CLIMATE PREDICTION USING ARIMA NETWORK IN DEEP LEARNING

CO8811 – PROJECT REPORT

Submitted by

GOKULA VIGNESH S	**211420118018**
GOWRI SANKAR P	**211420118019**
JAFFRIN GLASTO J	**211420118027**

in partial fulfillment for the award the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER AND COMMUNICATION ENGINEERING

PANIMALA``R ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

MARCH 2024

BONAFIDE CERTIFICATE

Certified that this project report **"CLIMATE PREDICTION USING ARIMA NETWORK IN DEEP LEARNING"** is the bonafide work of **GOKULA VIGNESH S (211420118018), GOWRI SANKAR P (211421118019), JAFFRIN GLASTO J (211421118027)** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.B.ANNI PRINCY M.E., Ph.D.,	Dr.R.ANAND BABU.,M.TECH.,Ph.D.,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR, COMPUTER AND COMMUNICATION ENGINEERING, PANIMALAR ENGINEERING COLLEGE, NAZARATHPETTAI, POONAMALLEE, CHENNAI- 600123.	ASSISTANT PROFESSOR, ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING, PANIMALAR ENGINEERING COLLEGE, NAZARATHPETTAI, POONAMALLEE, CHENNAI- 600123.

Certified that the above candidate(s) was/ were examined in the End Semester Project Viva-Voce Examination held on...........................

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors Tmt. **C.VIJAYARAJESWARI, Dr. C. SAKTHIKUMAR, M.E., Ph.D**., and **Dr.SARANYASREE SAKTHIKUMAR B.E., M.B.A., Ph.D**., for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D.,** for his timely concern and encouragement provided to us throughout the course.

We thank our HOD of Computer and Communication Engineering Department, **Dr. B. ANNI PRINCY, M.E., Ph.D**., Professor, for the support extended throughout the project.

We would like to thank our supervisor, **Dr.R.ANAND BABU.,M.TECH.,Ph.D.,** Assistant Professor, and all the faculty members of the Department of Computer and Communication Engineering for their advice and suggestions for the successful completion of the project.

GOKULA VIGNESH S
GOWRI SANKAR P
JAFFRIN GLASTO J

ABSTRACT

The importance of seismological research around the globe is very clear. Therefore, new tools and algorithms are needed in order to predict temperature, humidity, pressure, wind speed and manual location in collecting data set from internet , as well as to discover relations that allow us to better understand this phenomenon and thus be able to save countless human lives. However, given the highly random nature of global warming and the complexity in obtaining an efficient mathematical model, efforts until now have been insufficient and new methods that can contribute to solving this challenge are needed. In this work, a temperature, humidity, pressure, wind speed and manual location in collecting data prediction method is proposed, which is based on the composition of a known system whose behavior is governed according to the measurements of more than two decades of seismic events and is modeled as a time series using deep learning, specifically a network architecture based on ARIMA (Auto Regressive Integrated Moving Average) cells. The increased severity and frequency of Climate-Induced Disasters (CID) have been testing the resilience of cities worldwide. To predict the occurrence of CID, this study links different climate change indices to historical disaster records.. Additionally, the study highlights the association between flood disasters and temperature-related features, such as daily temperature gradient and the number of days with temperature below zero. Our proposed model has over 96% results which is better when compared to the previous models In response to climate challenges, a novel zero-shot learning approach is proposed to forecast various climate measurements at new and unmonitored locations. This method surpasses conventional forecasting techniques by leveraging knowledge extracted from other geographic locations. Specifically, it excels in predicting climate variables.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ARIMA	-	Auto Regressive Integrated Moving Average
CNN	-	Convolutional Neural Network
DNN	-	Deep Neural Network
RNNs	-	Recurrent Neural Networks
SGD	-	Stochastic Gradient Descent
NCAR	-	National Centre for Atmospheric Research
ECMWF	-	European Centre for Medium-Range Weather Forecasts
GISS	-	Goddard Institute for Space Studies
HTML	-	HyperText Markup Language
CSS	-	Cascading Style Sheet

CHAPTER 1

INTRODUCTION

1.1 Introduction

Climate change poses one of the most critical challenges to our planet in the present era. The continuous increase in global temperatures is leading to a higher frequency of extreme weather phenomena, causing profound disruptions in ecosystems worldwide. The need to accurately predict and effectively adapt to these ongoing changes has reached unprecedented levels. Through this important climate prediction endeavor, we aim to delve deeply into the intricate science guiding climate modeling, carefully examine historical data trends, and explore potential future scenarios on a global scale. Our overarching objective revolves around contributing significantly to the creation of a sustainable future by enhancing our understanding of climate change impacts and working towards effective mitigation strategies. Indeed, the essence of our exploration lies in the sophisticated climate models that proficiently simulate the intricate systems operating within our planet.

These advanced models serve as vital tools in predicting forthcoming temperature fluctuations, the rise in sea levels, as well as the altering patterns of precipitation across various regions. Understanding the specific regional implications of climate change is of paramount importance in our analysis. Climate change is a pressing global issue with far-reaching consequences. Climate change disrupts normal weather patterns, leading to more frequent and severe extreme weather events. Hurricanes, typhoons, and floods become more common, causing loss of life, habitat destruction, and property damage. As temperatures rise, glaciers melt, and ice sheets shrink, contributing to higher sea levels. Low-lying areas, islands, and coastlines face flooding, endangering lives and ecosystems. Warming oceans become more acidic, resulting in the loss of coral reefs. These reefs play a crucial role in protecting shorelines from waves and storms, and they support a quarter of the ocean's species. Climate change exacerbates poverty, affects livelihoods, and strains economies. Heat-related health issues, crop failures, and disruptions to food production all contribute to the negative impact on human well-being and financial stability. In summary, addressing climate change is essential to mitigate these adverse effects and protect our planet and its inhabitants. By meticulously analyzing local data points, identifying vulnerable areas, and formulating targeted adaptation strategies, we strive to create resilience in the face of environmental challenges. Deep learning has revolutionized various fields, including weather predictions. Deep learning models have shown particular promise in nowcasting, which involves predicting weather conditions up to 2-6 hours ahead .Previous work focused on using direct neural network models for weather data, extending forecasts from 0 to 8 hours with architectures like LSTM , and even generating continuations of radar data for up to 90 minutes ahead.

These models learn directly from observed data, bypassing the need for explicit physical laws, and can compute predictions faster than traditional physics-based techniques. In a recent advancement, the ARIMA model pushes the boundaries of neural precipitation forecasting to 12-hour predictions while maintaining a spatial resolution of 1 km and a time resolution of 2 minutes. By quadrupling the input context, adopting a richer weather input state, and capturing longer-range spatial dependencies, ARIMA significantly outperforms its predecessor, ARIMA. Compared to physics-based models, ARIMA surpasses the state-of-the-art HREF ensemble model for weather forecasts up to 12 hours ahead. Another approach is the Deep Learning Weather Prediction (DLWP) model, which uses deep convolutional neural networks (CNNs) for globally gridded weather prediction. DLWP CNNs directly map historical observations of weather variables to their future states, enabling accurate predictions for time intervals like 6 hours. Deep learning models can be trained to find weather patterns by analyzing satellite imagery. Instead of simulating entire weather systems, these models focus on visual patterns from a mosaic of pixels, enhancing our understanding of cloud behavior and aiding in weather prediction. In summary, deep learning empowers us to make more accurate and timely weather forecasts, benefiting various aspects of our daily lives, from planning activities to managing critical systems like transportation and energy grids. Our approach is rooted in leveraging this knowledge to engage in strategic discussions around reducing greenhouse gas emissions, promoting the use of renewable energy sources, and conserving our precious natural resources.

By accurately foreseeing potential future climate scenarios, we empower decision-makers across various sectors to plan and implement infrastructure projects, allocate resources efficiently, and devise policies of remarkable effectiveness. Foresight gained from precise climate predictions enables proactive preparations for extreme events, thereby helping communities brace themselves against calamitous hurricanes, floods, and prolonged droughts. This proactive stance in issuing early warnings allows for prompt preventive measures to be taken, effectively minimizing the impact on both lives and property. WRF(weather research and forecasting) is used to predict the wind field in Beijing-Tianjin-Hebei region in previous years with the historical data from the European Centre for Medium-Range Weather Forecasts.

The Weather Research and Forecasting (WRF) Model is a versatile numerical weather prediction system that caters to both atmospheric research and operational forecasting needs. WRF allows scientists to simulate atmospheric conditions using either real data (such as observations and analyses) or idealized scenarios. This flexibility enables detailed investigations into local weather patterns, complex phenomena, and atmospheric processes. Advancing Meteorology Researchers use WRF to push the boundaries of fine scale atmospheric simulation. By incorporating cutting-edge physics and numerics, they enhance our understanding of weather dynamics and improve forecasting accuracy. WRF serves as a robust platform for operational forecasting. Meteorologists can tailor forecasts to specific regions, adjusting scales from tens of meters to thousands of kilometers. The model integrates real-time observations through data assimilation techniques, enhancing forecast reliability. WRF benefits from global collaboration, with contributions from researchers worldwide. This collaborative spirit ensures continuous model improvements.

In summary, WRF bridges the gap between scientific research and practical weather prediction, making it an indispensable tool for atmospheric scientists and forecasters alike. Secondly, the datasets including temperature, humidity, pressure, wind speed and direction are obtained from the simulation results, which are treated as the characteristic data for the input of deep learning.The result of our work is to find the impact of global warning using ARIMA. We have successfully completed our research and are able to find out the impact of the global warning . We have accuracy of by using Deep Learning techniques and using ARIMA network. The depicts the predicted and actual values on ARIMA model explains the loss graph of trained and test data.

1.2 Existing System

Natural disasters are without any doubt a latent danger and become very devastating and threaten the entire ecosystem of one region. That is why the prediction of earthquakes plays such an important role since its goal is to specify the magnitude and geographical and temporary location of future earthquakes with enough precision and anticipation to issue a warning. Despite the efforts made to produce mechanical or computational models of the earthquake process, these still do not achieve real predictive power. Given the highly random nature of earthquakes with relatively high magnitude, their occurrence can only be analyzed using a statistical approach, but any synthetic model must show the same characteristics with respect to its distribution in size, time, and space, which is very hard to achieve. A lot of research work has been conducted in identifying how weather as a factor affects agriculture. But most of these studies require large complex information which is not directly available. The author states that obstacles faced for agriculture are usually Technical or Organizational problems. To predicting the impact of extreme weather events and mitigating its effect on global finance.

1.3 Proposed System

In this research we will create a ARIMA NETWORK cells. which will take the parameters as features and from that we will try to classify the damage made by the global warning in a particular region. At initial we will have six nodes and they will be predicting temperature, humidity, pressure, wind speed and manual location in collecting data set and Historical data. We will have three nodes as output Low damage, Moderate damage and High damage.

