Time Series Analysis on Air Quality and Weather Prediction

Master of Computer Application
In the Faculty of Engineering & Technology
Jadavpur University

By

ANIRBAN BERA

Exam Roll No**: MCA2360039**
Registration No**: 154245 of 2020-21**

Under the Guidance of
Prof. Ram Sarkar
Department of Computer Science & Engineering

Department of Computer Science & Engineering
Jadavpur University
Kolkata - 700 032

2023

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

FACULTY OF ENGINEERING & TECHNOLOGY
JADAVPUR UNIVERSITY

CERTIFICATE

I hereby recommend that the project entitled **"Time Series Analysis on Air Quality and Weather Prediction"** prepared under my supervision by **ANIRBAN BERA,** Exam- Roll No: **MCA2360039** be accepted for the degree of **Master of Computer Application** of **Jadavpur University, Kolkata**.

Supervisor

Prof. Ram Sarkar
Department of Computer Science & Engineering
Jadavpur University, Kolkata – 32

Prof. Nandini Mukhopadhyay
Head of the Department,
Computer Science & Engineering,
Jadavpur University, Kolkata-32

Prof. Ardhendu Ghoshal
Dean,
Faculty of Engineering & Technology,
Jadavpur University, Kolkata-32

JADAVPUR UNIVERSITY

FACULTY OF ENGINEERING & TECHNOLOGY

CERTIFICATE OF APPROVAL*

The foregoing project "**Time Series Analysis on Air Quality and Weather Prediction**" at instance is hereby approved as a creditable study of an engineering subject carried out and presented in a manner of satisfactory to warrant its acceptance as pre-requisite to the degree for which it has been submitted. It is notified to be understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed and conclusion drawn there in but approve the project only for the purpose for which it has been submitted.

Final Examination for the Evaluation of Project

Board of Examiners

_____

_____

(Signature of Examiners)

* Only in case project is approved

ACKNOWLEDGEMENTS

Though only my name appears on the cover of this dissertation, a great many people have contributed to its production. I owe my gratitude to all those people who have made this dissertation possible and because of whom my postgraduate experience has been one that I will cherish forever.

Foremost, I would like to express my profound gratitude and sincere thanks to my adviser, **Dr. Ram Sarkar**, **Professor, Department of Computer Science & Engineering**, **Jadavpur University**, for his valuable suggestions, guidance, constant encouragement and intent supervision at every stage of my work. I have been amazingly fortunate to have him as my project Supervisor who gave me the freedom to explore on my own, and at the same time guided me to recover when my steps faltered. It has been a great learning process for me.

Moreover, Prof. Ram Sarkar has been always there to listen and provide valuable advice. I am deeply grateful to him for the long discussions that helped me sort out the technical details of my work. I am thankful to him for his insightful comments and constructive criticisms at different stages of my research which were thought-provoking and helped me focus on my ideas. I am also thankful to him for encouraging the use of correct grammar and consistent notation in my writings and for carefully reading and commenting on countless revisions of this manuscript.

I am grateful to him for allowing me to use the computing facilities of "Center for Microprocessor Application for Training Education and Research (CMATER)" laboratory, Department of Computer Science and Engineering, Jadavpur University.

I am also thankful to the system staff who maintained all the machines in my lab so efficiently that I never had to worry about viruses, losing files, creating backups or installing software.

Many friends have helped me stay sane through these difficult years. Their support and care helped me overcome setbacks and stay focused on my postgraduate study. I greatly value their friendship and I deeply appreciate their belief in me.

Most importantly, none of this would have been possible without the love and patience of my family. My family, to whom this dissertation is dedicated to, has been a constant source of love, concern, support and strength all these years. I would like to express my heart-felt gratitude to my family.

Date:

Place: Kolkata

ANIRBAN BERA
M.C.A. (C.S.E)
Roll No: MCA2360039

Contents

ABSTRACT

Air quality and weather forecasting are two critical areas of research that have significant impacts on human health and the environment. Air pollution is a major concern in many urban areas, and its adverse effects on human health have been widely studied. Therefore, it is essential to develop accurate methods to predict air quality levels and provide early warnings to mitigate the effects of air pollution. Similarly, weather forecasting plays a crucial role in many fields, including agriculture, transportation, and disaster management. Accurate predictions of weather conditions can help prevent natural disasters, minimize their impact, and improve overall preparedness.

In this project work, the main focus is on the use of time series analysis techniques to predict air quality index (AQI) and weather conditions. Specifically, we explore the application of linear regression, decision tree, and long short-term memory (LSTM) neural networks for AQI and weather forecasting. We analyze historical data from multiple sources, including government agencies and public repositories, to train and evaluate these models.

Obtained results show that LSTM-based models outperform linear regression and decision tree models in terms of accuracy for both AQI and weather forecasting. We also demonstrate the importance of considering multiple factors in predicting AQI and weather conditions, as these factors can significantly impact the accuracy of the models. Additionally, we discuss the practical implications of our research, including the potential for early warning systems to be implemented based on our models to help mitigate the impact of air pollution and extreme weather conditions.

Overall, this project work highlights the significance of time series analysis for AQI and weather prediction and demonstrates the effectiveness of LSTM-based models in accurately forecasting these variables. We believe that our findings can contribute to the development of more accurate and effective methods for predicting air quality and weather conditions, with potential applications in multiple fields, including public health, transportation, and disaster management.

Chapter 1

1. Introduction

1.1. Time Series Analysis:

Time series analysis is an essential tool for understanding and forecasting complex phenomena that evolve over time. It has a rich history that spans several fields, including economics, engineering, and statistics, and has been applied to diverse domains such as finance, climate modeling, and healthcare.

Recent advances in time series analysis have been driven by the proliferation of data sources and the emergence of new challenges such as nonlinearity, high-dimensionality, and data heterogeneity. Researchers have developed new methods for handling these challenges and have improved the performance of existing techniques by leveraging insights from machine learning and deep learning.

One such method is the use of recurrent neural networks (RNNs) for time series analysis. RNNs are a class of neural networks that can model sequences of inputs and outputs and have been used for a variety of time series tasks, including forecasting, anomaly detection, and classification. For instance, Lipton et al. (2015) [1] showed that RNNs can learn to predict stock prices and achieve superior performance compared to traditional time series models.

Another recent development in time series analysis is the use of deep learning methods for feature extraction and dimensionality reduction. By combining deep learning with classical time series analysis techniques, researchers have been able to identify hidden patterns in data and improve the accuracy of time series models. For example, Che et al. (2018) [2] proposed a hybrid method that uses convolutional neural networks (CNN) for feature extraction and time series models for prediction.

Despite the growing interest in deep learning for time series analysis, classical statistical methods such as autoregressive integrated moving average (ARIMA) and exponential smoothing (ETS) remain popular and effective for many time series tasks. In particular, they are often used as benchmarks for comparing the performance of new methods (Hyndman and Athanasopoulos, 2018) [3].

This project aims to provide a comprehensive overview of time series analysis, with a focus on the latest developments in deep learning and statistical modeling. We will review the fundamentals of time series analysis, including stationarity, autocorrelation, and spectral analysis, and cover advanced topics such as multivariate time series analysis, nonlinearity, and time-varying models. We will also discuss the practical applications of time series analysis in various domains, including finance, climate modeling, and healthcare.

1.2. Time Series Analysis for Air Quality Prediction:

Air pollution is a major global health concern, with outdoor air pollution contributing to approximately 4.2 million premature deaths each year (WHO, 2016). AQI is a measure of air quality that describes the level of pollutants in the air and their potential impact on health. AQI is calculated based on the concentrations of several pollutants, such as ozone (O_3), nitrogen dioxide (NO_2), and particulate matter (PM2.5 and PM10), and is reported on a scale from 0 to 500.

The prediction of AQI is an important task for ensuring public health and safety, as well as for enabling effective policy and decision making. Accurate and timely AQI predictions can help individuals and authorities take necessary measures to reduce exposure to air pollution, such as avoiding outdoor activities, using protective equipment, or implementing pollution control measures.

Recent advances in machine learning and data analytics have enabled the development of new methods for AQI prediction, based on the analysis of historical air quality data, meteorological data, and other relevant factors. For example, Shao et al. (2019) [4] proposed a deep learning method based on CNN for AQI prediction, achieving superior performance compared to traditional regression models. Chen et al. (2017) [5] used a hybrid method that combined support vector regression (SVR) and autoregressive integrated moving average ARIMA to predict AQI in Beijing, China, achieving high accuracy and reliability. Despite these advances, AQI prediction remains a challenging task due to the complexity and variability of air pollution patterns and the influence of various environmental and social factors. Furthermore, the availability and quality of air quality and meteorological data can vary across regions and countries, posing additional challenges for developing accurate and robust AQI prediction models. Here we provide a comprehensive overview of AQI prediction, with a focus on the latest developments in machine learning and data analytics. We will review the fundamental concepts and methods for air quality analysis and modeling, including pollutant dispersion modeling, atmospheric chemistry modeling, and time series analysis.

1.3. Time Series Analysis for Weather Prediction:

Weather prediction is the process of estimating the state of the atmosphere for a future time and location, based on the current state and known physical laws. It plays a critical role in daily life, as it informs decisions related to agriculture, transportation, energy, and more. Accurate weather prediction can help save lives and property, while inaccurate predictions can lead to significant economic and social costs.

Traditionally, weather prediction has been based on numerical weather models that solve a set of mathematical equations to simulate the physical processes that govern the atmosphere. These models take into account various meteorological factors, such as temperature, humidity, pressure, wind speed and direction, and precipitation. However, weather forecasting is a complex and challenging problem, and despite significant advances in the field, it remains difficult to predict the weather accurately beyond a few days. In recent years, machine learning techniques have been applied to weather prediction with promising results. These techniques leverage historical weather data to learn patterns and relationships that can be used to make predictions. Machine learning models can handle large amounts of data and capture complex nonlinear relationships that may be difficult for traditional models to capture. In particular, deep learning models such as CNN and recurrent neural networks RNN have been shown to be effective for weather prediction.

Overall, weather prediction is a challenging and important problem, with significant implications for many aspects of society. With continued advancements in numerical modeling and machine learning, we can expect to see continued progress in weather forecasting accuracy and reliability.

1.4. Applications

In this section, we will explore some practical applications of time series analysis, specifically in the domains of air quality and weather prediction. Time series analysis can help in accurately predicting future trends and patterns, which can be beneficial in making informed decisions for various applications. Let's take a closer look at some examples

- **Finance:** Time series analysis is widely used in finance for forecasting stock prices, interest rates, and currency exchange rates. It can help investors and traders make informed decisions about buying and selling securities, as well as help financial institutions manage their risk.
- **Energy:** Time series analysis can be used to forecast energy demand and supply, such as electricity load forecasting, wind power forecasting, and solar power forecasting. This can help energy companies optimize their production and distribution, as well as help policymakers make informed decisions about energy policy.
- **Traffic:** Time series analysis can be used to predict traffic flow and congestion patterns, helping to optimize traffic management and reduce congestion. This can help reduce travel time for commuters and improve air quality in urban areas.

