision Trees, LSTMs, and CNNs. 2.Gradient Boosting Machines (98%) and Random Forests (97%) outperform traditional methods	1.High Accuracy 2.Real-World applications	1.Limited dataset details 2.High computational cost
3.	Real-Time Weather Forecasting Using Machine Learning - Snehjot Kaur, Ajay Pal Singh, Abhishek	1.Uses data from satellites, weather stations, and sensors that improves weather prediction	1.Higher accuracy 2.Real-Time updates 3.Improved accessibility	1.Data Quality issues 2.High Computational cost

	Pandey, Harshit Chauhan,2023	2. Uses Numerical weather prediction(NWP), ML algorithms and deep learning for improved accuracy		3.Limited accuracy for long term forecasts
4.	Weather Prediction using Machine Learning Algorithms- Aishwarya Shaji, Amritha A R, Rajalakshmi V R,2022	1. ML Techniques:- Uses Random Forest, Decision Tree, MLP Classifier, and Linear Regression 2. Method:- combines Random Forest and MLP Classifiers that yield higher accuracy	1.Better forecasting in remote areas 2.Adaptability	1.Data dependency 2.Limited long term accuracy
5	A Comprehensive Study on Weather Forecasting using Machine Learning - Deepti Mishra, Prathibha Joshi,2021	1. ML Techniques – Uses linear regression and ANNs for weather forecasting.	1.Accuracy 2.Automation 3.Scalability	1.Complexity 2.Data-dependence 3.Computation

		2. Prediction Focus – Predicts temperature, humidity, and pressure with improved accuracy. 3. Method – Combines decision trees and probability for better predictions		
6	Machine Learning for Applied Weather Prediction - Sue Ellen Haupt, Jim Cowie, Seth Linden, Tyler McCandles, Branko Kosovic, Stefano Alessandrini,2018	1. Focuses on DICast® (NCAR). 2. DICast blends NWP models + historical data, improving accuracy by 10-15%. 3. Uses AI techniques like k-means clustering, neural networks, and AnEn for solar forecasting	1. Improved accuracy 2. Smart AI use	1. Limited focus 2. No real time example

2.1 GAP ANALYSIS

Despite the availability of traditional weather forecasting methods such as Numerical Weather Prediction (NWP), these models often fall short in delivering highly accurate, localized forecasts, especially for specific regions like Trivandrum. These systems rely on solving complex physical equations, requiring enormous computational power, and yet struggle with predicting micro-level weather patterns that vary from one district to another. Furthermore, they are usually built to operate on a national or global scale, thereby lacking the precision needed for localized forecasts. This limitation becomes even more prominent when sudden weather changes occur, such as heavy rainfall or temperature fluctuations, which are frequent in coastal and tropical zones like Trivandrum.

Another significant gap in existing forecasting models is the limited use of modern data-driven techniques like machine learning. Most traditional systems focus on singular or a small set of meteorological features, often ignoring the interaction between variables like temperature, wind speed, and atmospheric pressure. Even when machine learning is used, many models rely on a single algorithm, which may not fully capture the complexity of weather patterns. Additionally, there is a lack of predictive models designed to offer multi-day forecasts from any chosen date, restricting users from planning ahead with confidence. The absence of dynamic, adaptive systems capable of learning from historical data and refining their predictions over time further limits the accuracy and usefulness of existing approaches.

Our project addresses these shortcomings by implementing a machine learning-based forecasting model using XGBoost, a powerful ensemble algorithm known for its high accuracy and efficiency. By training the model on five years of historical weather data specific to Trivandrum, we ensure the forecasts are localized and relevant. The model integrates multiple features such as maximum, minimum, and average temperature, wind speed, and pressure to produce reliable five-day-ahead weather predictions. This approach not only fills the technological and analytical gaps in current forecasting systems but also offers an accessible, low-resource solution that can be deployed efficiently, making it practical for real-world applications like agriculture, tourism, and daily planning.

CHAPTER 3

REQUIREMENT SPECIFICATION AND DESIGN

3.1 SOFTWARE REQUIREMENTS

This project is developed using Python and executed entirely within the Visual Studio Code environment. It involves machine learning libraries for model building and utilizes a simple local server setup to host the project interface through a web browser.

i . Python Environment

Python is the primary programming language utilized for implementing the XGBoost machine learning algorithm within the VS Code environment.

ii . VS Code

VS Code serves as the integrated development environment (IDE) for conducting exploratory data analysis, implementing the XGBoost machine learning model, and documenting the project workflow.

iii . Python Libraries

1. Pandas: Pandas, a powerful data manipulation library, is indispensable for data preprocessing, cleaning, and analysis. Pandas facilitate efficient handling of datasets and data frames. It is used to load the weather dataset (Excel file) , convert the date column into a usable format and to handle missing values in the dataset.
2. Numpy: NumPy, a fundamental library for numerical computations in Python, provides support for array operations and mathematical functions essential for machine learning tasks. It is used to perform numerical operations on weather data and to convert data into arrays that can be fed into the machine learning model.
3. XGBoost: XGBoost is a powerful and optimized machine learning library designed for structured data. It is widely used for regression and classification tasks. It is used to train the weather forecasting model using past weather data and to predict temperature, wind speed, and pressure for the next 5 days.