Compared with the traditional methods, machine learning model does not need to build control equations to describe the atmospheric motion, which can greatly reduce the time required for prediction. Machine learning model can analyze meteorological data to achieve the purpose of prediction and is more and more used in the of wind field prediction. The weather data was processed using ARIMA, keeping precipitation. The intensity of sunshine and temperature at ground level as variables to calculate the daily mean and monthly mean. We have extracted the top three crops giving the best yield throughout the year. The temperature, rainfall, wind speed and humidity of the region at which the crop had given such stellar outputs is also displayed for time series data.

1.4 Problem Statement

Rainfall forms the primary input to the river basin, affecting the water capacity a stream, particularly during the torrential rainfall event. Moreover, one of the major focuses of climate change study is to understand whether there is an extreme change in the occurrence and frequency of heavy rainfall events. The accuracy level of the ML models used in predicting rainfall based on historical data has been one of the most critical concerns in hydrological studies. An accurate ML forecasting model could give early alerts of severe weather to help prevent natural disasters and destruction. Hence, there are needs to develop ML algorithms capable in predicting rainfall with acceptable level of precision and in reducing the error in the dataset of the projected rainfall from climate change model with the expected observable rainfall.

1.5 Objectives

The proposed ARIMA model time's series deep learning scheme provides better prediction with 99% accuracy. The proposed scheme provides annual rainfall prediction and four seasonal rainfall predictions such as summer, autumn, rainy, and winter. The ARIMA trained on 1400 epoch for revealing very good results. Linear regression is acted as the best optimizer. Activation function is applied for providing the best deep learning models than other activation functions. The ARIMA model revealed better accuracy in all of the seasonal and annual rainfall.

1.6 Methodology

1.6.1 Deep Learning

Deep learning in climate prediction involves using neural networks to analyze large amounts of climate data to make predictions about future climate patterns. The methodology typically involves the following steps:

Data Collection

Collecting historical climate data from various sources such as satellites, weather stations, and climate models. This data typically includes temperature, precipitation, humidity, wind speed, and other relevant variables.

Data Preprocessing

Preprocessing the data to remove noise, handle missing values, and normalize the data to make it suitable for training the neural network.

Model Selection:

Choosing an appropriate deep learning model architecture for the climate prediction task. This may include convolutional neural networks (CNNs) for image-based data (e.g., satellite images), recurrent neural networks (RNNs) for sequential data (e.g., time series data), or a combination of both.

Model Training

Splitting the data into training, validation, and test sets, and then training the deep learning model on the training set. The model is optimized using techniques such as stochastic gradient descent (SGD) and backpropagation.

Model Evaluation

Evaluating the trained model using the validation set to ensure it is performing well and adjusting hyperparameters if necessary.

Prediction

Using the trained model to make predictions on new or unseen data to forecast future climate patterns.

Post-processing

Post-processing the model predictions to improve their accuracy and make them more interpretable for stakeholders.

Visualization: Visualizing the model predictions and communicating the results to stakeholders using graphs, maps, and other visualization techniques.

It's important to note that deep learning models for climate prediction are just one part of a broader methodology that also includes traditional climate models, physical understanding of climate processes, and expert knowledge in the field of climatology.

1.6.2 Neural Network

Neural networks are increasingly being used in climate prediction due to their ability to handle complex, non-linear relationships in data. Here's a general methodology for using neural networks in climate prediction:

Data Collection

Collect historical climate data from various sources, including temperature, precipitation, humidity, wind speed, and other relevant variables. This data is typically collected over a long period to capture seasonal and long-term trends.

Data Preprocessing

Clean the data to remove noise and outliers, and preprocess it for use in the neural network. This may involve normalizing the data, handling missing values, and encoding categorical variables.

Feature Selection/Extraction

Select relevant features (input variables) that are likely to have a significant impact on climate prediction. This can be done manually based on domain knowledge or using automated feature selection techniques.

Model Selection

Choose a suitable neural network architecture for climate prediction. This could be a simple feedforward neural network, a convolutional neural network (CNN) for spatial data, or a recurrent neural network (RNN) for temporal data.

Model Training

Split the data into training, validation, and test sets. Train the neural network using the training set and validate it using the validation set. Adjust hyperparameters such as learning rate, batch size, and number of epochs to optimize the model performance.

Model Evaluation

Evaluate the trained model using the test set to assess its performance. Common evaluation metrics for climate prediction include mean squared error (MSE), mean absolute error (MAE), and correlation coefficient.

Prediction

Once the model is trained and evaluated, use it to make predictions on new data. Monitor the model's performance over time and retrain it periodically to maintain its accuracy.

Uncertainty Estimation

Climate prediction is inherently uncertain due to the complex nature of climate systems. Consider using techniques such as dropout or ensemble to estimate uncertainty in the predictions.

Post-processing

Post-process the model outputs if necessary to derive useful information for decision-making. This could involve aggregating predictions over regions or time periods, or translating them into actionable insights.

Communication

Communicate the model results and uncertainties effectively to stakeholders, policymakers, and the public to support informed decision-making. This methodology provides a general framework for using neural networks in climate prediction, but specific implementations may vary based on the nature of the data and the goals of the prediction task.

1.6.3 ARIMA Network

ARIMA (Autoregressive Integrated Moving Average) is a widely used time series forecasting method with applications in climate prediction. In the context of climate science, ARIMA models can be valuable tools for understanding and predicting climate patterns and trends over time. The methodology for using ARIMA in climate prediction typically involves several key steps. First, historical climate data, such as temperature, precipitation, and humidity, is collected and preprocessed to ensure its suitability for time series analysis. This preprocessing may include removing noise, handling missing values, and checking for stationarity.

Next, the parameters of the ARIMA model are identified based on the characteristics of the data, including the order of the autoregressive (AR), differencing (I), and moving average (MA) components. These parameters are then estimated using the historical data, often through techniques like maximum likelihood estimation. The ARIMA model is then validated using a validation set or cross-validation to assess its performance. Once validated, the model can be used to make predictions for future climate data. It's important to monitor the model's performance over time and update it periodically as new data becomes available.

One of the advantages of ARIMA models is their ability to provide point forecasts, which can be valuable for short to medium-term climate prediction. However, ARIMA models may not capture the complex, nonlinear relationships present in climate data as effectively as other models, such as neural networks.

In conclusion, while ARIMA models can be useful tools in climate prediction, especially for short to medium-term forecasting, it's essential to consider their limitations and compare their performance with other forecasting approaches to determine the most suitable model for a given climate prediction task.

1.6.4 Linear Regression

Linear regression is a valuable tool in climate prediction, offering a straightforward approach to understanding the relationships between various climate variables. By analyzing historical climate data and relevant independent variables, linear regression models can provide insights into how changes in factors like greenhouse gas concentrations or solar radiation impact temperature, precipitation, and other climate patterns. While linear regression may not capture the full complexity of climate systems, it can still offer valuable insights and serve as a foundation for more advanced techniques. By effectively communicating the results and uncertainties of linear regression models, and policymakers can make more informed decisions to mitigate and adapt to climate change.

1.7 Language Used

Python

Python is a high-level, interpreted programming language known for its simplicity and readability. It was created by Guido van Rossum and first released in 1991. Python is widely used in various fields, including web development, data science, artificial intelligence, scientific computing, and more, due to its versatility and large standard library. Python emphasizes code readability and simplicity, making it an excellent choice for beginners and experienced programmers alike. It uses indentation to define code blocks, which helps to maintain clean and readable code. Python supports multiple programming paradigms, including procedural, object-oriented, and functional programming. It has a large and active community, which contributes to its extensive ecosystem of libraries and frameworks. Some of the key features of Python include readable and clean syntax, dynamic typing and automatic memory management (garbage collection), extensive standard library, compatibility with major operating systems, strong support for integration with other languages, high-level data structures, and built-in data types, and an easy-to-use syntax for writing complex programs. Overall, Python's simplicity, readability, and versatility make it a popular choice for a wide range of applications, from scripting and automation to web development and scientific computing.

HTML

HTML (Hypertext Markup Language) is the standard markup language used to create and design documents on the World Wide Web. It is the foundation of web development and provides the structure and content of web pages. HTML uses a system of tags to define the different parts of a web page, such as headings, paragraphs, links, images, and forms.

CSS

CSS (Cascading Style Sheets) is a style sheet language used to define the presentation and layout of HTML documents. CSS allows web developers to control the appearance of web pages, including aspects such as colors, fonts, spacing, and layout. By separating the content of a web page (defined in HTML) from its presentation (defined in CSS), developers can create more visually appealing and responsive websites.

1.7.1 System Configuration

H/W System Configuration

- RAM - 4 GB (min)
- SOFTWARE ID - python idle
- Mouse - Two or Three Button Mouse

S/W System Configuration

- Operating System - Windows 8/10
- Front End - CSS
- Language - Python (3.10) Version.
- Server - Flask.

1.8 Organization

Organizations at the forefront of climate prediction using deep learning are pioneering the application of advanced artificial intelligence to enhance our understanding of climate systems and improve the accuracy of climate forecasts. These organizations, such as the Climate Informatics Lab (CIL), Deep Climate AI, and Climate AI, focus on developing and applying deep learning models to analyze climate data and make predictions about future climate trends. By leveraging the power of deep learning, these organizations aim to address key challenges in climate science, such as predicting extreme weather events and assessing the impacts of climate change. Collaborations with research institutions like the National Center for Atmospheric Research (NCAR), European Centre for Medium-Range Weather Forecasts (ECMWF), and NASA's Goddard Institutes for Space Studies (GISS) further enhance the development and application of advanced AI techniques in climate science. Through their work, these organizations are advancing our ability to predict and mitigate the impacts of climate change, ultimately contributing to a more sustainable future.

CHAPTER 2

LITERATURE SURVEY

Title 1: Rain to Rain Learning Real Rain Removal Without Ground Truth **.**

Authors: Abderraouf Khodja,Zhonglong Zheng,Jiashuaizi Mo,Dawei Zhang,Liyuan Chen **.**

Year: 2021

Description:

Image draining is a low-level restoration task that has become quite popular during the past decades. Although recent data-driven draining models exhibit promising results, most of these models are trained on synthetic rain data sets which do not generalize well when applied to real rain images. While recent real-rain data sets have achieved favorable generalization performance, generating rain-free ground-truths can be tedious and time consuming. To address this problem, in this work, we present rain to rain training, an unsupervised training method for single image draining. Our experiments show that it is possible to train single image draining models by using only rain images. This can be achieved by simply training models to map pairs of rain images. We also introduce the idea of using the least overlapping training pairs, a method of selecting adequate training pairs that enables rain to rain training to achieve equivalent draining performance compared to supervised training. To address this problem, in this work, we present rain to rain training, an unsupervised training method for single image draining. Although recent data-driven draining models exhibit promising results, most of these models are trained on synthetic rain data sets which do not generalize well when applied to real rain images

Title 2: A Tensor Modeling for Video Rain Streaks Removal Approach Based on the Main Direction of Rain Streaks.

Author: Dou Yaping, Zhang Ping, Zhou Ying, Zhang Lingyi.