- **Healthcare:** Time series analysis can be used in healthcare to forecast patient volumes, disease outbreaks, and medical supply demand. This can help hospitals and healthcare providers allocate resources more effectively and respond to public health crises more quickly.
- **Retail:** Time series analysis can be used to predict consumer demand and sales trends, which can help retailers optimize inventory levels, plan promotions, and manage their supply chain. This can help reduce waste and improve profitability for retailers.
- **Agriculture**: Weather prediction can be used in agriculture to forecast crop yields, water demand, and disease outbreaks, which can help farmers optimize irrigation, fertilization, and pesticide applications. This can help improve crop quality and quantity, as well as reduce the use of water and other resources.
- **Aviation:** Weather prediction can be used in aviation to predict weather patterns and turbulence, helping to optimize flight paths and improve safety. This can help reduce flight delays and cancellations, as well as improve the passenger experience.
- **Disaster response:** Time series analysis can be used in disaster response to predict the impact of natural disasters, such as hurricanes, floods, and earthquakes, and help emergency responders allocate resources and plan evacuations. This can help reduce the loss of life and property damage during natural disasters.
- **Environmental monitoring:** AQI prediction can be used in environmental monitoring to forecast air quality levels and identify areas of high pollution, helping policymakers develop effective air pollution control measures. This can help reduce the health risks associated with air pollution and improve overall environmental quality.

These examples demonstrate how time series analysis, AQI prediction, and weather prediction can be applied in various fields to optimize resource allocation, protect public health, and improve disaster management.

1.5. Motivation

The prediction of air quality and weather conditions is of great importance to public health, environmental protection, agriculture, transportation, and many other domains. Accurate forecasts can help individuals and organizations to plan their activities, avoid potential hazards, reduce energy consumption, and make informed decisions. However, the prediction of these variables is often challenging due to their complex and dynamic nature, the presence of various

sources of uncertainty and variability, and the need for high spatiotemporal resolution.

To overcome these challenges, time series analysis has emerged as a powerful tool for modeling and forecasting air quality and weather data. Time series models can capture the temporal dependencies and patterns in the data, account for the effects of external factors and events, and provide probabilistic predictions with associated uncertainties. Moreover, recent advances in machine learning and deep learning have enabled the development of sophisticated models that can handle large-scale, heterogeneous, and missing data, and achieve state-of-the-art performance.

In this study, we have applied three different models to predict AQI and weather variables: linear regression, decision tree, and LSTM. Linear regression is a simple but powerful model that has been widely used in time series analysis. Decision trees are a popular method for predicting time series data and have been shown to be effective in various applications. LSTM, on the other hand, is a type of recurrent neural network that is specifically designed to handle time series data with long-term dependencies.

Each of these models has its own strengths and weaknesses, and we evaluated their performance on our dataset. By comparing the results, we were able to identify the most suitable model for predicting AQI and weather variables. Our findings provide insights into the effectiveness of different models for time series analysis and can inform future research in this area.

1.6. Scope of Present Work

The scope of this present work is to develop and compare different time series analysis models for predicting the AQI and weather conditions. Specifically, the models to be compared include linear regression, decision tree, and LSTM neural network. The study will focus on using historical AQI and weather data to forecast future AQI and weather conditions, and evaluating the accuracy of the developed models. The research will also explore the impact of different weather Parameter, such as temperature, humidity, wind speed, and precipitation, on AQI prediction. Additionally, the study will investigate the effectiveness of using AQI and weather forecasts as inputs for predicting each other, which can have significant implications for urban planning and public health management.

1.7. Organization of Project Work

In this thesis, we present a study on time series analysis for AQI and weather prediction. The goal of this work is to develop accurate prediction models that can help improve public health and safety by providing timely and reliable

information on air quality and weather conditions. This work can be divided into following chapters.

- **Chapter 1: Introduction**
  This chapter provides an overview of the research problem, objectives, and motivation behind the study. It also outlines the scope and limitations of the research, and presents a brief overview of the methodologies and approaches used.
- **Chapter 2: Related Work**
  This chapter reviews the existing work on time series analysis, AQI, and weather prediction. It discusses the various models and techniques used in these fields and highlights the strengths and limitations of each approach. The chapter also examines the recent developments in these areas and discusses the research gaps and opportunities for future research.
- **Chapter 3: Working Methodologies**
  This chapter presents the methodologies and approaches used in the study, including the models and techniques employed for time series analysis, AQI prediction, and weather prediction. It also discusses the model development process, including parameter tuning and model evaluation.
- **Chapter 4: Data Collection, Preprocessing, Result and Analysis**
  This chapter describes the data collection process, including the sources and types of data used in the study. It also discusses the steps taken to preprocess and clean the data, including missing data imputation, data normalization, and feature selection.
  This chapter also presents the results and analysis of the study, including the performance of the developed models in predicting AQI and weather variables. It also discusses the implications of the results and provides insights into the relationships between air quality index, weather variables, and their impact on human health.
- **Chapter 5: Conclusion and Future work**
  This chapter summarizes the key findings and contributions of the study and discusses the limitations and future directions for research. It also provides recommendations for policymakers and practitioners in the field of air quality management and weather forecasting.

Chapter 2

2. Related Work

This chapter provides a comprehensive literature overview on time series analysis related to the prediction of AQI and weather conditions. AQI is a crucial indicator of air pollution, and accurate forecasting plays a vital role in addressing environmental and health concerns. Weather variables such as temperature, humidity, wind speed, and precipitation significantly influence air quality, making their inclusion in prediction models crucial. In this chapter, we review a selection of eleven studies that employ various machine learning techniques to forecast AQI and incorporate weather data for enhanced prediction accuracy.

Box and Jenkins (1976) [6]provided a comprehensive foundation for time series analysis techniques in their book "Time Series Analysis: Forecasting and Control."

Chatfield (2003) [7] further expanded on the analysis of time series, introducing key concepts and methods.

Lütkepohl (2005) [8] contributed to the field with the introduction of multiple time series analysis, while Shumway and Stoffer (2017) [9] presented practical applications of time series analysis using R examples.

In the realm of stock market prediction, the study by Chatterjee et al. ("Stock market prediction using Altruistic Dragonfly Algorithm") [10] introduces the Altruistic Dragonfly Algorithm as a computational tool to optimize stock market prediction models. Their research demonstrates the potential effectiveness of this algorithm in enhancing forecasting accuracy. By incorporating their findings alongside other relevant studies, we aim to develop an innovative stock market prediction model that leverages the strengths of different algorithms and techniques, contributing to the advancement of accurate and reliable stock market forecasting methodologies.

Among the studies focusing on human activity recognition using smartphone sensors for healthcare applications, an article by Mukherjee et al. ("EnsemConvNet: a deep learning approach for human activity recognition using smartphone sensors for healthcare applications") [11] presents the EnsemConvNet model for accurately recognizing human activities based on smartphone sensor data. Their research demonstrates the effectiveness of this deep learning-based approach, which combines ensemble learning with convolutional neural networks. By considering the findings of Mukherjee et al. (2020) alongside other relevant studies, we aim to enhance our understanding and develop robust models for human activity recognition in healthcare applications.

The prediction of AQI has garnered significant attention in recent years. Researchers have explored various time series analysis techniques to accurately forecast AQI values.

The article by Maltare and Vahora ("Air Quality Index prediction using machine learning for Ahmedabad city") focuses on using machine learning algorithms to forecast the AQI specifically for Ahmedabad city. Their research contributes to understanding and predicting air quality dynamics, enabling proactive measures for pollution control and public health management. By considering the findings of Maltare and Vahora (2020) [12] alongside other relevant studies, we aim to enhance our understanding and develop effective models for AQI prediction in diverse urban areas.

The machine learning algorithms are widely used to predict, forecast and control pollution level. Three different machine learning algorithms are discussed as forecasting models for ground level ozone, nitrogen dioxide, and sulfur dioxide by Shaban et al. (2016) [13].

In Bekkar et al. (2021) [14] deep learning algorithms like LSTM, Bi-directional LSTM, GRU, Bidirectional GRU, CNN, and a hybrid CNN-LSTM models were examined for predicting PM2.5 concentration. The hybrid CNN-LSTM multivariate technique works better in terms of predictive performance and allows for more accurate predictions than any of the classical models on UCI Machine Learning Repository.

In the field of time series analysis and weather prediction, numerous studies have been conducted to improve our understanding of temporal patterns, enhance forecasting accuracy, and explore new methodologies. This literature review discusses five notable examples that cover a range of topics, including probabilistic forecasting, ensemble methods, recurrent neural networks, and machine learning techniques. These studies contribute valuable insights and advancements in the field, offering significant implications for weather prediction and time series analysis.

Delle Monache et al. (2013) [15] conducted a study on probabilistic weather prediction with an analog ensemble, which was published in the Monthly Weather Review. The authors propose a novel approach that combines the concepts of analog forecasting and ensemble forecasting. They use historical weather data to identify analogs, which are similar past weather patterns, and construct an ensemble forecast based on these analogs. The study demonstrates the effectiveness of the analog ensemble approach in improving the accuracy and reliability of probabilistic weather predictions.

Chapter 3

3. Working Methodologies

Machine Learning is a branch of Artificial Intelligence that aims to provide computers with the ability to learn how to perform specific tasks without being explicitly programmed by a human. This technique is based on the design of models that learn from data and make decisions or predictions when new data are available. Deep Learning can be seen as an evolution of machine learning that uses a structure of multiple layers called Artificial Neural Network (ANN). Deep learning algorithms require less involvement of humans because features are automatically extracted. However, an important difference with respect to other machine learning techniques is that deep learning requires massive data to work properly.

Although deep learning are recent concepts, the first computer learning program was written by Arthur Samuel in 1952 and the first neural network was proposed by Frank Rossenblatt in 1957[1]. Since the 1990s, the development in both ML and deep learning has been significant, mainly due to the increment of computation power and the availability of large amounts of data.

There exist many machine learning approaches that can be applied to solve different problems. In this section, we will discuss only those algorithms that have been used for predicting pollutant measures and rain prediction. We can distinguish between the ones based on regression analysis and the ones using neural networks. Moreover, in the first category we will distinguish between the use of classical regression algorithms and machine learning algorithms.

3.1. Machine Learning Based Regression Algorithms

Regression analysis is used to infer the relation between a dependent variable and a set of independent variables. On the basis of this relation, and using the values of the independent variables, the value of the dependent variable is estimated. Regression helps to predict a continuous value. Next, we review classical algorithms to carry out regression.

- **Linear Regression (LR):** Let y and x be the dependent variable and the independent variables, respectively. The goal of linear regression is to define a linear function f (x) that minimize the square mean error, that is, $min\ [(y - f(x))^2]$.

  The general equation of LR is:

  $$y = \beta_0 + \beta_1 x + \varepsilon$$

  - y is the predicted value of the dependent variable (y) for any given value of the independent variable (x)
  - β_0 is the intercept, the predicted value of y when the x is 0.
  - β_1 is the regression coefficient – how much we expect y to change as x increases.

[1] https://www.dataversity.net/a-brief-history-of-machine-learning/, last accessed on 15-May-23

- x is the independent variable (the variable we expect is influencing y).
- ε is the error of the estimate, or how much variation there is in our estimate of the regression coefficient.

  The goal of linear regression is to find the values of β_0 and β_1 that minimize the sum of the squared differences between the actual values of y and the predicted values of y based on the line of best fit. This is done using the method of least squares. In **Figure 1**[2] shows a sample graph of linear regression.