4. Scikit-learn: Scikit-Learn is a machine learning library that provides tools for model training, evaluation, and data preprocessing. It is used to split the dataset into training and testing sets and to evaluate model performance using the test data.
5. Pickle: Pickle is a built-in Python module used to save and load machine learning models. It is used to save the trained XGBoost model so it can be reused without retraining and to load the saved model when making new predictions.
6. DateTime: The datetime module allows working with dates and times in Python. It is used to calculate the next 5 days' dates for predictions and to format and display the predicted weather data properly.
7. Http Server: The http.server module provides a lightweight HTTP server for handling web requests. It is used to create a simple API that serves weather predictions to the frontend.
8. Json: The json module is used to convert data into JSON format, which is commonly used for APIs and web applications. It is used to convert weather forecast data into JSON format so the frontend can display it.
9. Urlib.parse: The urllib.parse module is used to parse URLs and extract parameters from web requests. It is used to extract the user-provided date from the URL when they request weather forecasts.

iv. Google Chrome Browser

Google Chrome is used to test and view the weather forecasting web application. Ensure that Google Chrome is installed on your system for running and verifying the web-based user interface of the project.

v. JavaScript Development Environment

A suitable JavaScript development environment, such as Visual Studio Code, is used for writing and debugging the frontend JavaScript code. JavaScript is used in the project for handling user input, making HTTP requests to the server, and displaying the weather forecast results dynamically on the webpage.

vi. HTML and CSS

HTML and CSS are used to design and structure the frontend interface of the weather forecasting web application.

- HTML defines the structure of the webpage, including input fields, buttons, tables, and result display sections.
- CSS enhances the visual appearance of the interface, making it more user-friendly by styling elements such as buttons, tables, and layout components.

3.2 HARDWARE REQUIREMENTS

Hardware requirements refer to the specifications and capabilities of physical components necessary for a system or software application to function optimally. Meeting or exceeding these requirements ensures that the system or software can perform tasks effectively and without significant performance degradation. The fundamental hardware requirements essential for the project are given below.

- Operating system: Windows 10/11
- RAM: Minimum 8 GB
- System type:64-bit, x64-based processor (Intel i5/i7 or AMD Ryzen equivalent recommended)

3.3 FUNCTIONAL REQUIREMENTS

The project's functional requirements include accurate weather forecasting for the next five days, real-time weather data processing, and an interactive web-based interface to provide users with easy access to predictions.

i. Weather Data Analysis

a. The system should analyse historical weather data and generate accurate forecasts using the XGBoost model.

b. The input features include maximum temperature (tmax), minimum temperature (tmin), average temperature (tavg), wind speed (wspd), and pressure (pres).

c. The model should predict weather conditions for the next five days based on a given date.

ii. Forecast Visualization & User Interaction

a. The web application should allow users to input a date and receive a five-day weather forecast.

b. The forecast should be displayed in a clear and interactive format, including tables, or other visual elements for better understanding.

c. Users should be able to access weather trends and historical data analysis for better insights.

3.4 NON-FUNCTIONAL REQUIREMENTS

The weather forecasting system prioritizes accuracy, efficiency, reliability, and user-friendly usability. It emphasizes well-structured code for easy maintenance and compliance with data privacy and cybersecurity standards.

ⅰ. Performance

The system should efficiently process weather data and provide accurate forecasts with minimal computation time. The web application should be optimized for fast loading times, ensuring a smooth and responsive user experience.

ⅱ. Reliability

The forecasting model should provide consistently accurate predictions, minimizing errors in temperature, wind speed, and pressure forecasts. The system should be robust against missing or incomplete data, ensuring reliable outputs.

ⅲ. Usability

The user interface should be intuitive and easy to navigate, requiring minimal effort for users to input a date and receive a 5-day weather forecast. It should present weather predictions in a clear and visually appealing manner, with options for graphical representation of trends.

ⅳ. Maintainability

The codebase should be well-structured, modular, and well-documented to facilitate easy debugging, updates, and improvements. The system should allow for the integration of new datasets or model upgrades without major structural changes.

v . Security & Compliance

The system should ensure data privacy and secure handling of user inputs. If storing weather data, it must comply with data protection guidelines. Web security best practices should be followed to prevent unauthorized access or data breaches.

vi . Scalability

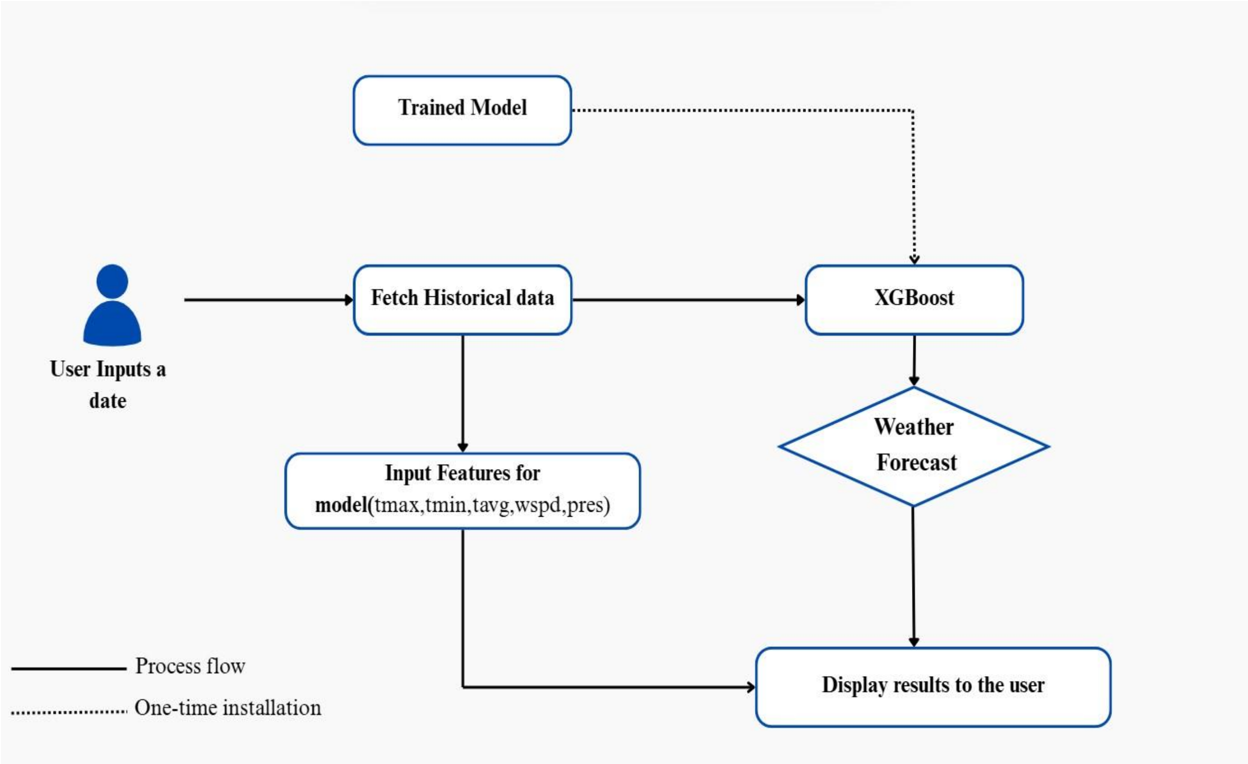
The system should be scalable to accommodate increased data volume and additional forecasting parameters if required. It should support future enhancements such as incorporating more weather variables or integrating external weather APIs.