Year: 2020

Description:

The algorithms for video rain streaks removal do not properly consider the influence of wind on the main direction of rain streaks. They do not rotate or only perform a rough rotation when rain streaks deviate from the vertical direction, resulting in residual rain patterns or blurred background. Therefore, a sparse tensor model based on the main direction of rain streaks is suggested for video rain streaks removal in this paper. First, the first-order directional derivative (FODD) filter is used to obtain the rain image with the best background suppression effect. Second, we calculate its histogram of oriented gradient (HOG) feature to match the rain streaks image library. The main direction of rain streaks and the rotation angle of the global model are determined by the matching result

Title 3: Removing Rain Streaks using Long Short Term Memory by Linear Model.

Authors: Yinglong Wang, Shuaicheng Liu , Dehua Xie, Bing Zeng.

Year: 2020

Description:

Removing rain streaks from a single image continues to draw attentions today in outdoor vision systems. In this paper, we present an efficient method to remove rain streaks. First, the location map of rain pixels needs to be known as precisely as possible, to which we implement a relatively accurate detection of rain streaks by utilizing two characteristics of rain streaks. The key component of our method is to represent the intensity of each detected rain pixel using a linear model: $p = \alpha s + \beta$, where p is the observed intensity of a rain pixel and s represents the intensity of the background (i.e., before rain-affected). To solve α and β for each detected rain pixel, we concentrate on a window centered around it and form an L 2 -norm cost function by considering all detected rain pixels within the window, where the corresponding rain-removed intensity of each detected rain pixel is estimated by some neighboring non-rain pixels. By minimizing this cost function, we determine α and β so as to construct the final rain-removed pixel intensity. Compared with several state of-the-art works, our proposed method can remove rain streaks from a single color image much more efficiently - it offers not only a better visual quality but also a speed-up of several times to one degree of magnitude compared to other conventional draining methods.

Title 4: Rain Effects on CFOSAT Scatter meter: Towards an Improved Wind Quality Control.

Authors: Wending Lin, Marcos Portabella, Xiaokang Zhao, Shuyan Lang

Year: 2020

Description:

Rain is known to be the most significant phenomenon in degrading the Ku-band scatter meter wind quality. After the decommission of the National Aeronautics and Space Administration scatter meter (NSCAT), little work has been done in characterizing the impact of rain on Ku-band fan beam scatter meter. In this paper, the rain impact on the backscatter measurements as well as the retrieved wind quality of the China-France Oceanography Satellite (CFOSAT) scatter meter (CSCAT) is investigated using the European Centre for Medium-range Weather Forecasts (ECMWF) winds and the Global Precipitation Measurement (GPM) mission's Microwave Imager (GMI) rain data as reference.

The dependence of rain effects on the observing incidence angle is studied with the objective to optimize the configurations of wind inversion and quality control (QC).

Title 5: GAN-Based Rain Noise Removal from Single-Image Considering Rain Composite Models.

Author: Takuro Matsui, Masaaki Ikehara **.**

Year: 2020

Description:

Under severe weather conditions, outdoor images or videos captured by cameras can be affected by heavy rain and fog. For example, on a rainy day, autonomous vehicles have difficulty determining how to navigate due to the degraded visual quality of images. In this paper, we address a single-image rain removal problem (de-raining). As compared to video-based methods, single-image based methods are challenging because of the lack of temporal information. Although many existing methods have tackled these challenges, they suffer from over fitting, over-smoothing, and unnatural hue change. To solve these problems, we propose a GAN-based de-raining method. The optimal generator is determined by experimental comparisons. To train the generator, we learn the mapping between rainy and residual images from the training dataset. Besides we synthesize a variety of rainy images to train our network. In particular, we focus on not only the orientations and scales of rain streaks but also the rainy image composite models.

CHAPTER 3

SYSTEM DEVELOPMENT

System development typically refers to the process of creating or improving systems, which are sets of interconnected components working together to achieve a common goal. This process involves various stages, such as planning, designing, implementing, testing, and maintaining the system. It often includes activities like gathering requirements from stakeholders, designing the system architecture, developing the software, and deploying the system. System development can be applied to various types of systems, including software systems, information systems, and hardware systems, among others.

3.1 Functional Block Diagram of the proposed system

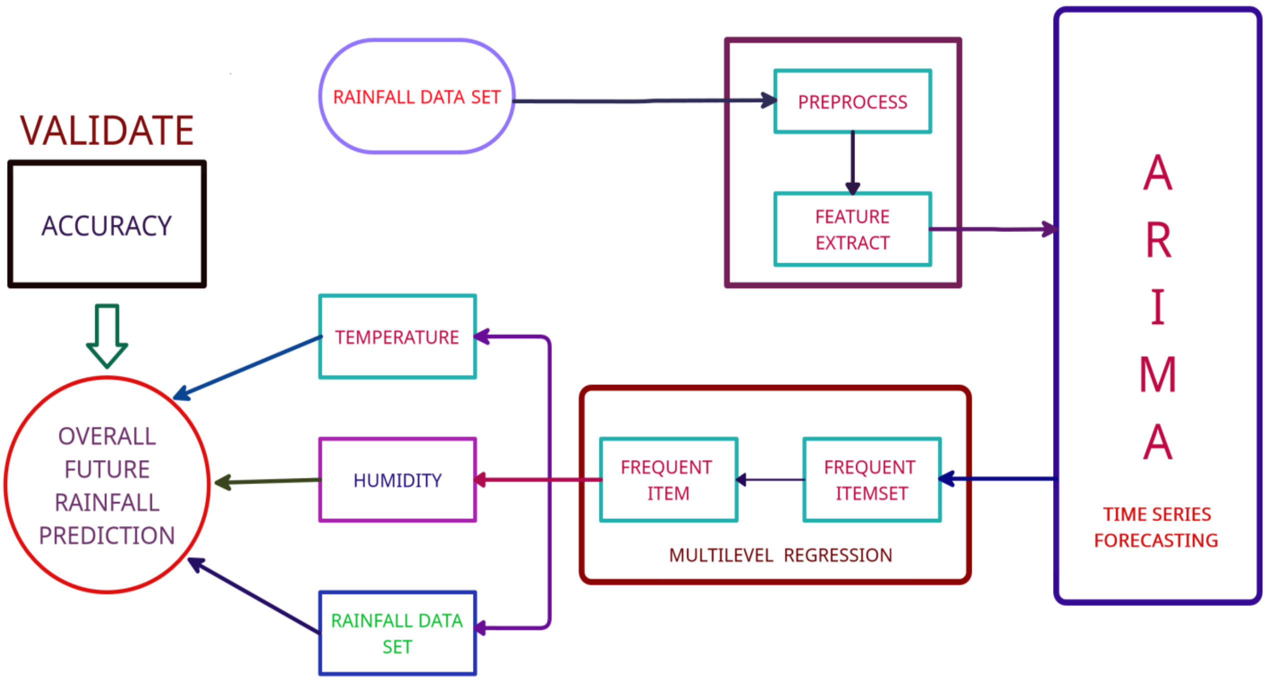

Figure 3.1 Block Diagram

Rainfall Data Set

In the context of predicting climate, a rainfall dataset would consist of historical or current data related to rainfall patterns. This dataset would typically include information such as the amount of rainfall recorded at different locations over time, along with additional variables that might influence rainfall patterns, such as temperature, humidity, wind speed, and geographical features. Researchers and climate scientists use rainfall datasets to analyze past trends, identify patterns, and develop models that can predict future rainfall patterns. These predictions can be crucial for various purposes, including agriculture, water resource management, disaster preparedness, and climate change research.

Data Preprocessing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. We have carried below preprocessing steps. Date feature can be expanded to Day, Month and Year and then these newly created features can be further used for other preprocessing steps. Feature hashing scheme is another useful feature engineering scheme for dealing with large scale categorical features. In this scheme, a hash function is typically used with the number of encoded features pre-set (as a vector of pre-defined length) such that the hashed values of the features are used as indices in this pre-defined vector and values are updated accordingly Our data set contains features with highly varying magnitudes and range. But since, most of the machine learning algorithms use Euclidean distance between two data points in their computations, this is a problem. The features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes. To suppress this effect, we need to bring all features to the same level of magnitudes. This can be achieved by scaling.

Feature Extract

Feature extraction refers to the process of selecting and transforming raw data (features) into a format that is suitable for use in climate prediction models. This process involves identifying relevant variables or parameters from various sources, such as atmospheric measurements, satellite data, oceanic observations, and historical climate records. Feature extraction aims to capture important information that can help improve the accuracy and effectiveness of climate prediction models. This may include extracting spatial and temporal patterns, identifying trends, and filtering out noise or irrelevant data. The extracted features are then used as input to machine learning algorithms or statistical models to make predictions about future climate conditions. feature extraction refers to the process of selecting and transforming raw data (features) into a format that is suitable for use in climate prediction models. This process involves identifying relevant variables or parameters from various sources, such as atmospheric measurements, satellite data, oceanic observations, and historical climate records.

ARIMA Time Series Forecasting

ARIMA models are employed to analyze historical climate data, such as temperature, precipitation, or atmospheric pressure, and to make predictions about future climate trends. The "Auto Regressive" component of ARIMA examines the relationship between an observation and a set of lagged observations, assuming that future values can be predicted based on past values. The "Integrated" component involves differencing the raw data to achieve stationarity, ensuring that the statistical properties of the time series remain constant over time. Finally, the "Moving Average" component models the relationship between an observation and residual errors from a moving average model applied to lagged observations.

Multilevel Regression

Multilevel regression refers to a statistical modeling approach that accounts for hierarchical or nested data structures. In climate prediction, this approach can be useful when the data exhibit a nested structure, such as climate measurements collected at multiple spatial or temporal levels. For example, in a study predicting regional climate patterns, you might have measurements collected at individual weather stations (lower level) nested within broader regions (higher level). Multilevel regression allows you to model the relationships between predictors (e.g., greenhouse gas concentrations, land use) and outcomes (e.g., temperature, precipitation) while accounting for the dependencies within and between these levels.

Overall Future Rainfall Prediction

Overall rainfall prediction is a complex process that involves using statistical or machine learning models to forecast the amount of rainfall over a specific period, such as daily, monthly, or seasonal. This prediction is based on various input parameters, including temperature, humidity, and potentially other factors. The process typically begins with collecting historical data on rainfall, temperature, humidity, and other relevant variables for the region of interest. After cleaning and preprocessing the data, features are selected or engineered based on domain knowledge. Machine learning models, such as linear regression, random forest, or gradient boosting, are then trained on this data to predict rainfall. The model's performance is evaluated using metrics like mean squared error or R-squared to assess its accuracy. However, predicting rainfall accurately can be challenging due to the dynamic nature of weather patterns, and incorporating local factors and topography is crucial for improving prediction accuracy.

3.2 Modules

- Data Exploration and Analysis.
- Data Preprocessing.
- Linear Regression.
- ARIMA (Auto Regressive Integrated Moving Average).
- Ensemble Methods and Bagging.