3

Figure 1 Illustration of Linear regression

- **Multiple Linear Regression (MLR):** Let y and $x_1, \ldots, x_p$ be the dependent variable and the independent variables, respectively. The goal of linear regression is to define a linear function f (x_1, ..., xp) that minimize the square mean error, that is, $min\ [(y - f(x_1, \ldots, x_p))^2]$.

  The general equation of multiple linear regression (MLR) is:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon$$

  - y is the predicted value of the dependent variable.
  - β_0 is the y-intercept (value of y when all other Parameter are set to 0).
  - $\beta_1 x_1$ is the regression coefficient (β_1) of the first independent variable (x_1) (a.k.a. the effect that increasing the value of the independent variable has on the predicted y value)
  - $\beta_n x_n$ is the regression coefficient of the last independent variable.
  - ε is the error of the estimate, or how much variation there is in our estimate of the regression coefficient.

  To estimate the coefficients $\beta_0, \beta_1, \beta_2, \ldots, \beta_n$, the least squares method is employed. This method minimizes the sum of the squared differences between the observed y values and the predicted y values based on the

[2] https://www.javatpoint.com/simple-linear-regression-in-machine-learning, last accessed on 16-May-2023.

li3near equation. The coefficients are estimated by solving a system of equations or using numerical optimization techniques.

- **Decision Tree:** The aim of this algorithm is to design a model for predicting a quantitative variable from a set of independent variables. The algorithm is based on a recursive partitioning. Trees are composed of decision nodes and leaves. Regression usually is built by considering the standard deviation reduction to determine how to split a node in two or more branches. The root node is the first decision node that is divided on the basis of the most relevant independent variable. Nodes are split again by considering the variable with the less sum of squared estimate of errors (SSE) as the decision node. The dataset is divided based on the values of the selected variable. The process finishes when a previously established termination criterion is satisfied. The last nodes are known as leave nodes and provide the dependent variable prediction. This value corresponds to the mean of the values associated to leaves. In **Figure 2** Balogun and Tella (2022) [16] shows a graphical representation of the general structure of a standard Decision Tree.

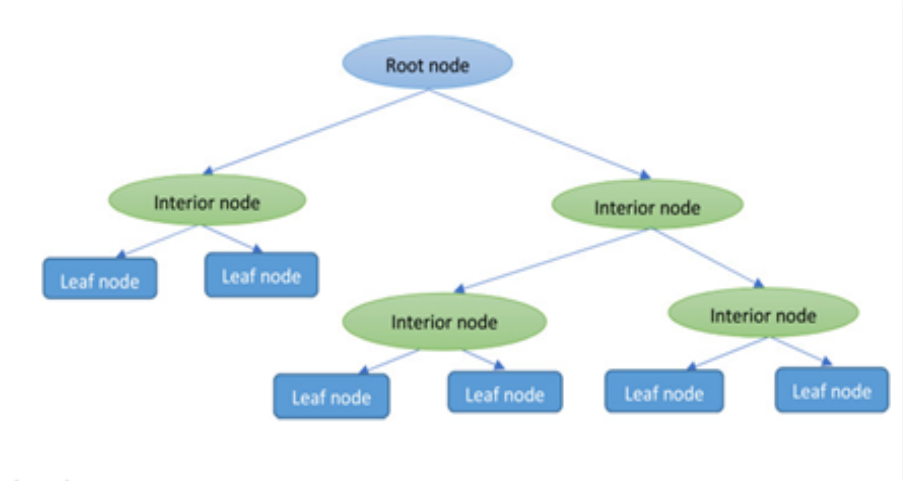

Figure 2 Decision tree structure

3.2. Deep Learning Models

Deep learning algorithms use ANNs. In this section, we will briefly describe different types of ANNs used in the literature for air quality prediction. In order to understand the internal behavior of an ANN, we will introduce its structure.

An ANN is an algorithm based on the biological neuronal connections that are comprised of neurons or nodes. These connections are organized in three-layer types. The input layer receives as input the original predictor variables. The output layer produces the predicted value for the given inputs. These two layers are connected by the hidden layers. The hidden layers (more than one in the case of deep learning) contain non-observable neurons which are in charge of the computation. Each node in a layer is connected with nodes in the next layer. Each connection has associated a weight that is used to combine the inputs. Each node

or neuron in the next layer receives the weighted value and transforms it by means of an activation function. Sigmoid and rectified linear unit functions are the most popular. The obtained result is the value that is passed as input to the nodes in the next layer. This process continues until the output layer is reached. At this point, the output prediction is produced.

The final aim of an ANN is to fit the weights to minimize an error function, commonly a quadratic function. To do this, the ANN uses the known as back-propagation algorithm. This algorithm employees the gradient descendent method using the layers partial derivatives to find the optimal weight of each node. Now, we describe the different types of ANNs that have been used for predicting pollutant measures and rain predictions.

- **Recurrent Neural Networks (RNN):** RNNs deal with either time series or sequential data, that is, information ordered and related. RNNs have an internal memory in the sense that a neuron can feedback itself by receiving as input the output it has previously produced. This allows the model to acquire short-term memory which is essential to time series forecasting.
- **Long-Short Term Memory Neural Networks (LSTM):** LSTMs are an extension of RNNs. They have an extended memory that allows to deal with long-term dependencies. LSTMs can remember information over arbitrary time intervals. The core component is the cell state that carries information throughout the processing of the data. The information is updated on the basis of three gates. Each of them controls the information that should be in the cell state using a sigmoid activation function. The forget gate determines which part of the previous state information should be forgotten. The input gate decides the new information that will be used to update the memory. Using a hyperbolic-tangent function, it creates a vector candidate to be added to the cell state. The last gate, known as output gate, uses a tanh activation function for determining which part of the updated cell state will be used as output. Now we discuss about each gate of the LSTM models[3].
  - **Input Gate:** It discover which value from input should be used to modify the memory. Sigmoid function decides which values to let through 0 or 1. And tanh function gives weightage to the values which are passed, deciding their level of importance ranging from -1 to 1.

$$i_t = \sigma(w_i . [h_{t-1}, x_t] + b_i)$$
$$c_t = \tanh(w_c . [h_{t-1}, x_t] + b_c)$$

    - i_t is input gate.
    - σ is sigmoid function.
    - w_i is weight for the input gates.
    - h_{t-1} is output of the previous LSTM block (at timestamp t-1).
    - x_t is input at current timestamp.

[3] https://www.javatpoint.com/long-short-term-memory-rnn-in-tensorflow, last accessed on 16-May-2023.

- b_i is biases for the input gate.
- c_t is cell state (memory) at timestamp (t).

➢ **Forget Gate:** It discover the details to be discarded from the block. A sigmoid function decides it. It looks at the previous state (h_{t-1}) and the content input (x_t) and outputs a number between 0 (omit this) and 1(keep this) for each number in the cell state c_{t-1}.

$$f_t = \sigma\left(w_f . [h_{t-1}, x_t] + b_f\right)$$

- f_t is output gate.
- σ is sigmoid function.
- w_f is weight for the forget gate.
- h_{t-1} is output of the previous LSTM block (at timestamp t-1).
- x_t is input at current timestamp.
- b_f is biases for the forget gate.

➢ **Output gate:** The input and the memory of the block are used to decide the output. Sigmoid function decides which values to let through 0 or 1. And tanh function decides which values to let through 0, 1. And tanh function gives weightage to the values which are passed, deciding their level of importance ranging from -1 to 1 and multiplied with an output of sigmoid.

$$o_t = \sigma(w_o . [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh(c_t)$$

- o_t is output gate.
- σ is sigmoid function.
- w_o is weight for the output gate.
- h_{t-1} is output of the previous LSTM block (at timestamp t-1).
- b_o is biases for the output gate.
- c_t is cell state (memory) at timestamp (t).

In **Figure 3** shows a graphical representation of a LSTM model.

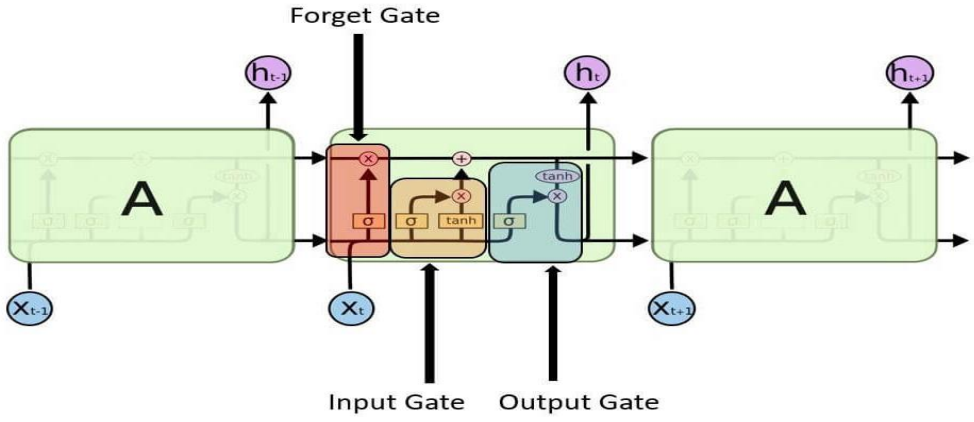

Figure 3 Architecture of a LSTM Model

Chapter 4

4. Results and Discussion

In this section, we take a closer look at the datasets we used in our study. We provide detailed information about where the data came from, how we collected it, and the steps we took to prepare it for analysis. We then present the results sharing what we found and how well different models and algorithms performed. This chapter gives a comprehensive overview of the datasets we used and offers a critical examination of the outcomes and their implications for AQI and weather prediction.

4.1. Evaluation Metrics

In this study, Mean Absolute Error, Mean Squared Error, and Root Mean Squared Error metrics have been used. These validation metrics are used to determine whether regression models are accurate or misleading.

- **Mean Absolute Error (MAE):** A measure of errors between paired observations is called mean absolute error (MAE) [17]. The larger MAE is the larger the error. The error is the difference between predicted value ($\hat{y}_i$), actual value (y_i) and n is total number of observations.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$

- **Mean Squared Error (MSE):** Mean squared error (MSE) is a popular evaluation metric in regression analysis, used to quantify the average squared difference between the predicted values and the true values. It is calculated by taking the sum of the squared residuals (the differences between the predicted and true values) and dividing it by the total number of observations. The formula for MSE is:

$$MSE = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}$$

  where n is the number of observations, y represents the true values, and $\hat{y}$ denotes the predicted values. The MSE provides a comprehensive measure of the model's accuracy, with larger errors being emphasized due to the squaring operation. Minimizing the MSE during model training helps to identify the best-fitting model that minimizes the overall squared differences, indicating a closer alignment between the predicted and actual values.

- **Root Mean Squared Error (RMSE):** The measure of how well a regression line fits the data points is called RMSE [18]. RMSE can alternatively be thought of as the residuals' standard deviation. The square root of the mean of the squares of the errors is used to calculate RMSE of

projected values for time t of the dependent variable y_t in a regression using variables observed over T periods.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}}$$

4.2. Dataset Descriptions

The use of an appropriate dataset is very much necessary to prove the usefulness of any method. To this end, three publicly available datasets are used to evaluate performance of the proposed method. The first one is an air quality prediction data from which we will predict air quality of West Bengal.[4] The second one[5] is the dataset of weather prediction dataset of city Austin, a city in Texas [19], USA from which we will predict the rainfall, and the third dataset[6] is the city of Szeged [20], a city in Hungary from which we will predict the humidity using temperature. The dataset distribution is given in **Table 1**. During the training of the individual regression models, 20% of the training samples are typically used as validation samples, and the rest are used for model training.