3.5 OVERALL ARCHITECTURE

The overall architecture begins with the one-time setup of the trained machine learning model within the application environment. Once the system is ready, the user inputs a specific date for which the weather forecast is needed. The backend system then automatically fetches relevant historical weather data, such as maximum temperature (tmax), minimum temperature (tmin), average temperature (tavg), wind speed (wspd), and pressure (pres), to construct input features for the prediction model.

These features are then passed to the pre-trained XGBoost model, which has been optimized and trained using five years of historical weather data from Trivandrum. The model processes the inputs and generates predictions for the next five consecutive days from the user-specified date. The forecast includes daily values for temperature, wind speed, and pressure, offering a comprehensive view of the expected weather conditions.

The predicted results are then displayed to the user through the front-end interface, either in graphical or tabular form. This allows users to make informed decisions for agriculture, travel, or daily planning. The system ensures real-time prediction without the need for any manual data analysis, delivering a fast, efficient, and user-friendly weather forecasting experience.



3.1 Overall Architecture of proposed model

3.6 SUMMARY

The proposed system relies on a Python environment within Jupyter Notebook for implementing machine learning algorithms, including XGBoost for weather forecasting. Additionally, Pandas and NumPy are utilized for data manipulation and numerical computations, while Matplotlib aids in data visualization. The system is designed to predict the next five days of weather for the Trivandrum district based on historical weather data.

The system's functional requirements include data preprocessing, model training, weather prediction, and performance evaluation. Meanwhile, non-functional requirements emphasize accuracy, reliability, usability, maintainability, and scalability.

The overall architecture involves collecting and preprocessing historical weather data, extracting key features such as temperature (tmax, tmin, tavg), wind speed (wspd), and pressure (pres), and training the XGBoost model for forecasting. The trained model is then used to predict future weather conditions. The system aims to provide accurate weather predictions, assisting in better planning and decision-making.

CHAPTER 4

METHODOLOGY

The proposed methodology utilizes the XGBoost machine learning algorithm to forecast weather conditions in Thiruvananthapuram for the next five days. The approach includes multiple phases: data collection, preprocessing, model training, evaluation, and deployment. Initially, historical weather data is sourced from Meteostat, containing meteorological parameters like maximum temperature, minimum temperature, average temperature, atmospheric pressure, and wind speed. The dataset undergoes preprocessing, including handling missing values, formatting date attributes, and feature scaling to ensure optimal model performance. The refined dataset is used to train the XGBoost algorithm, which is optimized using hyperparameter tuning to enhance predictive accuracy. If deployed as a web-based application, the model's predictions will be accessible through an interactive interface, enabling users to input a date range and retrieve weather forecasts. This methodology ensures an accurate, scalable, and efficient weather prediction system leveraging machine learning techniques.

4.1 DETAILED DESIGN

The weather dataset undergoes preprocessing to ensure high-quality input for model development. The key preprocessing steps include

Handling Missing Values: Missing data is identified and addressed using appropriate imputation techniques to maintain data integrity.

Feature Selection: Only relevant meteorological attributes, such as temperature, pressure, and wind speed, are retained to enhance model efficiency.

Data Formatting: The dataset is structured with correctly formatted date attributes to align with the forecasting model's input requirements.

Feature Scaling: Numerical values are scaled appropriately to ensure uniformity in data distribution, improving model performance.

If deployed as a web-based application, an interactive user interface (UI) will allow users to input a date range and receive weather forecasts dynamically. The system will fetch predictions from the trained XGBoost model and present them in a clear, user-friendly format, ensuring easy accessibility and interpretation.