3.3 Module Description

3.3.1 Data Exploration and Analysis

Exploratory Data Analysis is valuable to machine learning problems since it allows getting closer to the certainty that the future results will be valid, correctly interpreted, and applicable to the desired business contexts. Such level of certainty can be achieved only after raw data is validated and checked for anomalies, ensuring that the data set was collected without errors. EDA also helps to find insights that were not evident or worth investigating to business stakeholders and researchers. We performed EDA using two methods - Univariate Visualization which provides summary statistics for each field in the raw data set and Pair-wise Correlation Matrix which is performed to understand interactions between different fields in the data set.

3.3.2 Data Preprocessing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. We have carried below preprocessing steps. Date feature can be expanded to Day, Month and Year and then these newly created features can be further used for other preprocessing steps. Feature hashing scheme is another useful feature engineering scheme for dealing with large scale categorical features. In this scheme, a hash function is typically used with the number of encoded features pre-set (as a vector of pre-defined length) such that the hashed values of the features are used as indices in this pre-defined vector and values are updated accordingly Our data set contains features with highly varying magnitudes and range. But since, most of the machine learning algorithms use Euclidean distance between two data points in their computations, this is a problem. The features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes. To suppress this effect, we need to bring all features to the same level of magnitudes. This can be achieved by scaling.

3.3.3 Linear Regression

Linear Regression is a method that describes the relationship between a dependent variable and a set of independent variables. The equation of the line is given. It provides an estimate of rainfall using various atmospheric variables like cloud cover, humidity, wind, and average temperature to predict rainfall. An estimate of rainfall is easy to determine at any given point since the regression method uses the previous correlation between the various atmospheric variables.

3.3.4 ARIMA Model (Auto Regressive Integrated Moving Average)

This model is used for time series prediction and analysis and forecasting. It contains four methods and is proposed by Box and Jenkins. The following are the four steps used in the ARIMA model.

Stage-1:

Identification of a series of responses is done in the first stage which is used in calculating the time series and autocorrelations using statement IDENTIFY

Stage-2:

In this stage Estimation of the previously identified variables is done and also the parameters are estimated using the statement ESTIMATE.

Stage-3:

Diagnostics checking of the above-collected variables and parameters is done in this stage. Stage-4: In this stage the predicting values of time series are forecasted which are future values, using the ARIMA model using the statement FORECAST.

The parameters used in this model are p,d,q which describes ‘p’ as the number of lag observations, ‘q’ as the degree of differencing and’ as the moving average order.

3.3.5 Ensemble methods and Bagging

An ensemble method constructs multiple prediction models for a given dataset and then combines these models into a final prediction model. The first ensemble algorithm to be proposed was bagging, based on the bootstrap method, followed later by boosting based on AdaBoost algorithm. There is also the random forest algorithm proposed ARIMA. Various other ensemble methods have been developed since then, but bagging, boosting, and ARIMA time series forecasting are the most popular ensemble methods that remain widely in use, and many studies have established that these methods can maximize the prediction performance of models.

Bagging, boosting, and ARIMA time series forecasting mainly use a single model repeatedly to aggregate the results. The model used is usually the decision tree model explained above. Since decision trees can be applied to classification as well as regression problems, ensemble methods that use decision trees can also be applied to both kinds of problems. While a single decision tree divides the space of explanatory variables into discrete partitions, ensembles using decision trees as base models average or vote over several differently partitioned decision trees. Thus, ensembles have the advantage of naturally learning nonlinear effects in addition to linear effects.

Bagging generates multiple bootstrap data from the original dataset, constructs a prediction model in a uniform way for each bootstrap data, and combines the models to arrive at the final model. Here, the term “bootstrap data” refers to a dataset obtained by random sampling with a replacement that has the same size as the original dataset.

3.4 Algorithm of the model

3.4.1 ARIMA model (Auto Regressive Integrated Moving Average)

This model is used for time series prediction and analysis and forecasting. It contains four methods and is proposed by Box and Jenkins. The following are the four steps used in the ARIMA model.

Stage-1:

Identification of a series of responses is done in the first stage which is used in calculating the time series and autocorrelations using statement IDENTIFY

Stage-2:

In this stage Estimation of the previously identified variables is done and also the parameters are estimated using the statement ESTIMATE.

Stage-3:

Diagnostics checking of the above-collected variables and parameters is done in this stage. Stage-4: In this stage the predicting values of time series are forecasted which are future values, using the ARIMA model using the statement FORECAST.

The parameters used in this model are p,d,q which describes ‘p’ as the number of lag observations, ‘q’ as the degree of differencing and’ as the moving average order.

3.4.1.1 Overview of the components

- Autoregression (AR): This refers to a model that uses the dependent relationship between an observation and a number of lagged observations (previous time steps).
- Integrated (I): This indicates that the raw observations are differenced to make the time series stationary. This might involve subtracting the

observation at the previous time step from the current observation.

- Moving Average (MA): This part of the model uses the dependency between an observation and a residual error from a moving average model applied to lagged observations
- The ARIMA model is specified by three order parameters: p, d, and q.
- p: The number of lag observations included in the model (autoregression order).
- d: The number of times that the raw observations are differenced (degree of differencing).
- q: The size of the moving average window (moving average order). ARIMA models are commonly used in time series analysis and forecasting for a wide range of applications, such as finance, and weather forecasting.

3.4.1.2 Steps of ARIMA Model

STEP 1: Plot tractor sales data as time series.

STEP 2: Difference data to make data stationary on mean (remove trend)

STEP 3: log transform data to make data stationary on variance.

STEP 4: Difference log transform data to make data stationary on both mean and variance.

STEP 5: Plot ACF and PACF to identify potential AR and MA model.

STEP 6: Identification of best fit ARIMA model

STEP 7: Plot ACF and PACF for residuals of ARIMA model to ensure no more information is left for extraction

3.4.1.3 Overview of how the ARIMA algorithm works

Stationarity Check

The algorithm begins by checking if the time series is stationary, meaning that its statistical properties (such as mean and variance) remain constant over time. If the series is not stationary, it is differenced until it becomes stationary.

Model Identification

The next step is to identify the appropriate values of p, d, and q for the ARIMA model. p (AR order): The number of lag observations included in the model. d (Differencing order): The number of times the differencing operation was applied to make the series stationary. q (MA order): The size of the moving average window.

Parameter Estimation

Once the model parameters (p, d, q) are determined, the next step is to estimate the coefficients of the model. This is typically done using the method of least squares.

Model Validation

The model is then validated using techniques such as residual analysis to ensure that it adequately captures the underlying patterns in the data.

Forecasting

Finally, the model can be used to forecast future values of the time series based on the estimated parameters. ARIMA models are widely used in various fields such as finance, economics, and meteorology for forecasting and analyzing time series data.

3.4.2 Linear Regression

In weather forecasting, linear regression can be used to predict weather-related variables (like temperature, precipitation, humidity, etc.) based on other relevant factors. Here's a simplified explanation of how linear regression can be applied in weather forecasting:

Data Collection

Collect historical weather data for the area of interest. This data should include the variables you want to predict (e.g., temperature) as well as other variables that might influence the prediction (e.g., humidity, wind speed, etc.).

Data Preprocessing

Clean the data by removing any outliers or missing values. Also, normalize the data if needed to ensure that all variables are on the same scale. Feature Selection: Choose the relevant features (independent variables) that will be used to predict the target variable (dependent variable). For example, if you're predicting temperature, features could include humidity, wind speed, and air pressure.

Split the Data

Split the data into training and testing sets. The training set will be used to train the linear regression model, while the testing set will be used to evaluate its performance.

Model Training

Use the training data to fit a linear regression model. The model will learn the relationship between the features and the target variable based on the training data.

Model Evaluation

Evaluate the performance of the model using the testing data. You can use metrics like mean squared error (MSE) or R-squared to assess how well the model is able to predict the target variable. Prediction: Once the model is trained and evaluated, you can use it to make predictions on new data. For example, you can use the model to predict tomorrow's temperature based on the current weather conditions.

Refinement

Refine the model as needed by adding more features, changing the model's parameters, or using a different algorithm altogether to improve its performance. Linear regression is just one of many techniques used in weather forecasting, and its effectiveness depends on the quality of the data and the choice of features. More advanced techniques, such as time series analysis or machine learning algorithms, are often used in conjunction with linear regression to improve forecast accuracy

3.4.2.1 Steps of Linear Regression

STEP 1: Load the data into python. Follow these four steps for each dataset: ...

STEP 2: Make sure your data meet the assumptions.

STEP 3: Perform the linear regression analysis.

STEP 4: Check for homoscedasticity.

STEP 5: Visualize the results with a graph.

STEP 6: Report your results.

3.4.2.2 Overview of how the Linear Regression algorithm works

Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables. In the context of climate prediction, linear regression can be used to predict climate variables (e.g., temperature, precipitation) based on historical data and other relevant factors (e.g., greenhouse gas concentrations, solar radiation). Here's an overview of how linear regression works in climate prediction:

Data Collection

Collect historical climate data, including the dependent variable (e.g., temperature) and independent variables (e.g., greenhouse gas concentrations, solar radiation), over a significant period.

Data Preprocessing

Clean the data to remove noise and outliers. Handle missing values by imputation or removal. Ensure the data is in a suitable format for linear regression analysis.

Feature Selection

Select relevant independent variables that are likely to have a significant impact on the dependent variable. This can be done based on domain knowledge or using techniques like correlation analysis.

Model Training

Split the data into training and test sets. Train the linear regression model using the training set.

Prediction

Once the model is trained and evaluated, use it to make predictions on new data. Monitor the model's performance over time and retrain it periodically to maintain its accuracy.

Uncertainty Estimation

Linear regression provides point estimates, but it's important to also estimate the uncertainty around these estimates. This can be done using techniques like bootstrapping or by calculating confidence intervals.

Post-processing

Post-process the model outputs if necessary to derive useful information for decision-making. This could involve aggregating predictions over regions or time periods, or translating them into actionable insights.

Communication

Communicate the linear regression model results and uncertainties effectively to stakeholders, policymakers, and the public to support informed decision-making.

CHAPTER 4

PERFORMANCE ANALYSIS

4.1 Data Set

The dataset being used for our prediction models comprises of weather records of the city in focus collected over a period of time using various different parameters like temperature, humidity, atmospheric pressure, and so on. Till date it consists of a record of weather over a period of 20 years (2004 - 2024).

The data enclosed in our dataset is classified into the following categories: -

i) Temperature
ii) Pressure
iii) Humidity

Temperature is a measure of the degree of hotness or coldness of the surroundings. It, like all weather conditions, varies from instance to instance. Similarly, atmospheric pressure and humidity, that plays a vital role in predicting whether an area will receive precipitation or not, is also included in the dataset. Details about fog and dew point are included in the dataset as well, as they only contribute to improving the accuracy of the predictions made by the prediction models.

All the data gathered in the dataset was collected from that has an easy to use in ARIMA networks, which makes data collection all the more simpler.