Table 1 Distribution of training and test sets for all three datasets used here.

Dataset	Train Sample	Test Sample
AQI dataset	17970	4493
Austin weather dataset	1055	264
Szeged weather dataset	400	100

4.3. Data Preprocessing and Visualization

Pre-processing contributes in transforming the data into an efficient input format that will be fed to the model. The different pre-processing methods used in this research work are elaborated in this section.

- **AQI Dataset:**

  For this dataset at first, we check if there is any missing value is present or not, and we see there are lots of missing value present in the dataset. Deleting all rows having invalid or missing values can result in information loss, which would result in imprecise output [21]. So, we fill all rows having invalid or missing values using zero or one. After that, we count the industrial and residential areas in our datasets which is shown in **Figure 4.**

[4] https://www.kaggle.com/datasets/shrutibhargava94/india-air-quality-data, last accessed on 17-May-2023.
[5] https://www.kaggle.com/datasets/grubenm/austin-weather, last accessed on 17-May-2023.
[6] https://www.kaggle.com/datasets/budincsevity/szeged-weather, last accessed on 17-May-2023.

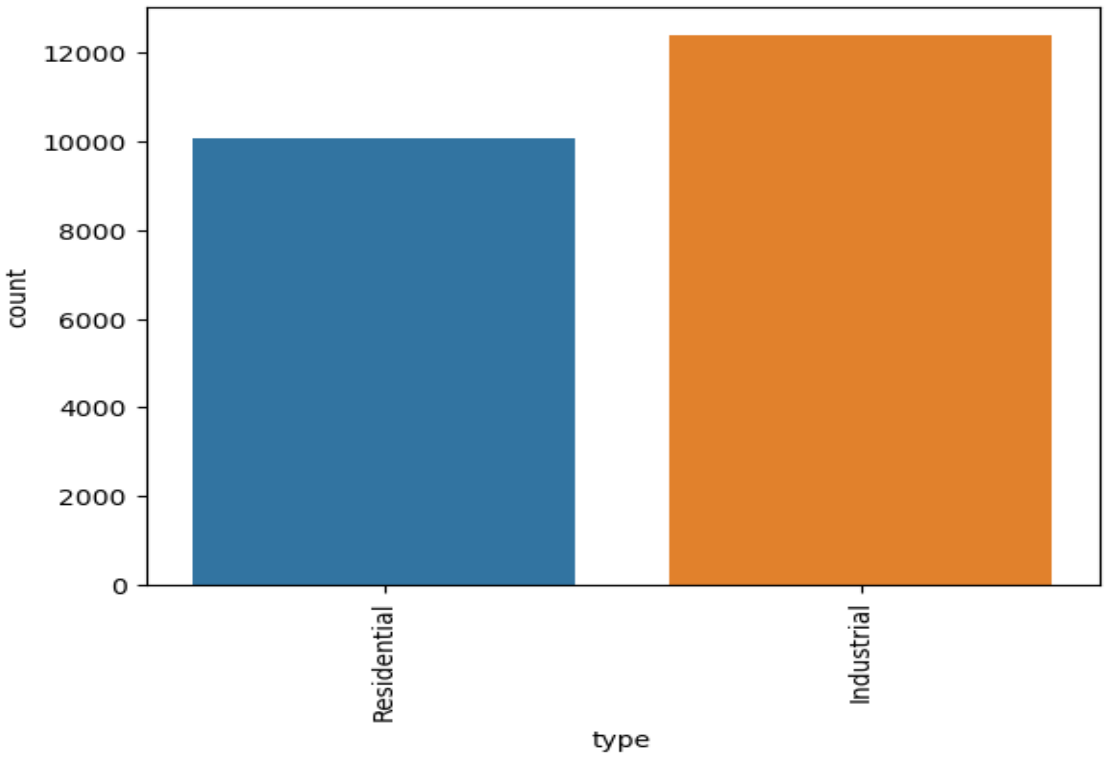

Figure 4 Count the type of areas in data

Next, we count location wise industrial and residential areas in our datasets which is shown in **Figure 5**

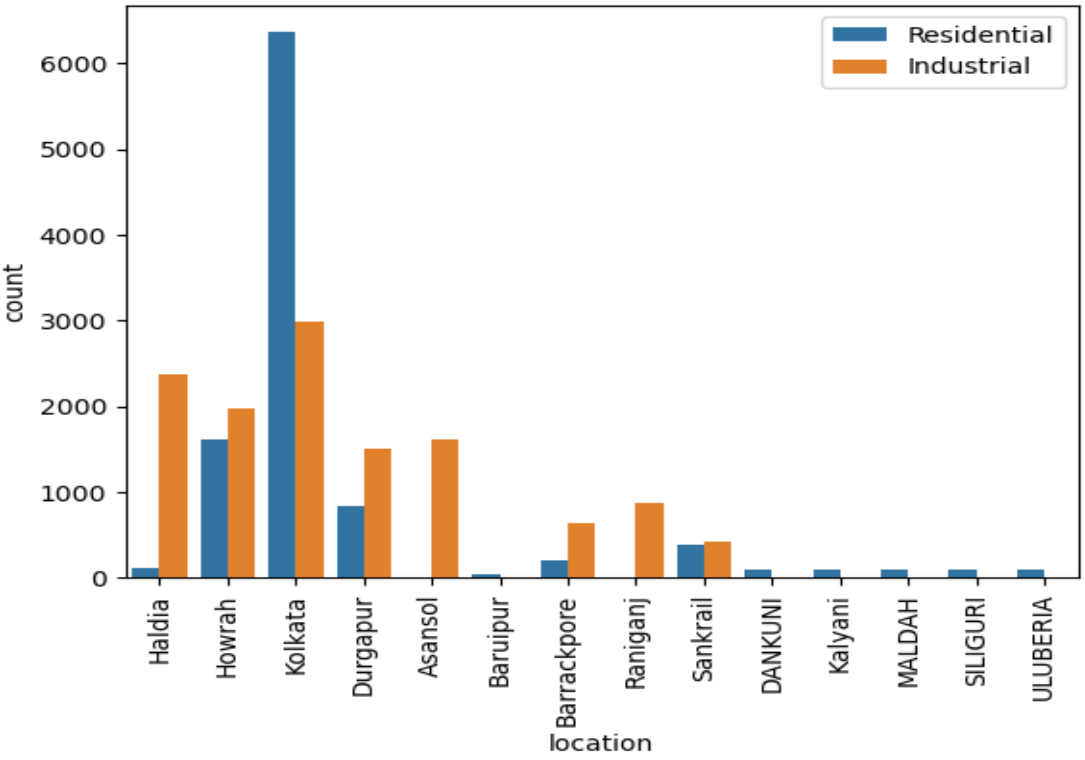

Figure 5 Location wise types of areas

Now we see the location wise values of NO_2, SO_2, and SPM values which is shown in **Table 2**, **Figure 6**, **Figure 7** and **Figure 8** respectively.

Table 2 Location wise values of NO_2, SO_2 and SPM

Location	NO_2	SO_2	SPM
Asansol	57.749410	7.944357	291.123037
Barrackpore	58.591637	14.226796	281.823753
Baruipur	68.500000	9.100000	576.000000
DANKUNI	43.679612	15.087379	47.000000
Durgapur	58.247069	8.656687	295.172845
Haldia	43.904346	12.261811	194.541993
Howrah	77.147545	16.595665	254.378141
Kalyani	36.038462	2.894231	47.000000
Kolkata	57.938711	11.294017	233.541559
MALDAH	16.346154	2.509615	47.000000
Raniganj	58.851869	34.607814	319.939977
SILIGURI	17.932692	3.288462	47.000000
Sankrail	57.430311	9.472174	340.607660
ULUBERIA	43.317308	14.615385	47.000000