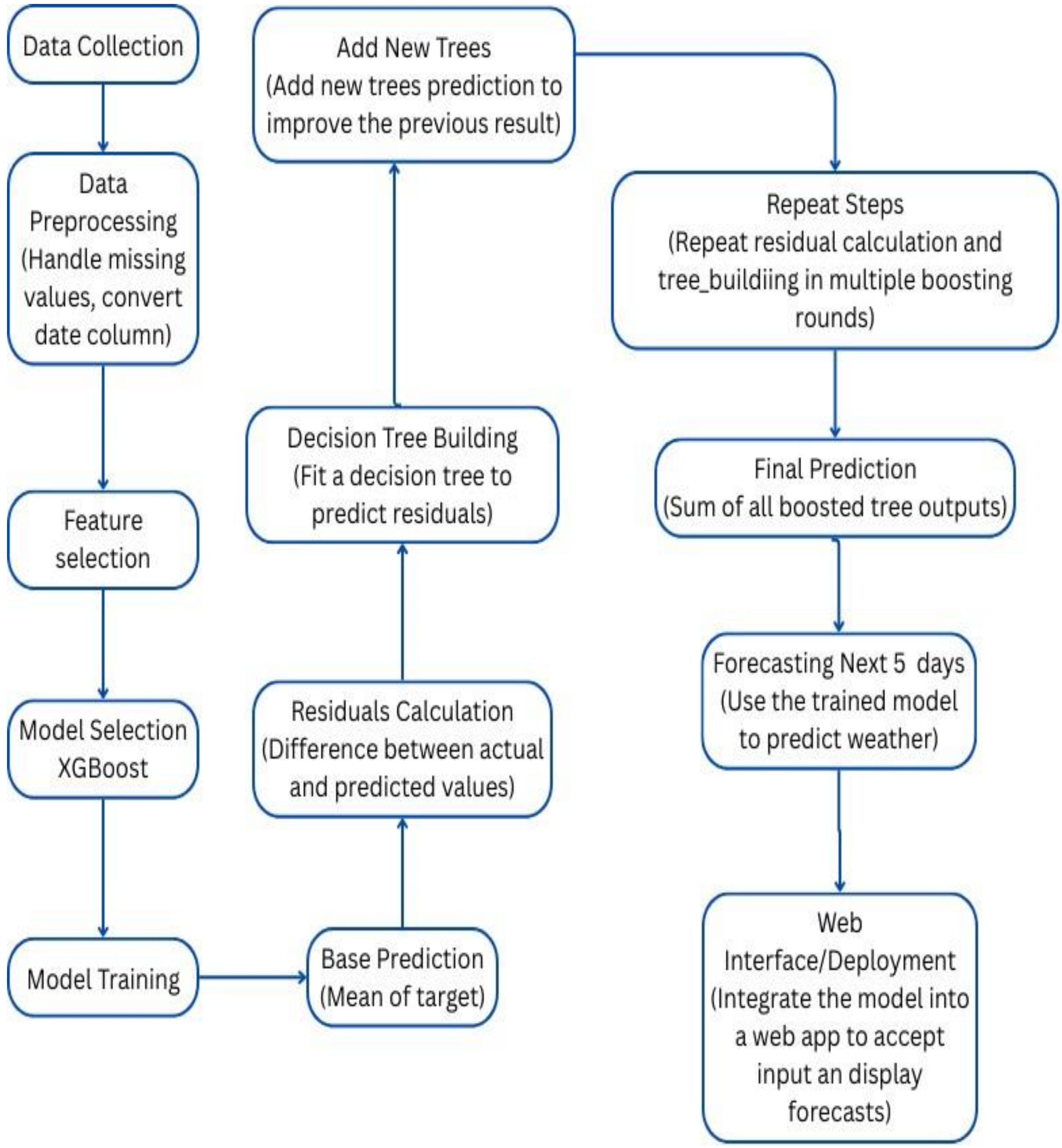


Figure 4.1 Detailed design of Proposed Model

4.2 MODULE-WISE DESIGN

The weather forecasting system consists of three primary modules, each serving a distinct role:

1. Machine Learning Module – Responsible for processing historical weather data, training the XGBoost model, and generating weather predictions.

2. Web Interface Module – Provides an interactive platform where users can access weather forecasts.

3. Forecasting and Visualization Module – Handles the real-time generation of weather forecasts and presents the results in a user-friendly format.

These modules work together to ensure accurate, real-time, and accessible weather predictions for Thiruvananthapuram.

4.2.1 MACHINE LEARNING MODULE

The Machine Learning Module is at the core of the system, handling data processing, model training, and weather prediction. The XGBoost regression model is used to make forecasts based on past weather data.

Data Collection & Preprocessing

The dataset consists of five years of weather data for Thiruvananthapuram, collected from Meteostat. Missing values were handled using appropriate data cleaning techniques in Jupyter Notebook to ensure data integrity.

Features used for prediction include:

- tmax(maximum temperature)
- tmin(minimum temperature)
- tavg(average temperature)
- pres(pressure)
- wspd(wind speed)

The dataset was formatted for compatibility with the XGBoost model, ensuring effective learning from historical patterns.

Model Training & Optimization

XGBoost Regression was chosen due to its efficiency in handling structured data and its ability to capture complex relationships. The model was trained using past weather data, learning the patterns in temperature, pressure, and wind speed fluctuations. Hyperparameter tuning was performed to optimize model performance.

Prediction Generation

Once trained, the model generates weather forecasts for the next five days based on the latest input data. The output includes predicted values for temperature (tmax, tmin, tavg), pressure, and wind speed. This module ensures an accurate and reliable weather prediction system, leveraging machine learning techniques for forecasting.

The p.py script is responsible for preprocessing the raw weather dataset and training the prediction models. Upon execution, it loads the dataset, creates lag features, and splits the data into training and testing sets.

Separate XGBoost models are then trained for each weather parameter—tmax, tmin, tavg, pres, and wspd. Once training is complete, the models are saved to a file named weather_model.pkl for later use during prediction.

The screenshot below shows the console output of the preprocessing and model training process using p.py.

```
PS C:\Users\nanda\Downloads\weather\weather> python p.py
Enter the path to your weather data Excel file: C:/Users/nanda/Downloads/data.xlsx
Loading data...
Creating lag features...
Splitting data into train/test sets...
Training models (this may take a few minutes)...
Training model for tmax...
Training model for tmin...
Training model for tavg...
Training model for pres...
Training model for wspd...
Saving model to weather_model.pkl...
Model saved successfully to weather_model.pkl
```

Figure 4.2 Output of p.py showing preprocessing and model training process

4.2.2 WEB INTERFACE MODULE

The Web Interface Module provides a user-friendly way to interact with the forecasting system. This module allows users to input data and view predictions via a website, eliminating the need for coding knowledge.