4.2 Accuracy

Accuracy is one of the measures to evaluate classification models. Accuracy refers to how close a measurement is to its true or accepted value. Accurate measurements are essential for understanding the external world. When an instrument provides accurate values, it means that the measured quantity closely aligns with the standard or true value. This assesses an instrument's accuracy at a specific point on its scale. However, it doesn't provide information about the overall accuracy of the instrument. The uniform scale range determines accuracy. For instance, a thermometer with a scale range up to 500ºC and an accuracy of ±0.5% of the scale range will have a maximum error of ±2.5ºC. This type of accuracy compares measured values to their true values, allowing us to assess accuracy within ±0.5% of the true value.

The Accuracy is the fraction of the predictions given by the classification model. The precision has the following definition:

$$\text{Accuracy} = \text{Total no. of the correct forecasts} / \text{From predictions}$$

For the binary classification, the accuracy can be calculated as negative and positive in the following way:

$$\text{Accuracy } \% = ((TP+TN)/(TP+TN+FP+FN)) \times 100$$

TP = True Positives,

TN = True Negatives,

FP = False Positives, and

FN = False Negatives

4.3 Test Setup

The test process is already in-built in our system. The testing process taking place just after the model is trained. After the completion of the training process, we analyze each data entry in the test set. In order to analyze each entry, we use descriptors to extract features. Now we compare these feature values with the feature values which were initially retained using the train set. The comparison is done according to the Machine Learning model used and finally the output for each entry is received. Since each data entry is already labeled, we can compute accuracy by comparing the predicted value with the received values.

4.4 Recall

Recall it helps to determine how much the false negatives

$$Recall = TP / (TP + FN)$$

Where TP is true positive

FN is false negative number.

Recall refers to the classification capability to find all the classified samples.

Table 4.1 Real value and Predicted Values

	Predicted Value	**Predicted Value**
Real Value	<u>True Positive(TP)</u> Reality: Rain ML model predicted: Rain	<u>False Positive(FP)</u> Reality: No Rain ML model predicted: Rain
Real Value	<u>False Negative(FN)</u> Reality: Rain ML model predicted: No Rain	<u>True Negative(TN)</u> Reality: Benign ML model predicted: No Rain

4.5 Precision

The precision determines how often it is correct when the model predicts positive. Precision refers to the closeness of multiple measurements of the same quantity to each other. Precision and accuracy are crucial concepts in scientific measurement. Striving for high precision minimizes errors in measurements and calculations. Precise measurements yield consistent results. Variation using the same measurement process across different instruments and operators over longer time periods. In summary, accuracy reflects closeness to the true value, while precision focuses on consistency among repeated measurements. Both factors contribute to reliable scientific results. Accuracy helps determine when the cost of false positives is high.

$$\text{Precision} = TP / (TP + FP)$$

where TP is the number of real positives

FP the number of false positives.

Precision refers to the ability of the classifier not to designate a positive sample as negative.

4.6 Experiment Result

The results of the implementation of the project are demonstrated below.

Multilevel Regression

Multilevel regression is a statistical technique used to analyze data sets that are hierarchical in nature, often consisting of subjects nested within groups. The model assumes that there is one single outcome or response variable that is measured at the lowest level, and explanatory variables at all existing levels.

The Mean Square Error are detected by the following formula

mse = ((y_forecasted - y_truth)

The Confidence Factor for Predicted Values are calculated by

pred_ci = pred_uc.conf_int()

Multilevel modeling is a generalization of linear and generalized linear modeling in which regression coefficients are themselves given a model, whose parameters are also estimated from data. This regression model has high variance, hence turned out to be the least accurate model. Given below is a snapshot of the actual result from the project implementation of multilevel regression**.**

Table 4.2 Actual Values Vs Prediction Values

S.No	Actual Value	Predicted Value
1.	0	0.0459157
2.	0	0.0423579
3.	0	0.0474239
4.	1	0.8654278
5.	0	0.0325468
6.	0	0.0023542
7.	0	0.1236582

Table 4.3 Differentiating Factors

Differentiating Factors	Existing Models (in %)	Proposed Models (in %)
Accuracy %	87.5	90
Precision %	89.5	90
Mean Square Error %	91	96
Confidence Factor %	91	96

The above factors mentioned are the differentiating factors which influence the overall functionality of an algorithm based for the weather prediction methodology. It is noted that the Existing models like LSTM (Long Short Term Memory) Algorithm which was previously used for the development of Weather Prediction model has provided with 87.5 % accuracy whereas our Proposed model which is ARIMA Networks algorithm has provided with 90 % accuracy.

The Existing models like LSTM has around 89.5 % precision which provides faster results whereas our proposed model ARIMA model has 90% precision, which is better when compared to the previous models. The Existing models like LSTM has around 91% Mean Square Error which provides faster results whereas our proposed model ARIMA model has 90% Mean Square Error, which is better when compared to the previous models.

The Existing models like LSTM has around 91% Confidence Factor which provides faster results whereas our proposed model ARIMA model has 96% Confidence Factor, which makes the results better. With the provided factors and analysis made we can surely determine the best model.

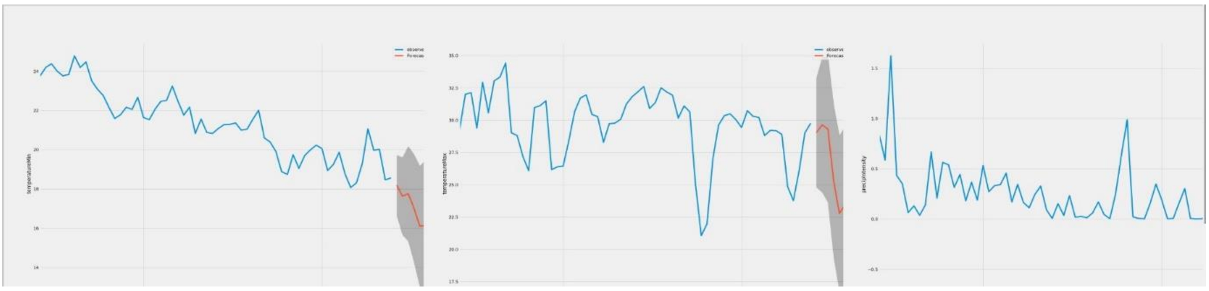

Figure 4.1 Predicted Weather Timeline in the proposed system

The Following is the weather predicted in the

Location : ASSAM

Date : 17/02/2024

Time : 19:35:25

Minimum Temperature in C : 18.18627577

Maximum Temperature in C : 29.03948401

Precipitation Intensity : -0.03301360677

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusion

This research paper introduces an innovative methodology that employs machine learning techniques to analyse and predict regional wind patterns in the context of global warming data. Researchers, particularly in scenarios involving remote sensing observations, often face challenges in obtaining complete datasets, leading to a specific focus on thermodynamic parameters dictated by the constraints of satellite-mounted instruments. The adoption of a machine learning approach represents a significant advancement in establishing a correlation between thermodynamic and kinetic parameters, offering a unique ability to forecast kinetic elements solely based on thermodynamic data, a capability beyond traditional numerical models.

The study commences by utilizing the widely recognized numerical model WRF to simulate and forecast intricate wind behaviours in the Beijing-Tianjin-Hebei region, generating comprehensive datasets comprising essential variables such as temperature, humidity, pressure, wind speed, and directional indicators. Subsequent stages involve the meticulous development of various machine learning models to effectively capture the intricate relationship between meteorological metrics and wind dynamics. To overcome the challenge of limited sample size hindering optimal training of machine learning models, the concept of transfer learning is introduced, addressing this issue effectively. The trained machine learning model is applied to a dataset containing 1700 samples to accurately predict the wind patterns. As a prospective strategy to improve the study's efficacy, expanding the dataset to incorporate a larger sample size is recommended, as this adjustment is poised to unlock heightened predictive capabilities. Looking towards the future, the integration of extensive data in subsequent research endeavours is anticipated to bring about transformative improvements in the domain of numerical prediction methodologies. Reliable weather predictions help us plan our daily activities. Whether it’s deciding to carry an umbrella or scheduling, accurate forecasts enhance convenience. Farmers rely on weather forecasts for planting, irrigation, and pest control. Timely information improves crop yield and reduces losses. Airlines, shipping companies, and road transport services use weather data to optimize routes and ensure passenger safety. Climate prediction models provide early warnings for natural disasters such as hurricanes, cyclones, and floods. This allows communities to evacuate and take preventive measures.

By predicting extreme weather events, we can minimize damage to property, infrastructure, and human lives. Energy Efficiency and Sustainability: Climate predictions guide the placement of solar panels, wind turbines, and hydropower plants. This ensures efficient energy production and reduces reliance on fossil fuels. Architects and engineers use climate data to design energy-efficient buildings. Proper insulation, ventilation, and orientation contribute to sustainability. Predictions help public health authorities prepare for heat waves. Vulnerable populations receive timely advice on staying hydrated and avoiding heat-related illnesses Predicting rainfall patterns assists water management. It helps allocate water for agriculture, industry, and domestic use. Climate models guide conservation efforts by identifying areas at risk due to changing temperatures and habitats. Accurate predictions inform international climate agreements (e.g., the Paris Agreement). Countries collaborate to mitigate climate change. Policymakers use climate data to develop strategies for adapting to changing conditions.

5.2 Future Work

The proposed work presents the development of a comprehensive rainfall recommendation system that leverages the combined power of ARIMA and K-means clustering algorithms to yield highly efficient results in terms of computational speed and accuracy. This innovative model is designed to analyze a diverse spectrum of rainfall patterns and corresponding agricultural yields within specific geographical areas, taking into consideration various weather and seasonal parameters extracted from time series datasets unique to each region.
Inspecting the visualization outputs generated by K-Means clustering provides valuable insights into the average produce levels associated with different rainfall groups, enabling informed decision-making in agricultural planning.

The integration of ARIMA and recommender functions focuses on seasonal rainfall patterns, allowing for a deeper exploration of the intricate relationships between crucial environmental factors such as optimal temperature, wind speed, humidity, soil conditions, and seed requirements within a given region. Furthermore, the scalability of this system facilitates its potential application for providing tailored rainfall recommendations to other states by following a similar methodological framework..

Moreover, the system's capabilities could be further augmented by incorporating factors like irrigation techniques into its analytical framework, thereby refining its predictive capabilities and enhancing its utility for farmers and agricultural stakeholders. By expanding the functionality to include early warnings about potential anomalies in rainfall patterns during specific seasons, farmers can be proactively advised on the types of fertilizers or soil nutrients needed to optimize crop growth and ensure high levels of accuracy in ARIMA-based predictions.