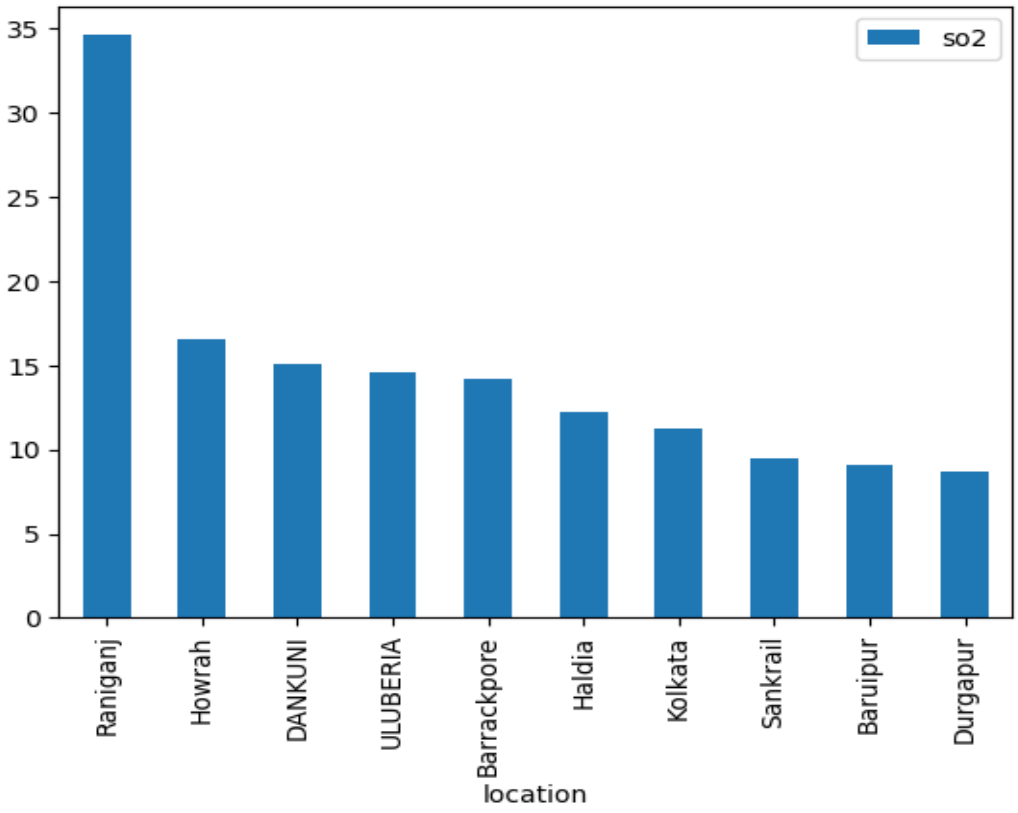

Figure 6 SO_2 in each location

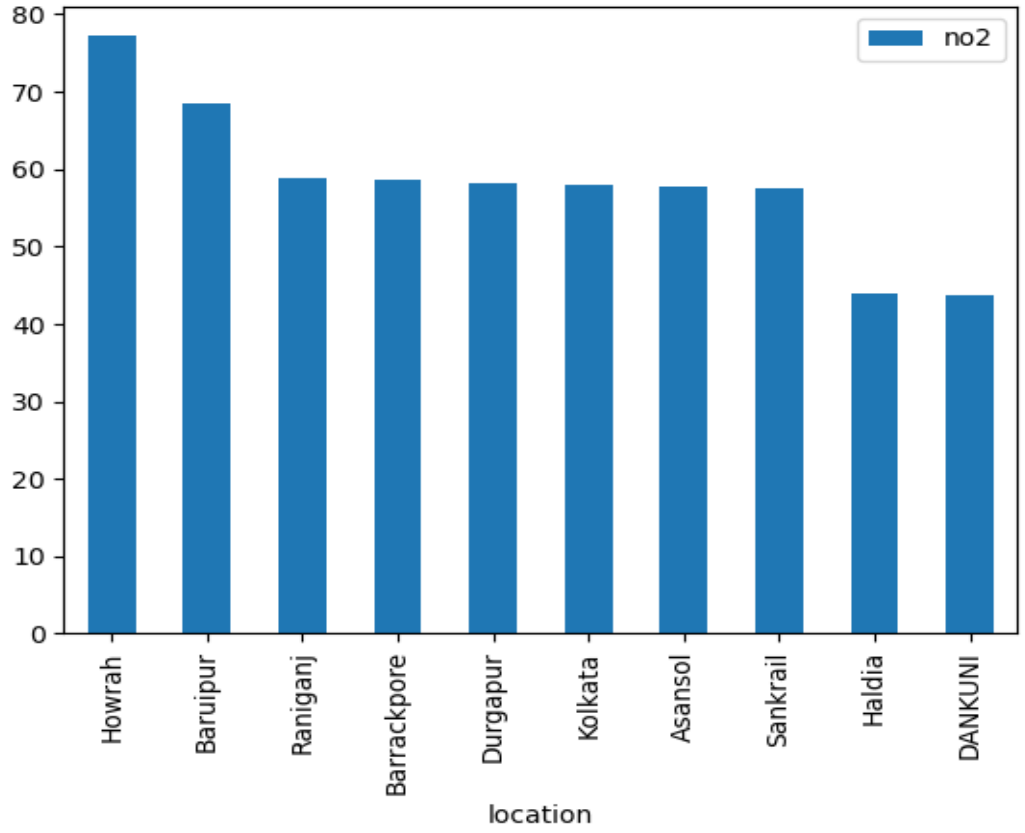

Figure 7 NO_2 in each location

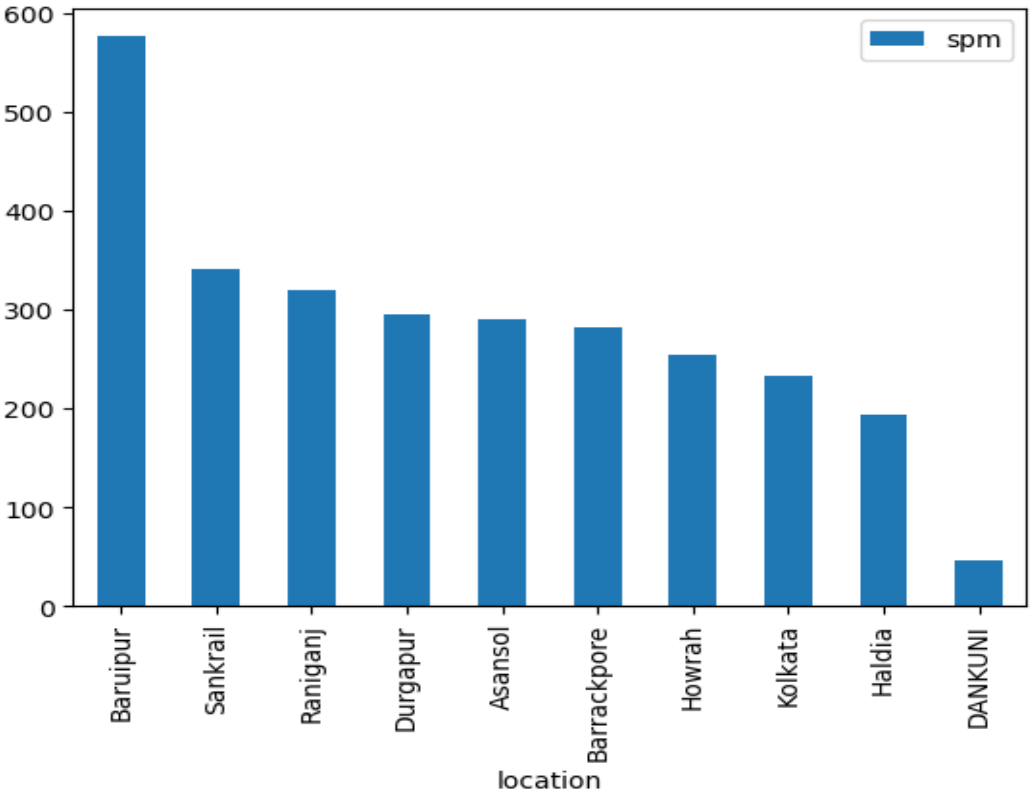

Figure 8 SPM in each location

Next, we calculate the AQI of all the location which is shown in **Table 3** and **Figure 9** respectively.

Table 3 AQI of all location

Location	AQI
Asansol	292.552434
Barrackpore	284.544602
Baruipur	576.000000
DANKUNI	48.242718
Durgapur	295.478682
Haldia	194.747506
Howrah	255.964923
Kalyani	49.269231
Kolkata	238.754075
MALDAH	47.000000
Raniganj	322.476029
SILIGURI	46.548077
Sankrail	340.610352
ULUBERIA	48.961538

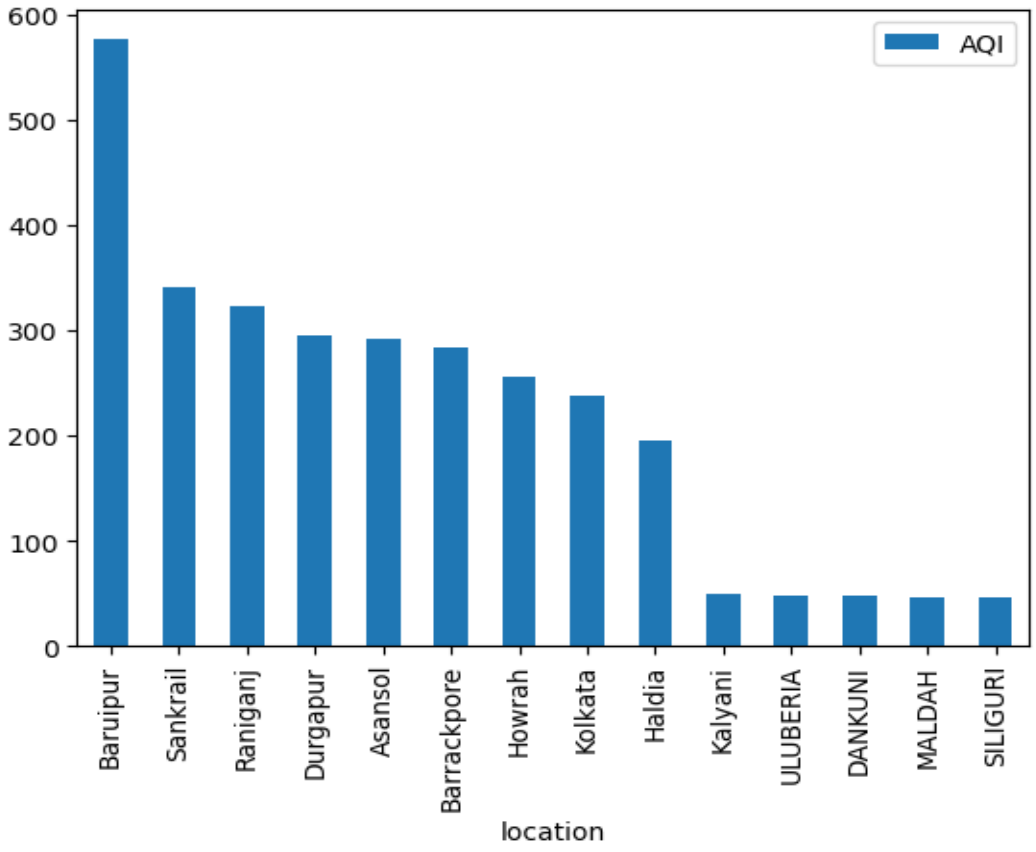

Figure 9 AQI of all location

Now, we see the year wise AQI in **Figure 10**.

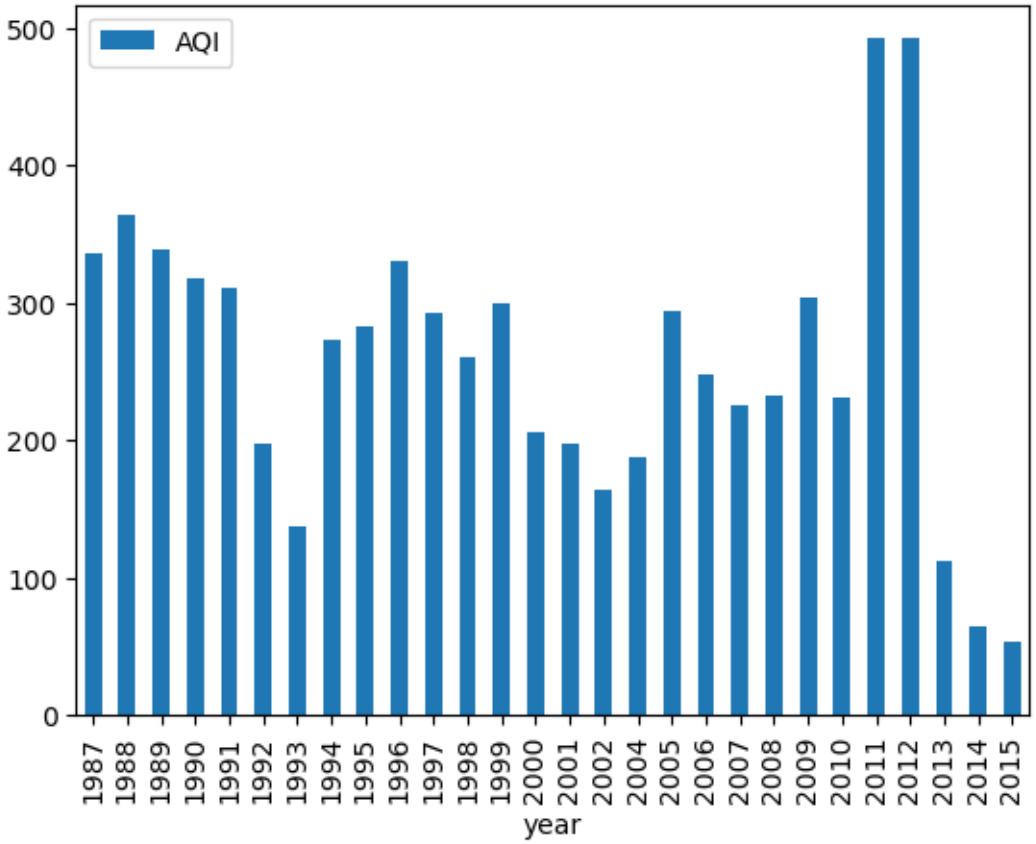

Figure 10 AQI in each year

- **<u>Austin Weather Dataset</u>:**

  For this dataset, we check if there is any missing value is present or not. We found that there is no missing value but there is some missing rows and there is some character values in the dataset. Now we see the density of rainfall which is shown in **Figure 11**.