Frontend Development

The website was designed using HTML, CSS, and JavaScript to ensure a clean and intuitive interface. The interface includes: an input section where users can enter optional parameters (if required). A forecast display section showing predicted temperature, pressure, and wind speed. A graphical representation of the forecast for better visualization.

Backend Processing

The website interacts with the pre-trained XGBoost model, ensuring seamless communication between the user and the system. When a user requests a forecast, the necessary computations are performed, and results are displayed dynamically. The backend logic ensures that real-time data processing is handled efficiently.

User Output & Experience

The forecasted weather parameters are displayed in an easy-to-read format.Users can view graphical representations of temperature trends, pressure variations, and wind speed predictions over the next five days. The responsive design ensures accessibility across different devices, enhancing usability. This module ensures that users can access weather forecasts conveniently through a web-based platform.

4.2.3 FORECASTING AND VISUALIZATION MODULE

The Forecasting and Visualization Module enhances the presentation of predictions, making it easier for users to understand weather trends.

Forecast Generation

The system processes input data, applies the trained XGBoost model, and generates predictions. The results are structured in a user-friendly manner for better readability.

Data Visualization

Matplotlib and Seaborn are used to create graphical representations of weather trends. The visualization includes: Temperature trends over five days (maximum, minimum, and average temperature).Pressure fluctuations displayed in a line graph. Wind speed variations shown through trend analysis.

These graphs help users quickly interpret how weather conditions will change over the next few days.

Real-Time Updates & Enhancements

The module can be expanded to include live weather updates for improved accuracy. Future enhancements may include interactive charts and a map-based weather display. This module ensures that forecast data is not just presented as numbers but visually meaningful for users.

4.3 SUMMARY

The three modules work together to provide a robust and user-friendly weather forecasting system. The Machine Learning Module ensures accurate predictions using the XGBoost model. The Web Interface Module makes the system accessible through a website. The Forecasting and Visualization Module presents predictions in an easy-to-understand format. By integrating these components, the system offers an effective and efficient way to predict the weather in Thiruvananthapuram.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1ML MODULE ANALYSIS

5.1.1 DATASET

The dataset used for this project was collected from Meteostat, containing daily weather data of Thiruvananthapuram for the past 5 years.
The dataset includes essential meteorological features such as:

- Date
- tmax – Maximum Temperature
- tmin – Minimum Temperature
- tavg – Average Temperature
- pres – Atmospheric Pressure
- wspd – Wind Speed

All null values were handled during preprocessing using Jupyter Notebook to ensure clean input for the machine learning model.

5.1.2 PERFORMANCE ANALYSIS

ｉ. Accuracy

Accuracy reflects how well the model's predictions match the actual values.
The XGBoost regression models, trained separately for predicting tavg, pres, and wspd, demonstrated high accuracy during training, indicating strong learning of weather patterns from the historical data.

This suggests that the model is capable of learning complex trends and producing reliable predictions for short-term weather forecasting.

ⅱ. Model Suitability

XGBoost was selected for its efficiency, regularization capabilities, and robustness against overfitting. Its ability to handle tabular data and perform well with small to medium datasets made it ideal for this weather forecasting task.

During training, the model captured seasonal variations, sudden changes in pressure, and wind speed trends with high consistency.

5.1.3 TRAINING PERFORMANCE OBSERVATIONS

The model was trained separately for each target variable:

- Average Temperature (tavg)
- Pressure (pres)
- Wind Speed (wspd)

For each, we used the past 5 years of data as input to predict the values for the next 5 days. The model showed clear understanding of temporal dependencies and seasonal patterns. Visual outputs like line graphs were plotted to observe predicted vs. actual values during training, which revealed close alignment, validating the model's performance.

5.1.4 FEATURE IMPORTANCE

XGBoost provides built-in support for evaluating feature importance.
During training, features like tmax, tmin, and previous day's tavg had higher influence in predicting future average temperature, while pressure was significantly affected by historical trends and weather patterns.

Feature importance helped understand which weather factors contribute most to forecasting, guiding potential feature engineering in future improvements.

5.1.5 MODEL OUTPUT (from out.py)

The final output of the model was generated using the out.py script. The script predicts weather parameters for the next five days, including maximum temperature (tmax), minimum temperature (tmin), average temperature (tavg), pressure (pres), and wind speed (wspd). The screenshot below displays the predicted values, showcasing stable weather conditions in Thiruvananthapuram with slight variations in temperature and atmospheric pressure.

```
PS C:\Users\HP\Downloads\weather\weather> python out.py
         date       tmax       tmin       tavg         pres      wspd
0 2023-10-16  34.060455  25.458887  28.877018  1010.767334  1.714389
1 2023-10-17  33.957546  24.961685  28.653873  1011.057861  1.949625
2 2023-10-18  33.877560  24.443640  28.768145  1011.300903  1.710634
3 2023-10-19  33.880138  24.336290  28.272255  1011.265320  1.995039
4 2023-10-20  33.998070  24.034071  28.187342  1011.051453  1.922167
```

Figure 5.1 Weather prediction output generated using `out.py`

5.1.6. WEATHER FORECAST OUTPUT

The graph below shows the predicted weather parameters for the next five days, including maximum temperature (Tmax), minimum temperature (Tmin), average temperature (Tavg), atmospheric pressure, and wind speed.