ANNEXURE

6.1 Sample code

arima.py:

```
import warnings
import itertools
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')

import requests
import json
import sys
import os
import pyodbc
import datetime

import pandas as pd

import collections
import time
from datetime import datetime, timedelta
```

```
import os

df = pd.read_csv("weather.csv").set_index("time")

city = str(input("Enter city: "))
#city = "Karnataka"
df = df.loc[df['city'] == city]

df.index = pd.to_datetime(df.index)
#df.sort_values("time")
predictfeature = str(input("Enter feature to predict: "))
#predictfeature = "temperatureMax"
feature = [predictfeature]

data = df[feature]
print(data)
y = data

# The 'MS' string groups the data in buckets by start of the month

if predictfeature == 'precipIntensity':
    y = y.precipIntensity.resample('d').mean()
elif predictfeature == 'precipIntensityMax':
    y = y.precipIntensityMax.resample('d').mean()
elif predictfeature == 'precipProbability':
    y = y.precipProbability.resample('d').mean()
elif predictfeature == 'temperatureMin':
```

```
    y = y.temperatureMin.resample('d').mean()
elif predictfeature == 'temperatureMax':
    y = y.temperatureMax.resample('d').mean()
else:
    print("Invalid feature")
    exit(1)

# The term bfill means that we use the value before filling in missing values
y = y.fillna(y.bfill())
print(y)

temp = y.head(1)
print(temp)
temp = np.array(temp.index)
print("temp = " , temp)

lastdate = ''
for i in  temp:
    t = str(i).split("T")
    t = t[0]
    t = t.split("-")
    t= datetime(int(t[0]),int(t[1]),int(t[2]))
    lastdate = t

y.plot(figsize=(15, 6))
plt.show()
```

```
p = d = q = range(0, 2)

# Generate all different combinations of p, q and q triplets
pdq = list(itertools.product(p, d, q))

# Generate all different combinations of seasonal p, q and q triplets
seasonal_pdq = [(x[0], x[1], x[2], 12) for x in list(itertools.product(p, d,
q))]

#print('Examples of parameter combinations for Seasonal ARIMA...')
#print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[1]))
#print('SARIMAX: {} x {}'.format(pdq[1], seasonal_pdq[2]))
#print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[3]))
#print('SARIMAX: {} x {}'.format(pdq[2], seasonal_pdq[4]))

warnings.filterwarnings("ignore") # specify to ignore warning messages

for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(y,
                                            order=param,
                                            seasonal_order=param_seasonal,
                                            enforce_stationarity=False,
                                            enforce_invertibility=False)


            results = mod.fit()
```

```
        #print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal,
results.aic))
      except:
        continue



mod = sm.tsa.statespace.SARIMAX(y,
                                order=(1, 1, 1),
                                seasonal_order=(1, 1, 0, 12),
                                enforce_stationarity=True,
                                enforce_invertibility=False)

results = mod.fit()

pred = results.get_prediction(start=pd.to_datetime(lastdate),
dynamic=False)
pred_ci = pred.conf_int()
print(pred.predicted_mean)



y_forecasted = pred.predicted_mean
y_truth = y[lastdate:]

# Compute the mean square error
mse = ((y_forecasted - y_truth) ** 2).mean()
print('The Mean Squared Error of our dynamic forecasts is
```

```
{}'.format(round(mse, 2)))

pred_dynamic = results.get_prediction(start=pd.to_datetime(lastdate), dynamic=True, full_results=True)
pred_dynamic_ci = pred_dynamic.conf_int()

#ax = y['2017':].plot(label='observed', figsize=(20, 15))
#pred_dynamic.predicted_mean.plot(label='Dynamic Forecast', ax=ax)
print(pred_dynamic.predicted_mean)
# Extract the predicted and true values of our time series
y_forecasted = pred_dynamic.predicted_mean
y_truth = y[lastdate:]

# Compute the mean square error
mse = ((y_forecasted - y_truth) ** 2).mean()
print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 2)))

# Get forecast 500 steps ahead in future
pred_uc = results.get_forecast(steps=7)
print(pred_uc.predicted_mean)

# Get confidence intervals of forecasts
pred_ci = pred_uc.conf_int()

ax = y.plot(label='observed', figsize=(20, 15))
pred_uc.predicted_mean.plot(ax=ax, label='Forecast')
```

```
ax.fill_between(pred_ci.index,
                pred_ci.iloc[:, 0],
                pred_ci.iloc[:, 1], color='k', alpha=.25)
ax.set_xlabel('Date')
ax.set_ylabel(predictfeature)

plt.legend()
plt.show()
```

regression.py :

```
#import matplotlib
#import matplotlib.pyplot as plt
import numpy as np


import pandas as pd
import statsmodels.api as sm
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, median_absolute_error
from sklearn.model_selection import train_test_split
import requests, json
import collections
import time
import datetime
import os

def get_target_date():
    """Return target date 1000 days prior to current date."""
    current_date = datetime.now()
    target_date = current_date - timedelta(days=1000)
    return target_date

def derive_nth_day_feature(df, feature, N):
    nth_prior_measurements = df[feature].shift(periods=N)
```

```
    col_name = f'{feature}_{N}'
    df[col_name] = nth_prior_measurements

features = [
        'time', 'precipIntensity', 'precipIntensityMax',
        'precipProbability',
       'temperatureMin',  'temperatureMax',
        'apparentTemperatureMin',
        'apparentTemperatureMax',
]

df = pd.read_csv("weather.csv").set_index("time")

city = str(input("Enter city: "))

df = df.loc[df['city'] == city]
df.dropna()

print(df.sort_values("time"))
print(df.columns)

nextday = datetime.datetime.today()
nextday += datetime.timedelta(days=1)
temp = str(nextday).split(" ")[0]
temp = (temp).split("-")
temp = datetime.datetime(int(temp[0]),int(temp[1]),int(temp[2]))

nextday = temp
print(nextday)

record = [[nextday,",",",",",",","]]
newdf = pd.DataFrame(record, columns=features).set_index('time')
print(newdf)

df.index = pd.to_datetime(df.index)
newdf.index = pd.to_datetime(newdf.index)


features = [
        'precipIntensity', 'precipIntensityMax',
        'precipProbability',
```

```
      'temperatureMin',  'temperatureMax',
       'apparentTemperatureMin',
       'apparentTemperatureMax',
]

data = df[features]

data = data.sort_values(by=['time'])
data = data.resample('d').mean().dropna(how='all')
#print("Edited database with no dublicates \n", data)
data = data.append(newdf)

df = data

# target measurement of mean temperature
predictfeature = str(input("Enter feature to predict: "))
ft = [predictfeature]

#print(tmp[feature][1])
# a list representing Nth prior measurements of feature
# notice that the front of the list needs to be padded with N
# None values to maintain the constistent rows length for each N

for feature in features:
   if feature != 'time':
         for N in range(1, 4):
           derive_nth_day_feature(df, feature, N)


print("Dataframe with nth day features: " , df)

to_remove = [
   feature for feature in features
   if feature not in ft
]
#print(to_remove)

# make a list of columns to keep
to_keep = [col for col in df.columns if col not in to_remove]
#print(to_keep)
```

```
# select only the columns in to_keep and assign to df
df = df[to_keep]

df = df.apply(pd.to_numeric, errors='coerce')

#print(df.info())
# Call describe on df and transpose it due to the large number of columns
spread = df.describe().T

# precalculate interquartile range for ease of use in next calculation
IQR = spread['75%'] - spread['25%']

# create an outliers column which is either 3 IQRs below the first quartile or

spread['outliers'] = (spread['min'] <(spread['25%'] -(3 * IQR))) | (spread['max'] > (spread['75%'] + 3 * IQR))
#print(spread)
#print(spread.iloc[spread.outliers,])

#print("Current: ",df)

trial = df.loc[nextday]
#print("Testing dataset: " , trial)

df = df.dropna()
#print(df)

df_corr = df.corr()[[predictfeature]].sort_values(predictfeature)
#print(df_corr)
df_corr_fil = df_corr[abs(df_corr[predictfeature]) > 0.30]
#print(df_corr_fil)

unwanted = [predictfeature]
predictors = df_corr_fil.index.tolist()
predictors = [i for i in predictors if i not in unwanted]
print("Predictors: ", predictors)

df2 = df[[predictfeature] + predictors]
trial = trial[[predictfeature] + predictors]
```

```
X = df2[predictors]
trial = trial[predictors]

y = df2[predictfeature]
alpha = 0.05

# Add a constant to the predictor variable set to represent the Bo intercept
X = sm.add_constant(X)
#print("Testing dataset: ", trial)
#print("X dataset: ", X)

def stepwise_selection(X,
                       y,
                       initial_list=predictors,
                       threshold_out=alpha,
                       verbose=True):

    included = list(initial_list)
    #print("Initial list : ", initial_list)


    while True:
        #print("List:", included)
        changed = False
        model = sm.OLS(y,X[included]).fit()
        # use all coefs except intercept
        pvalues = model.pvalues.iloc[1:]
        #print("Values: ", pvalues)
        worst_pval = pvalues.max()  # null if pvalues is empty
        if worst_pval > threshold_out:
            changed = True
            worst_feature = pvalues.idxmax()
            #print("Worst Feature:", worst_feature)
            included.remove(worst_feature)
            #print("List:", included)
            if verbose:
                print('Drop {:30} with p-value {:.6}'.format(worst_feature,
worst_pval))
        if not changed:
            break
    return included
```

```
result = stepwise_selection(X, y)

print('Resulting features:')
print(result)

X = X[result]
trial=trial[result]
#print("X: ", X)
#print("Testing: ", trial)

model = sm.OLS(y, X).fit()
print(model.summary())

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=12)

regressor = LinearRegression()

regressor.fit(X_train, y_train)
prediction = regressor.predict(X_test)
#print("X_test : " , X_test)
#print("Prediction: ", prediction)

trial = [trial]
print(trial)
predicttest = regressor.predict(trial)
print("Prediction of testing: ", predicttest)
print('The Explained Variance: %.2f' % regressor.score(X_test, y_test))
print('The Mean Absolute Error: %.2f degrees celcius' % mean_absolute_error(
    y_test, prediction))
print('The Median Absolute Error: %.2f degrees celcius' %
      median_absolute_error(y_test, prediction))
```