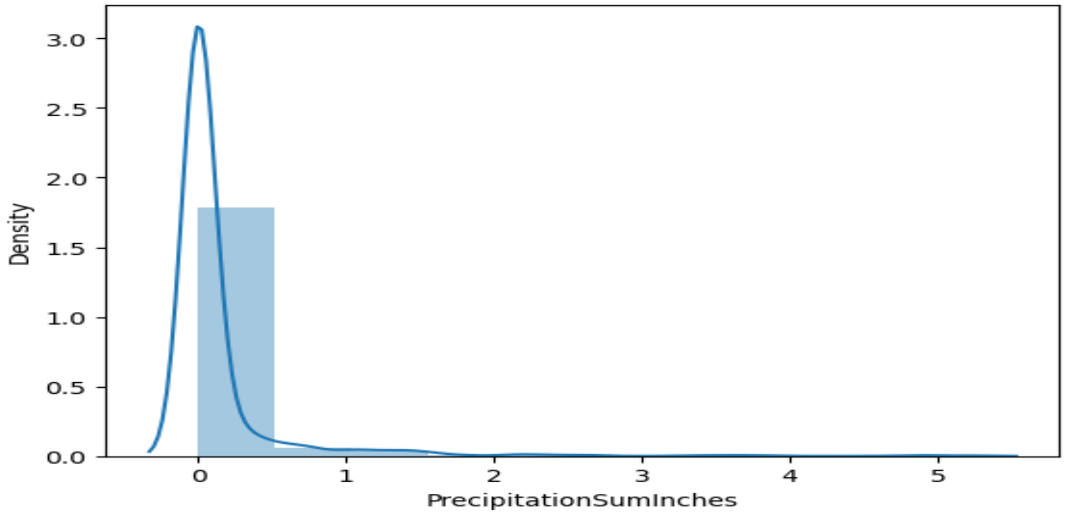

Figure 11 Density of precipitation

Now, we boxplot precipitation which is shown in **Figure 12.**

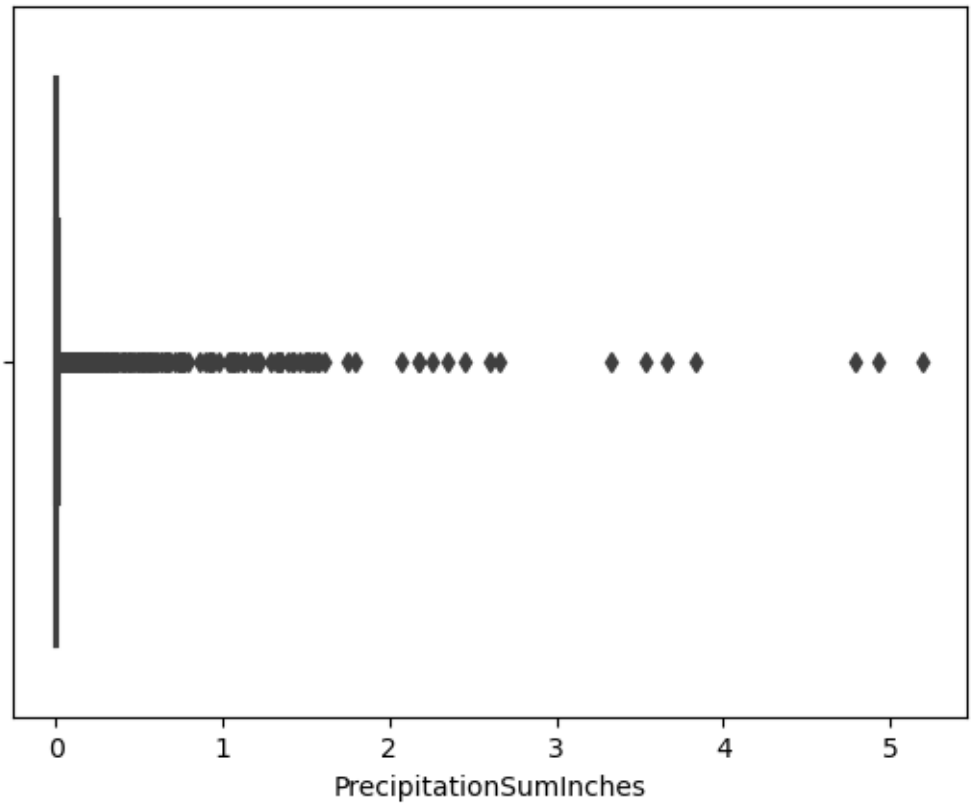

Figure 12 Boxplot of precipitation

Now, we see the month wise precipitation and box plotting them which is shown in **Figure 13** and **Figure 14**.

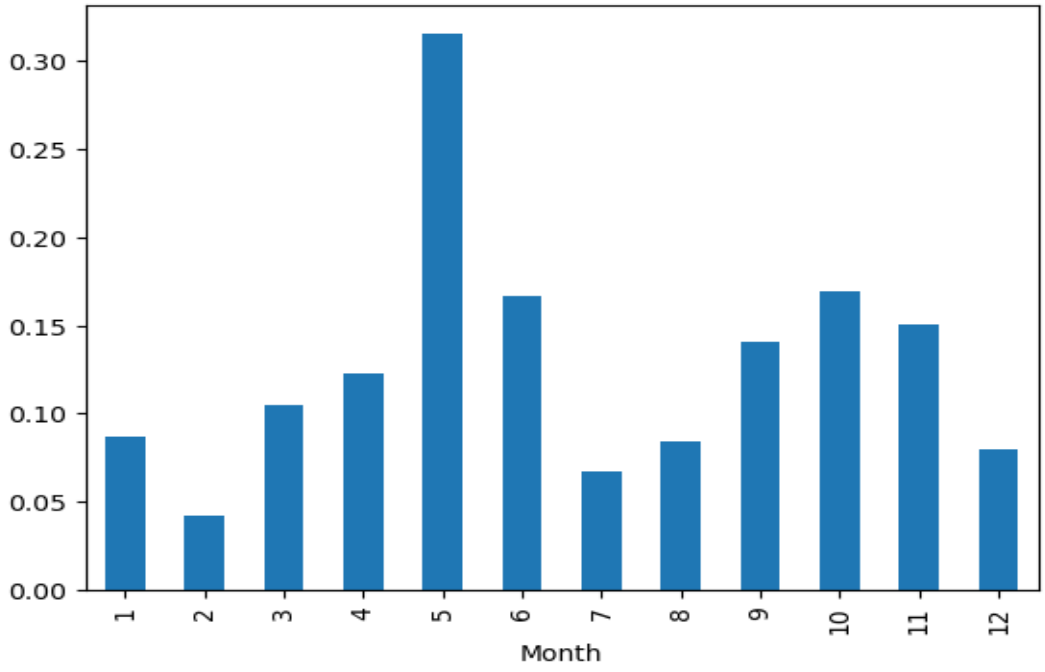

Figure 13 Average precipitation in each month

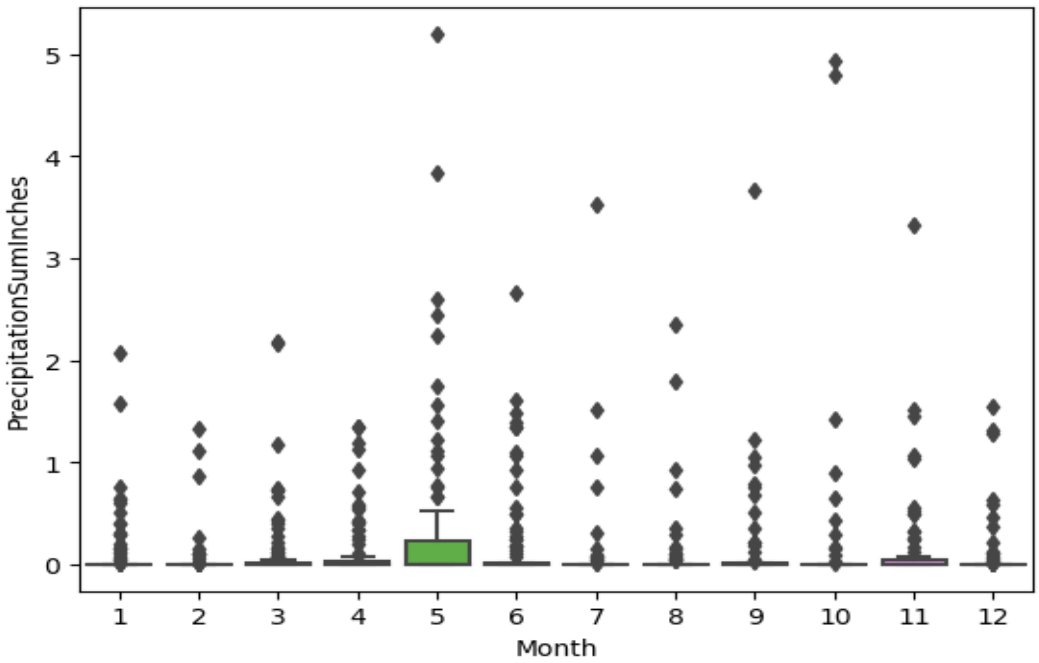

Figure 14 Boxplot of month wise average precipitation

- **<u>Szeged Weather Dataset</u>:**

  In this dataset, first we check if there is any null value or missing value present or not, and we find that there is no missing value or null value present in the dataset. Now, we plot to find the relation between temperature, apparent temperature and humidity which is shown in **Figure 15**.