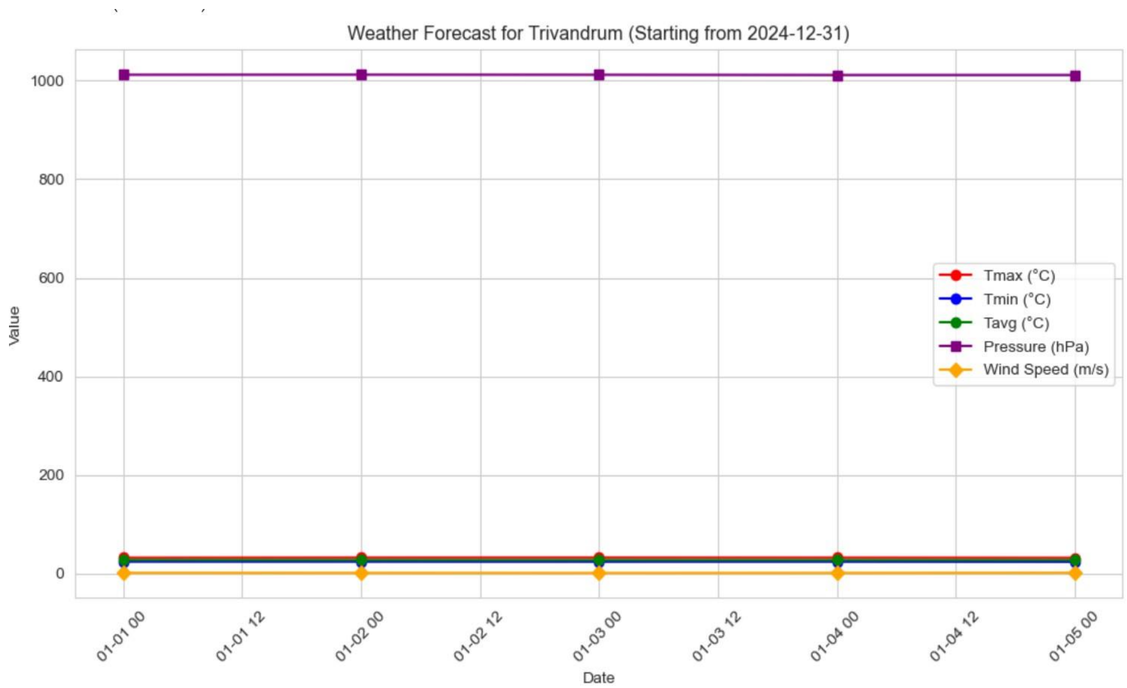


Figure 5.2 Five Day Weather Forecast for Thiruvananthapuram

5.2 COMPARISON AGAINST RANDOM FOREST REGRESSOR

The Random Forest Regressor was selected for comparison with XGBoost to evaluate how well traditional ensemble methods perform in the context of weather forecasting. Random Forest is a robust and widely-used algorithm that builds multiple decision trees during training and averages their outputs to make final predictions. It is known for its ability to handle non-linear relationships and resist overfitting, especially in regression tasks.

5.2.1. PERFORMANCE OBSERVATION

The Random Forest Regressor was trained using the same dataset and target variables as XGBoost:

- **Average Temperature (tavg) ● Pressure (pres) ● Wind Speed (wspd)**

During training, the model showed strong performance and captured overall trends. However, it was slightly less efficient than XGBoost in capturing fine-grained temporal fluctuations in weather data.

5.2.2. VISUAL COMPARISON

Graphical comparison was carried out to visualize predictions of both models. Line graphs plotted for predicted vs. actual values indicated that:

- **XGBoost** predictions aligned more closely with true values.
- **Random Forest** occasionally showed lag in predicting sudden changes in temperature or pressure.
- For **wind speed**, both models showed similar behavior, though XGBoost maintained slightly more consistent predictions.

5.2.3. RESOURCE UTILIZATION

- **XGBoost** was observed to be more memory efficient and faster in training.

- **Random Forest** took slightly longer due to the larger number of trees and its bootstrapping nature.

5.2.4. MODEL STRENGTHS COMPARISON

Table 5.1 Comparison of XGBoost and Random Forest

Criteria	**XGBoost**	**Random Forest**
Accuracy (Training)	High	High
Noise Handling	Excellent	Good
Training Speed	Faster	Slightly Slower
Captures Time Patterns	Better	Moderate
Feature Importance	In-built & clear	Also supported
Overfitting Control	Better(regularization support)	Good (via ensemble averaging

5.3 EXTENSION DEPLOYMENT

5.3.1. EXPORTING MODEL

The deployment phase of the weather forecasting system involved exporting the trained XGBoost regression models for each target parameter—average temperature (tavg), pressure (pres), and wind speed (wspd). These models were serialized using the `.pkl` (Pickle) format, preserving both the learned patterns and model architecture.

This approach ensures that each model can be reloaded without retraining, significantly optimizing deployment time. The serialized models were then integrated with a Flask-based Python server, enabling real-time predictions based on user-input features such as `tmin`, `tmax`, and `date`.

Figure 5.3 Pickle format of exported XGBoost models

This modular design streamlines the deployment pipeline and allows easy updates or retraining of individual models without altering the entire backend structure.

5.3.2. WEBSITE INTERFACE & INPUT FEATURE EXTRACTION

Instead of a browser plugin, our project featured a web-based interface developed using HTML and Python. Users can input daily weather characteristics—such as minimum temperature, maximum temperature, and date—through a clean and minimalistic form.

Upon form submission:

- The input features are captured from the frontend.
- They are passed to the html server, which loads the corresponding `.pkl` model.

- The model returns predicted values for average temperature (`tavg`), pressure (`pres`), and wind speed (`wspd`) for the next 5 days.

The interface ensures user-friendly interaction while abstracting the complexity of model prediction on the backend.