APPENDIX

SCREENSHOTS

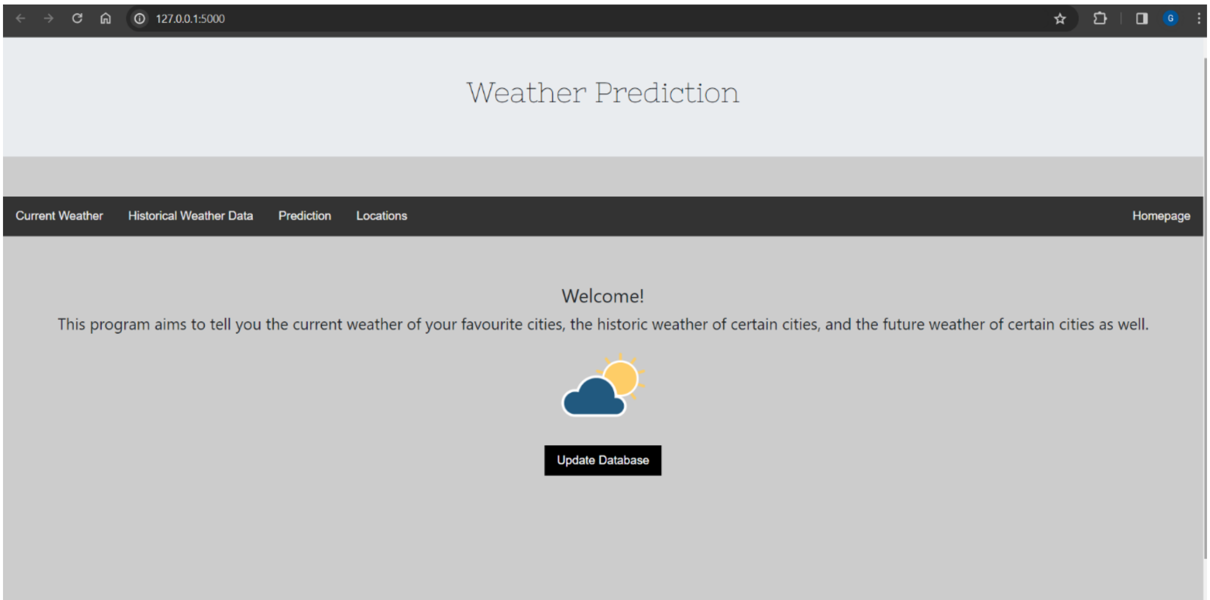

Figure 7.1 : HOME PAGE

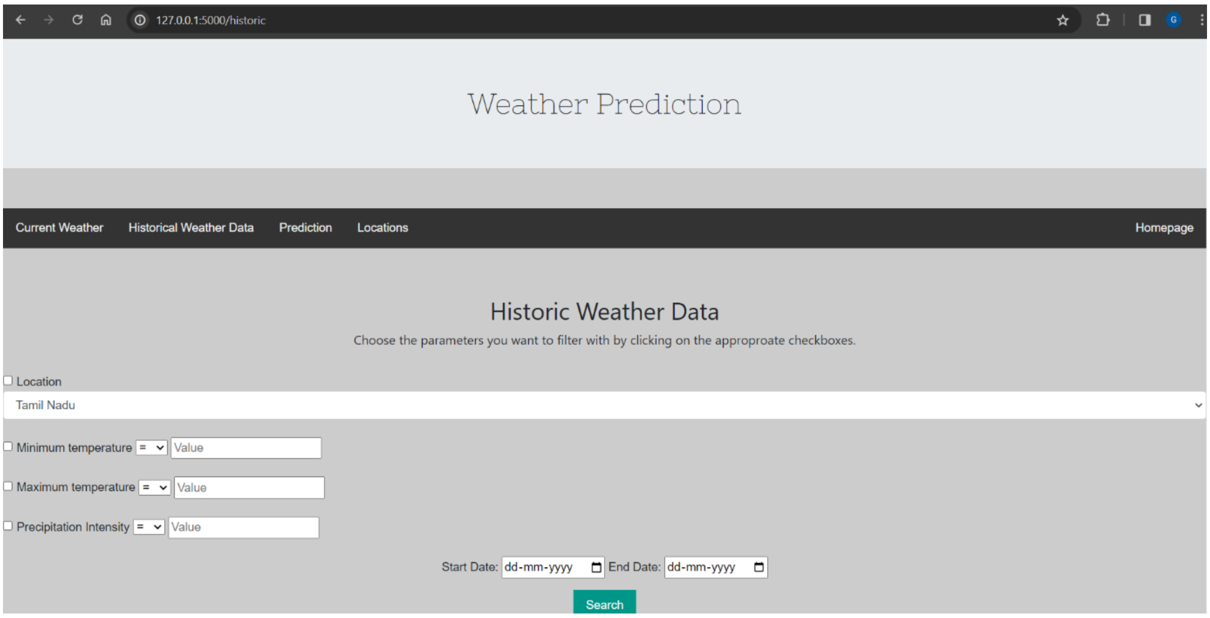

Figure 7.2: HISTORICAL WEATHER DATA

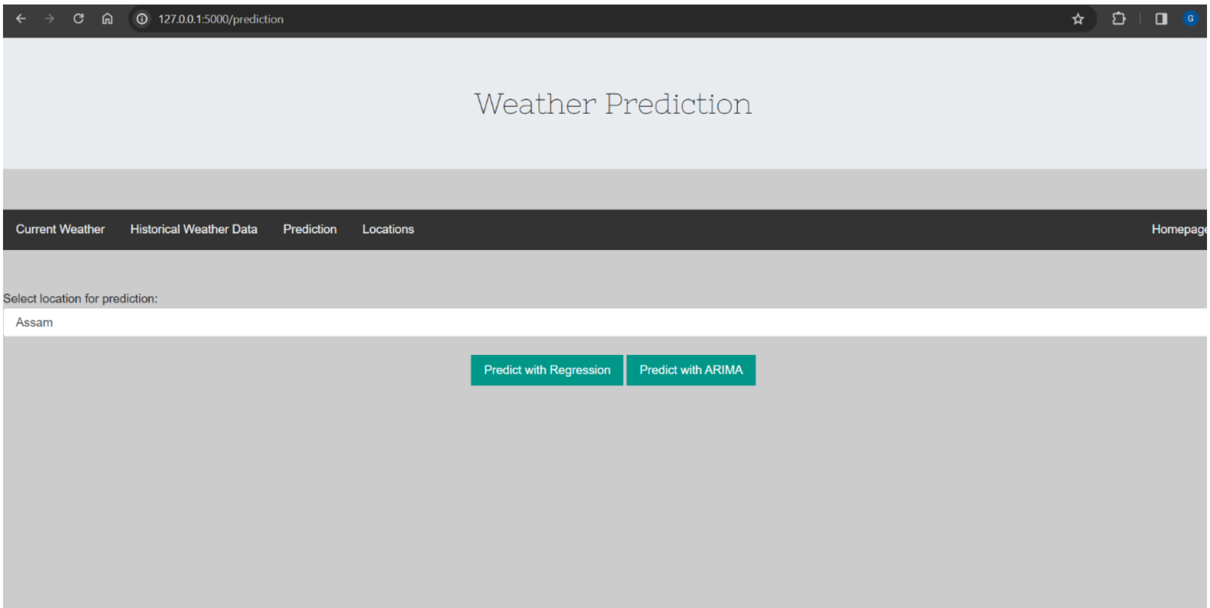

Figure 7.3 TYPES OF PREDICTION

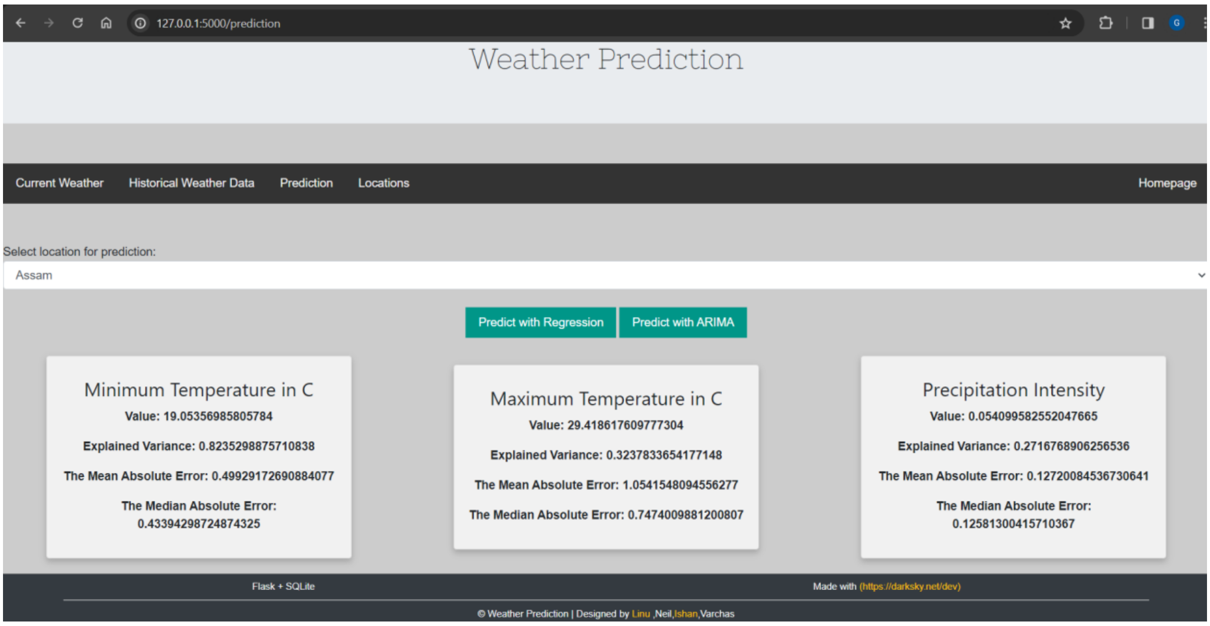

Figure 7.4 PREDICTION WITH REGRESSION

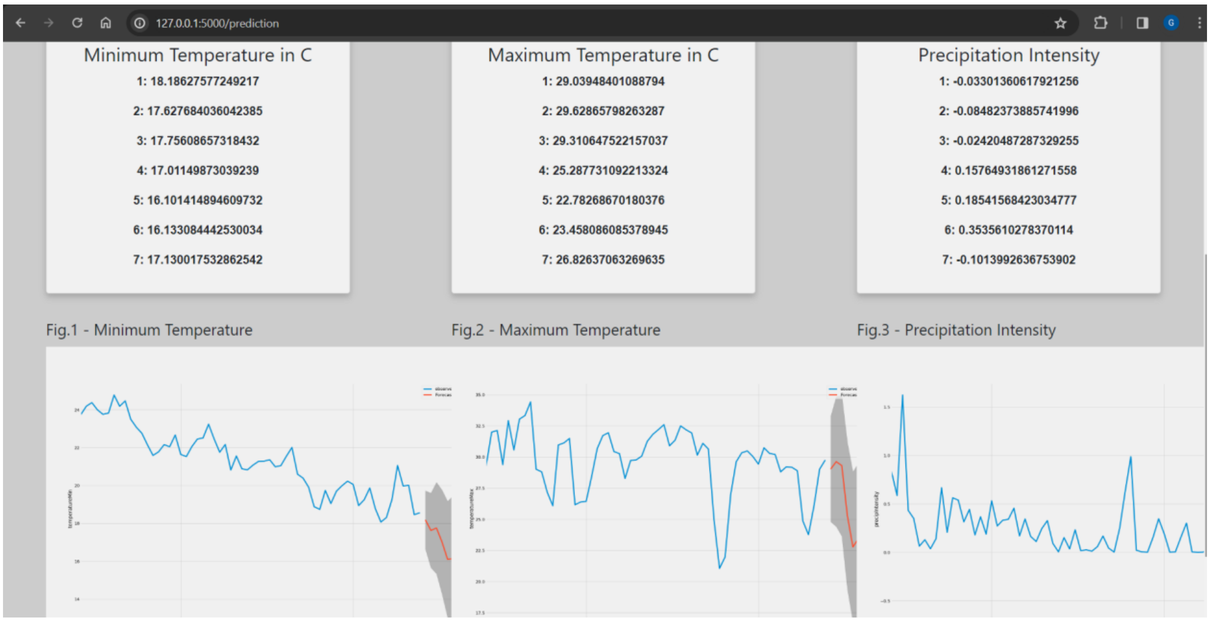

Figure 7.5: PREDICTION WITH ARIMA

REFERENCES

[1] Araujo,Dembele,Christensen,Saka,''Short-term wind speed prediction using hybrid machine learning techniques,'' Environ. Sci. Pollut. Res., Jul. 2021, doi: 10.1007/s11356-021-15221-6.