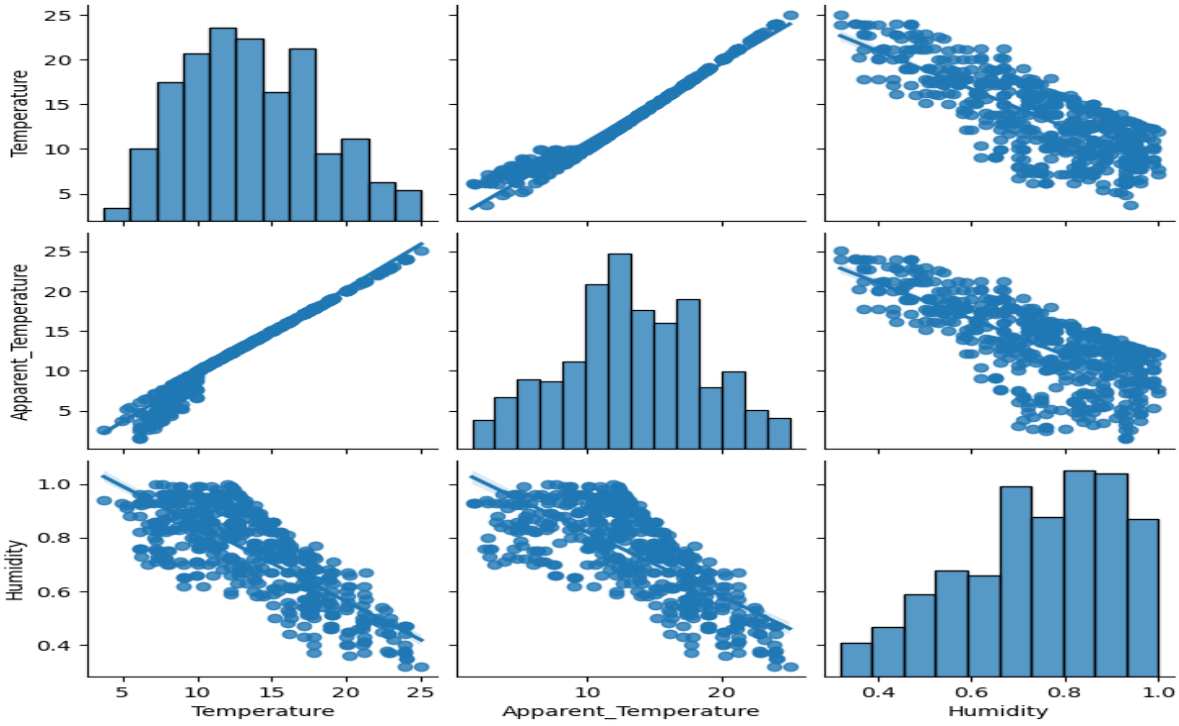

Figure 15 Relation between temperature, apparent temperature and humidity

4.4. Detail Experimental Findings

In this section, we will discuss the results we get after evaluating all models which we mention previously.

- **Austin Weather Dataset:**

  In this dataset, we first evaluate the LR model. For this, we take 'humidity low percent' as independent variable or in x axis and we take 'precipitation' as dependent variable. After evaluating the model, we get these errors which are shown in **Table 4**.

Table 4 Errors of LR

Parameter	Value
MAE	0.18842512568756828
MSE	0.09504560300751855
RMSE	0.30829466911952685

Now we plot the actual vs predicted value which is shown in **Figure 16**, where in x-axis we place humidity values and in y axis we place the precipitation values.

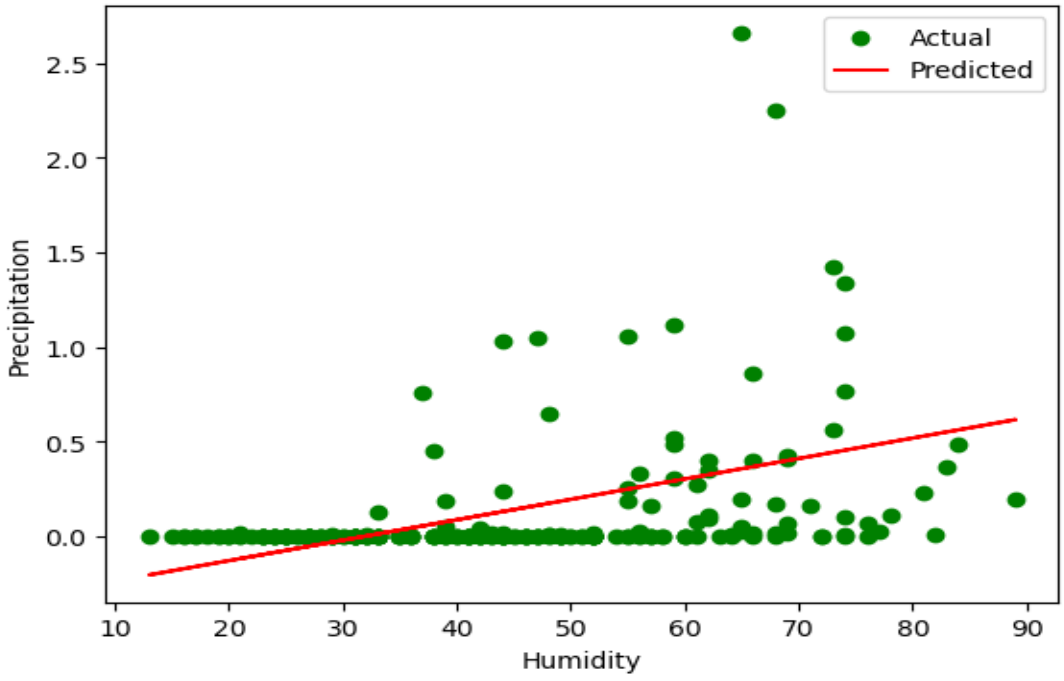

Figure 16 Actual vs Predicted values using LR

Now, we evaluate the MLR model. For this, we take 'humidity low percent' and many others as independent variable or in x axis and we take 'precipitation' as dependent variable. After evaluating the model, we get these errors which are shown in **Table 5**.

Table 5 Errors of MLR

Parameter	Value
MAE	0.1849228023235533
MSE	0.08204053215332008
RMSE	0.2864271847316872

Now we plot the actual vs predicted value which is shown in **Figure 17**, where in x-axis we place actual precipitation values and in y axis we place the predicted values.

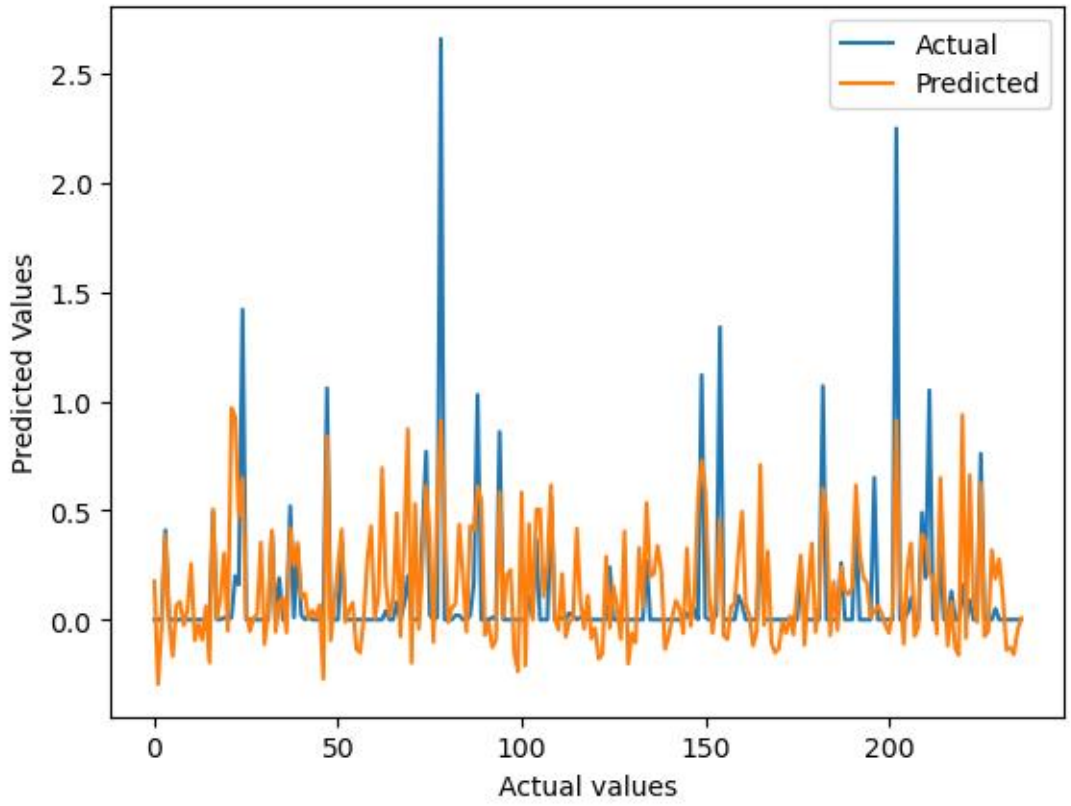

Figure 17 Actual vs Predicted values using MLR

Now, we compare the errors of LR and MLR which is shown in **Table 6**.

Table 6 Comparison between LR and MLR error values

Parameter	LR error	MLR error
MAE	0.18842512568756828	0.1849228023235533
MSE	0.09504560300751855	0.08204053215332008
RMSE	0.3082946691195268 5	0.2864271847316872

And if we compare them graphically, which is shown in **Figure 18**.

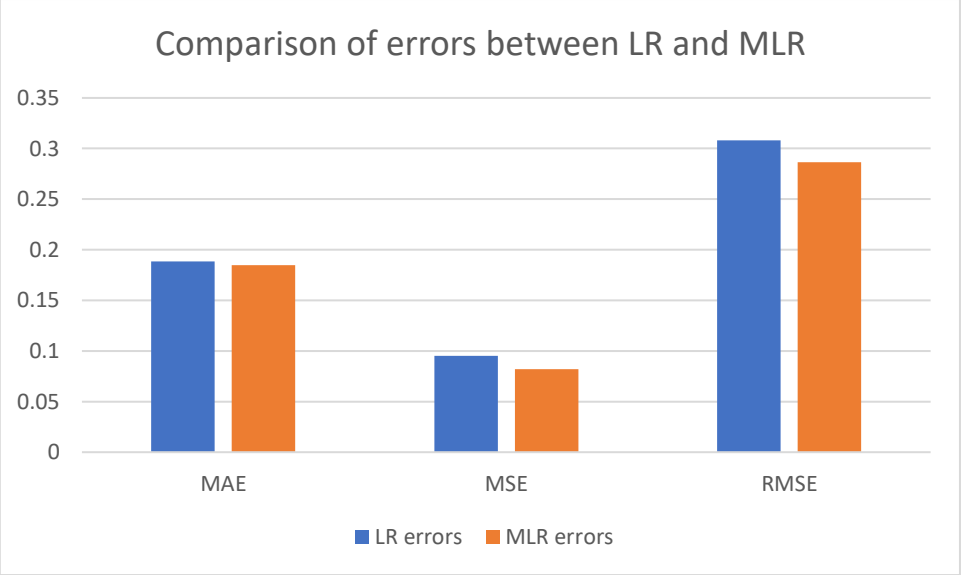

Figure 18 Comparison of errors between LR and MLR

Now, we evaluate the decision tree where we use 'MSE' as criterion and we initialize max depth as 5. After evaluating we get the errors which is shown in **Table 7**.

Table 7 Errors of decision tree when depth is 5

Parameter	Value
MAE	0.26328604054601673
MSE	0.22036683733011853
RMSE	0.46943246301264524

And we plot the actual vs predicted values which is shown in **Figure 19**, where in x-axis we place actual precipitation values and in y axis we place the predicted values.