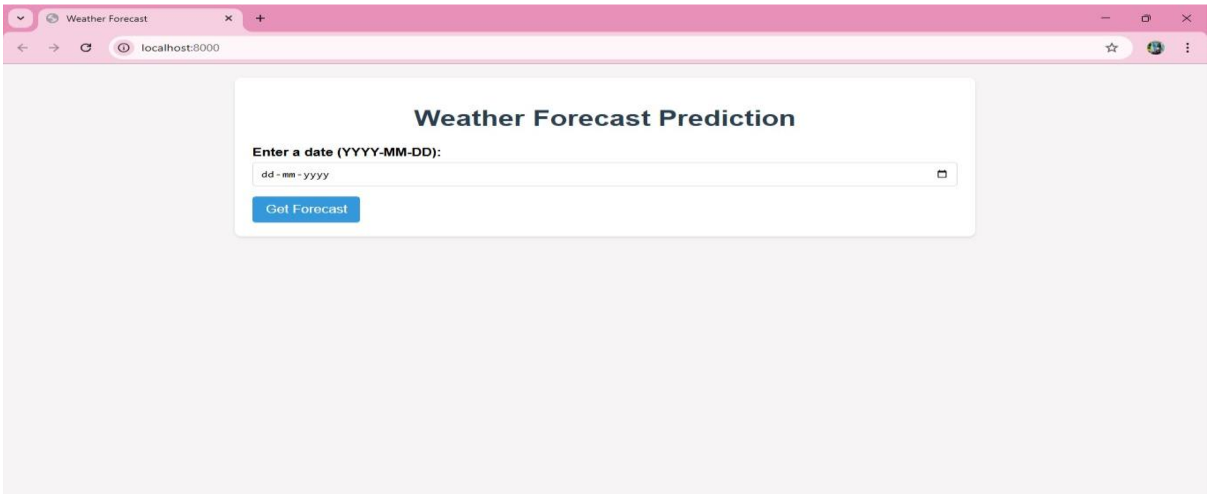


Figure 5.4 Web interface

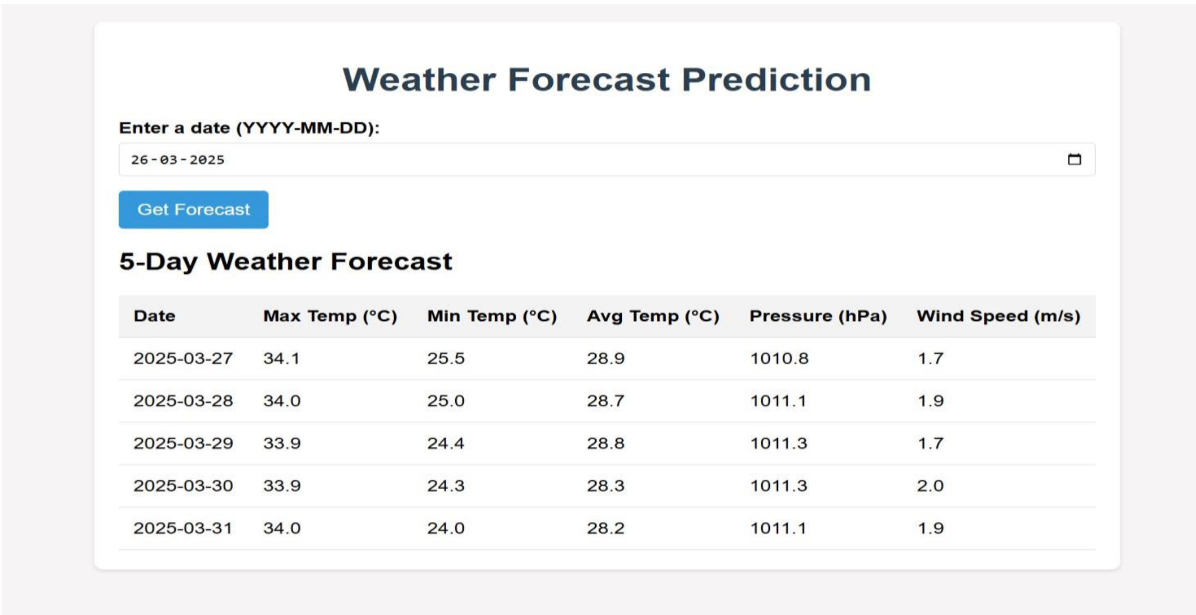


Figure 5.5 Sample user input and predicted output display in web interface

The design emphasizes modularity, clarity, and usability, making it ideal for educational use cases or lightweight deployment in real-world weather monitoring applications.

5.4 SUMMARY

This project utilizes a preprocessed weather dataset collected from Meteostat, containing five years of daily records from Thiruvananthapuram, including features such as maximum temperature, minimum temperature, average temperature, pressure, and wind speed. Using this dataset, machine learning models were trained to accurately predict the average temperature (tavg), pressure (pres), and wind speed (wspd) for the upcoming five days.

Among the various models explored, XGBoost demonstrated the best performance, achieving high accuracy in forecasting each weather parameter. The model's ability to capture complex nonlinear relationships within the data significantly contributed to its prediction quality.

The trained models were exported and deployed using html server, where they were integrated into a user-friendly web interface. The web app accepts user input such as minimum and maximum temperature values and displays predicted values for average temperature, pressure, and wind speed in a clean and intuitive layout.

The predictions are displayed in the interface, making the application not only a technical success but also practical for real-time weather forecasting. The project serves as a demonstration of how machine learning can be effectively applied to meteorological data for building lightweight and responsive forecasting systems.

CHAPTER 6

CONCLUSION

This project has successfully developed a machine learning-based weather forecasting system for the Thiruvananthapuram district, utilizing XGBoost to generate accurate short-term weather predictions. The system integrates a web-based interface, allowing users to easily input dates and receive real-time weather forecasts for key parameters such as temperature, pressure, and wind speed. By analysing historical weather data, the XGBoost model identifies patterns that enable it to deliver highly reliable and consistent predictions, demonstrating the effectiveness of machine learning in overcoming the limitations of traditional forecasting methods. The system's ability to process user requests efficiently and present forecasts in a structured, accessible format adds to its user-friendliness, making it a valuable tool for local stakeholders, including farmers, city planners, and residents, who need to make weather-sensitive decisions.