[2] Alisha.K,''Multi-step influenza outbreak forecasting using deep LSTM network and genetic algorithm,'' Expert Syst. Appl., vol. 180, Oct. 2021, Art. no. 115153, doi: 10.1016/j.eswa.2021.115153.

[3] Benjamin Häusler, K. Choudhary,Y. Matsuda, M. Pätzold, , N. Sugimoto,H. Ando,, M. Takagi, T. Imamura,H. Sagawa, S. Limaye, S. Tellmann,,, and M. Antonita, ''Thermal structure of the Venusian atmosphere from the sub-cloud region to the mesosphere as observed by radio occultation,'' Sci. Rep., vol. 10, no. 1, pp. 1–7, Dec. 2020, doi: 10.1038/s41598-020-59278-8.

[4] Chattopadhyay A, Nabizadeh E, Hassanzadeh P (2020) Analog forecasting of extreme-causing weather patterns using deep learning. J Adv Model Earth Syst 12(2):e2019MS001958. https://doi.org/10.1029/2019MS001958.

[5] ChoDet al (2022) A novel ensemble learning for post-processing of NWP model's next-day maximum air temperature forecast in summer using deep learning and statistical approaches. Weather Climate Extremes 35:100410

[6] Chen Yang, L. Zhu, and J. Min, ''Impact study of FY-3B MWRI data assimilation in WRFDA,'' Atmosphere, vol. 12, no. 4, p. 497, Apr. 2021, doi: 10.3390/atmos12040497.

[7] Dhandapani Gupta, N. Natarajan, and M. Berlin, ''Short-term wind speed prediction using hybrid machine learning techniques,'' Environ. Sci. Pollut. Res., Jul. 2021, doi: 10.1007/s11356-021-15221-6.

[8] Fister D, et al (2023) Accurate long-term air temperature prediction with Machine Learning models and data reduction techniques. Appl Soft Comput 110118.

[9] Grönquist P et al (2021) Deep learning for post-processing ensemble weather forecsts.PhilosTransRSoA379(2194):20200092.https://doi.org/10.1098/rsta.

[10] Gokul.H, N. Sugimoto,H. Ando,, M. Takagi, T. Imamura,H. Sagawa, S. Limaye, S. Tellmann,, R. K. Choudhary, and M. Antonita, ‘‘Thermal structure of the Venusian atmosphere from the sub-cloud region to the mesosphere as observed by radio occultation,’’ Sci. Rep., vol. 10, no. 1, pp. 1–7, Dec. 2020, doi: 10.1038/s41598-020-59278-8.

[11] Hans Zimmer.A, T. Imamura, S. Tellmann, M. Pätzold, B. Häusler, N. Sugimoto, M. Takagi, H. Sagawa, S. Limaye, Y. Matsuda, R. K. Choudhary, and M. Antonita, ‘‘Thermal structure of the Venusian atmosphere from the sub-cloud region to the mesosphere as observed by radio occultation,’’ Sci. Rep., vol. 10, no. 1, pp. 1–7, Dec. 2020, doi: 10.1038/s41598-020-59278-8.

[12] Hewage P et al (2021) Deep learning-based effective fine-grained weather forecasting model. Pattern Anal Appl 24(1):343–366. https://doi.org/10.1007/s10044-020-00898-1

[13] Hyosung, S. Du, R. Wang, J. Wang, Y. Wang, C. Wei, and Q. Liu, ‘‘Potential for vertical heterogeneity prediction in reservoir basing on machine learning methods,’’ Geofluids, vol. 2020, pp. 1–12, Aug. 2020, doi: 10.1155/2020/3713525.

[14] Isak Joshua. M. Nezhad, E. Abbasnejad, S. Mirjalili, L. B. Tjernberg D. A. Garcia, B. Alexander, and M. Wagner, ‘‘A deep learning-based evolutionary model for short-term wind speed forecasting: A case study of the lillgrund offshore wind farm,’’ Energy Convers. Manage., vol. 236, May 2021, Art. no. 114002, doi: 10.1016/j.enconman.2021.114002

[15] JamalD. Domingo, J. Gomez-Garcia-Bermejo, and E. Zalama, ‘‘Optimization and improvement of a robotics gaze control system using LSTM networks,’’ Multimedia Tools Appl., vol. 81, no. 3, pp. 3351–3368, 2021, doi: 10.1007/s11042-021-11112-7.

[16] Jaffer.Lv, X. Zheng, M. Pawlak, W. Mo, and M. Miśkowicz, ‘‘Very short-term probabilistic wind power prediction using sparse machine learning and nonparametric density estimation algorithms,’’ Renew. Energy, vol. 177, pp. 181–192, Nov. 2021, doi: 10.1016/j.renene.2021.05.123

[17] Kandha Shivam, J.-C. Tzou, and S.-C. Wu, ‘‘Multi-step short-term wind speed prediction using a residual dilated causal convolutional network with nonlinear attention,’’ Energies, vol. 13, no. 7, p. 1772, Apr. 2020,

[18] Lita.C, et al (2020) Air temperature forecasting using traditional and deep learning algorithms. In: 2020 7th International conference on information science and control engineering (ICISCE), 18–20 December , Changsha, China, pp 189–194

[19] Mahmood.M, M. Pätzold, B. Häusler, N. Sugimoto,H. Ando,, M. Takagi, T. Imamura,H. Sagawa, S. Limaye, S. Tellmann,, R. K. Choudhary, and M. Antonita, ''Thermal structure of the Venusian atmosphere from the sub-cloud region to the mesosphere as observed by radio occultation,'' Sci. Rep., vol. 10, no. 1, pp. 1–7, Dec. 2020, doi: 10.1038/s41598-020-59278-8

[20] Mohamed Neshat, M. M. Nezhad, E. Abbasnejad, S. Mirjalili, L. B. Tjernberg D. A. Garcia, B. Alexander, and M. Wagner, ''A deep learning-based evolutionary model for short-term wind speed forecasting: A case study of the lillgrund offshore wind farm,'' Energy Convers. Manage., vol. 236, May 2021, Art. no. 114002, doi: 10.1016/j.enconman.2021.114002.

[21] Mario Pätzold, B. Häusler, N. Sugimoto,H. Ando,, M. Takagi, T. Imamura,H. Sagawa, S. Limaye, S. Tellmann,Y. Matsuda, R. K. Choudhary, and M. Antonita, ''Thermal structure of the Venusian atmosphere from the sub-cloud region to the mesosphere as observed by radio occultation,'' Sci. Rep., vol. 10, no. 1, pp. 1–7, Dec. 2020, doi: 10.1038/s41598-020-59278-8.

[22] Nakamura.S, Y. Matsuda, M. Pätzold, B. Häusler,,H. Ando,, M. Takagi, T. Imamura,H. Sagawa, S. Limaye, S. Tellmann,, R. K. Choudhary, and M. Antonita, ''Thermal structure of the Venusian atmosphere from the sub-cloud region to the mesosphere as observed by radio occultation,'' Sci. Rep., vol. 10, no. 1, pp. 1–7, Dec. 2020, doi: 10.1038/s41598-020-59278-8.

[23] Rangarajan.C,Y. Matsuda, M. Pätzold, B. Häusler, N. Sugimoto,H. Ando,, M. Takagi, T. Imamura,H. Sagawa, S. Limaye, S. Tellmann,,, and M. Antonita, ''Thermal structure of the Venusian atmosphere from the sub-cloud region to the mesosphere as observed by radio occultation,'' Sci. Rep., vol. 10, no. 1, pp. 1–7, Dec. 2020, doi: 10.1038/s41598-020-59278-8.

[24] Subhan.D, R. Wang, C. Wei, Y. Wang, Y. Zhou, J. Wang, and H. Song, ''The connectivity evaluation among wells in reservoir utilizing machine learning methods,'' IEEE Access, vol. 8, pp. 47209–47219, 2020, doi: 10.1109/ACCESS.2020.2976910.

[25] Shakila.L, S. Tellmann, M. Pätzold, B. Häusler, N. Sugimoto,H. Ando,, M. Takagi, T. Imamura,H. Sagawa, S. Limaye, Y. Matsuda, R. K. Choudhary, and M. Antonita, ‘‘Thermal structure of the Venusian atmosphere from the sub-cloud region to the mesosphere as observed by radio occultation,’’ Sci. Rep., vol. 10, no. 1, pp. 1–7, Dec. 2020, doi: 10.1038/s41598-020-59278-8.

[26] Sarvesh.T , M. Pätzold, B. Häusler, N. Sugimoto,H. Ando,, M. Takagi, T. Imamura,H. Sagawa, S. Limaye, Y. Matsuda, R. K. Choudhary, and M. Antonita, ‘‘Thermal structure of the Venusian atmosphere from the sub-cloud region to the mesosphere as observed by radio occultation,’’ Sci. Rep., vol. 10, no. 1, pp. 1–7, Dec. 2020, doi: 10.1038/s41598-020-59278-8.

[27] Trisha.I, S. Tellmann, M. Pätzold, B. Häusler, N. Sugimoto,H. Ando,, M. Takagi, H. Sagawa, S. Limaye, Y. Matsuda, R. K. Choudhary, and M. Antonita, ‘‘Thermal structure of the Venusian atmosphere from the sub-cloud region to the mesosphere as observed by radio occultation,’’ Sci. Rep., vol. 10, no. 1, pp. 1–7, Dec. 2020, doi: 10.1038/s41598-020-59278-8.

[28] Yamie.D, Wang.J, L. Xiao, and T. Fu, ‘‘Short-term wind speed time series forecasting based on a hybrid method with multiple objective optimization for non-convex target,’’ Energy, vol. 215, Jan. 2021, Art. no. 119180, doi: 10.1016/j.energy.2020.119180.

[29] Yamal.I, Y. Xiang, L. Ali, and Z. Abdul-Rauf, ‘‘A bidirectional LSTM deep learning approach for intrusion detection,’’ Expert Syst. Appl., vol. 185, Dec. 2021, Art. no. 115524, doi: 10.1016/j.eswa.2021.115524.

[30] Yasir.M, M. Pätzold, B. Häusler, N. Sugimoto,H. Ando,, M. Takagi, T. Imamura,H. Sagawa, S. Limaye, S. Tellmann,, R. K. Choudhary, and M. Antonita, ‘‘Thermal structure of the Venusian atmosphere from the sub-cloud region to the mesosphere as observed by radio occultation,’’ Sci. Rep., vol. 10, no. 1, pp. 1–7, Dec. 2020, doi: 10.1038/s41598-020-59278-8.