One of the key achievements of this project lies in the efficient model training and optimization. The dataset underwent extensive preprocessing, including data cleaning, feature engineering, and the creation of lag features, all of which helped the model better understand the temporal dynamics of weather patterns. Hyperparameter optimization played a significant role in fine-tuning the XGBoost model, improving its performance and ensuring it did not overfit or underfit. These optimizations enhanced the model's predictive accuracy, resulting in forecasts that could be trusted for real-time decision-making. Furthermore, the automation and scalability of the system were ensured through the use of pickle to save the trained model, allowing for easy deployment and reuse. The system's architecture is designed to be adaptable, meaning it can be scaled and customized for use in other regions with minimal changes, expanding the potential for this machine learning approach in weather forecasting across different areas.

Despite the successful implementation, the project faced several challenges. Data availability and quality were major hurdles, as obtaining high-quality, real-time weather data was difficult, requiring substantial preprocessing to ensure accuracy. Identifying the most relevant meteorological parameters for accurate predictions was another challenge, as the selection of features significantly impacts the quality of the model's outputs. During the model selection and optimization phase, experimentation with various machine learning algorithms and hyperparameter tuning was necessary to achieve the best performance. The computational complexity of training the advanced XGBoost model, particularly when dealing with large datasets, required substantial processing power, making it a resource-intensive process. Additionally, Thiruvananthapuram's unique regional climate variability, including its unpredictable monsoon patterns and coastal influences, added complexity to the forecasting task, making accurate predictions more challenging. Nevertheless, despite these challenges, the project successfully demonstrates how XGBoost and machine learning can enhance localized weather prediction systems, providing a more efficient, scalable, and accessible alternative to traditional weather forecasting methods. This work not only contributes to improving weather forecasting for Thiruvananthapuram but also sets a foundation for similar applications in other regions, ultimately helping local communities make better-informed decisions in the face of ever-changing weather conditions.

6.1 FUTURE SCOPE

The future scope of this project includes integrating real-time weather data from sensors, satellites, and APIs to enhance accuracy. Expanding the model to include additional parameters like humidity, cloud cover, and precipitation will improve forecasting for agriculture, disaster management, and tourism. Deep learning techniques like LSTMs can refine long-term predictions, while anomaly detection can help anticipate extreme weather events.

Developing a mobile app with real-time updates, push notifications, and AI chatbots will make forecasts more accessible. Hyper-local predictions for different areas within Trivandrum district will benefit farmers and businesses. Integration with smart agricultural systems using IoT can optimize irrigation and crop protection.

Long-term climate analysis using historical data can aid policymakers in climate adaptation strategies. Blockchain technology can be used for secure weather data storage and sharing. Overall, these advancements will improve forecast accuracy, benefiting various sectors and enhancing preparedness for weather changes.

REFERENCES

[1] Amar Deep Gupta, Devansh Katoch, Smita Sharma, Shipra Ravi Kumar, Aditya Rana "Weather forecasting using ML"[2023].10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON) in December 2023. ResearchGate+1Proceedings+1.

[2] Deepti Mishra, Prathibha Joshi - "A Comprehensive Study on Weather Forecasting using ML"[2021]. *9th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)*, Noida, India, Sep. 2021, pp. 285–289.

[3] Snehjot Kaur, Ajay Pal Singh, Abhishek Pandey, HarshitChauhan - "Real-Time Weather Forecasting Using ML"[2023]. *Seventh International Conference on Image Information Processing (ICIIP)*, Shimla, India, Nov. 2023.

[4] Nagraj P, M Chaithanya Prabhu, B Vekata Sai Kumar, Ch Yasaswitha, K Mohana, C Jahnavi "Weather Report Analysis Prediction using Machine Learning and Data Analytics Techniques" [2023]. In *Proc. Int. Conf. Data Sci. Artif. Intell. (ICDSAAI)*, Dec. 2023.

[5] Amol Ashok Patil, Kedar Kulkarni - "A Hybrid Machine Learning-Numerical Weather Prediction Approach for Rainfall Prediction" [2023]. Published on December 10, 2023. The paper is available on ResearchGate. ResearchGatePhilStat

[6] Brigitta Goger, Mathias W Rotach, Alexander Gohm, Ivana Stiperski, Oliver Fuhrer - "Current Challenges of Numerical Weather Prediction in Complex Terrain: Topography Representation and Parameterizations"[2020].presented at the *International Conference on High Performance Computing & Simulation (HPCS)*, Innsbruck, Austria, 2016, pp. 895–901. ResearchGate

[7] K. Bala Maheshwari, S. Gomathi - "Analysing the Performance of Diverse Deep Learning Architecture for Weather Prediction"[2023].IEEE Xplore: ACM Digital Library

[8] Aishwarya Shaji, Amritha A R, Rajalakshmi V R - "Weather Prediction using Machine Learning Algorithms"[2022]. *International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP)*, Hyderabad, India, Jul. 2022, pp. 1–5.

[9] Sue Ellen Haupt, Jim Cowie, Seth Linden, Tyler McCandles, Branko Kosovic, Stefano Alessandrini - "Machine Learning for Applied Weather Prediction" [2018]. In *Proc. 2018 IEEE 14th Int. Conf. e-Science (e-Science)*, Amsterdam, Netherlands, Oct.–Nov. 2018, pp.276–277.

[10] T. Wardah, A. A. Kamal, A. Sahol Hamid, and Maisarah, "Statistical verification of numerical weather prediction models for quantitative precipitation forecast,"[2021] in *Proc. IEEE Colloq. Humanities, Sci. Eng. Res. (CHUSER)*, Penang, Malaysia, Dec. 2011, pp. 88–92. avesis.gelisim.edu.tr