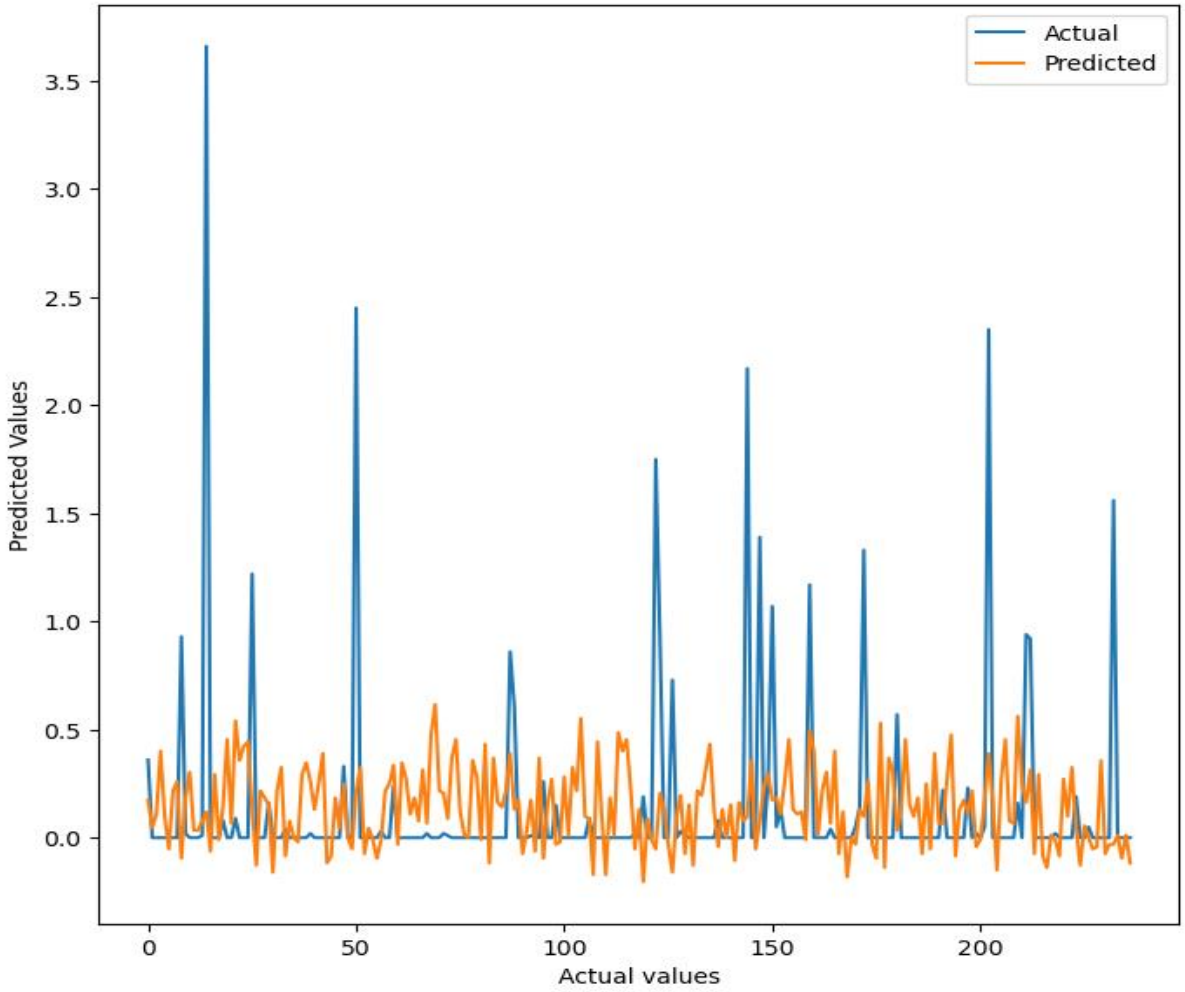

Figure 19 Decision tree when depth is 5

Now we increase the max depth of the decision tree from 5 to 10 and calculate the errors which is shown in **Table 8**.

Table 8 Decision tree errors when depth is 10

Parameter	Value
MAE	0.16850073247571365
MSE	0.35952830481770476
RMSE	0.5996067918375381

Now we plot the actual vs predicted precipitation values, which is shown in **Figure 20,** where in x-axis we place actual precipitation values and in y axis we place the predicted values.

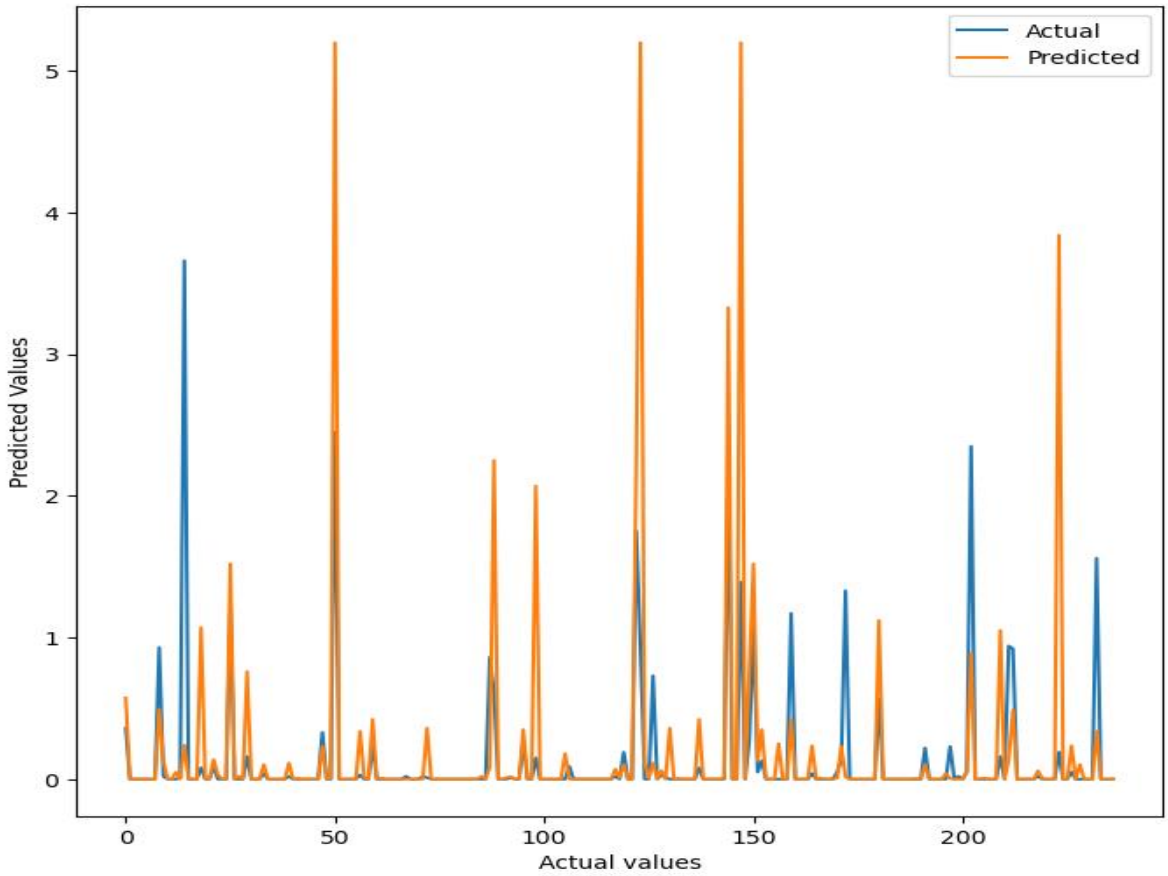

Figure 20 Decision tree results when depth is 10

Now we increase the max depth of the decision tree from 10 to 15 and calculate the errors which is showing in **Table 9**.

Table 9 Decision tree errors when depth is 15

Parameter	Value
MAE	0.16048523206751056
MSE	0.3018798523206751
RMSE	0.5494359401428661

Now we plot actual vs predicted precipitation values which is shown in **Figure 21,** where in x-axis we place actual precipitation values and in y axis we place the predicted values.

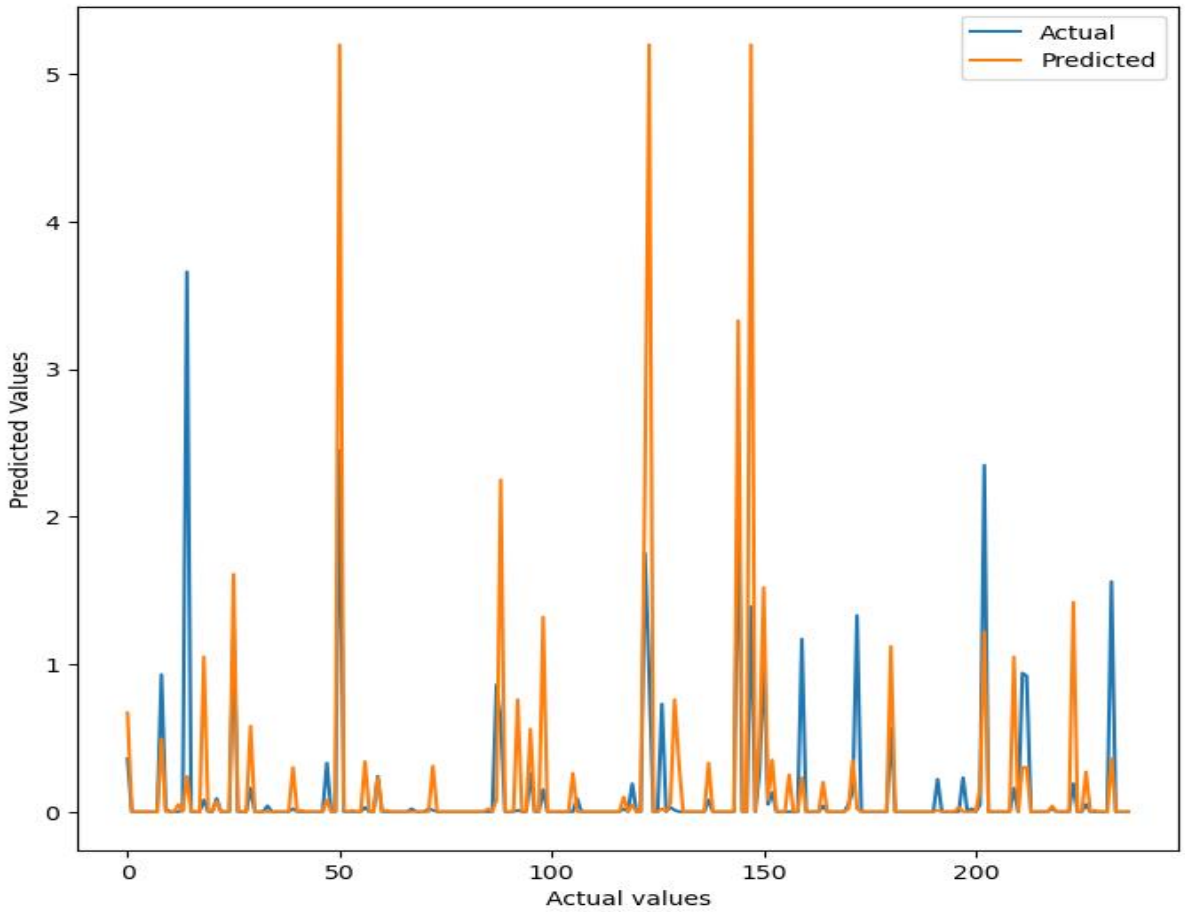

Figure 21 Decision tree when depth is 15

Now, we compare the errors of decision tree which is shown in **Table 10.**

Table 10 Comparison of Decision tree errors

Parameter	Depth 5	Depth 10	Depth 15
MAE	0.26328604054601673	0.16850073247571365	0.16048523206751056
MSE	0.22036683733011853	0.35952830481770476	0.30187985232060751
RMSE	0.46943246301264524	0.59960679183753381	0.54943594014286661

Now, we plot this comparison which is shown in **Figure 22**.

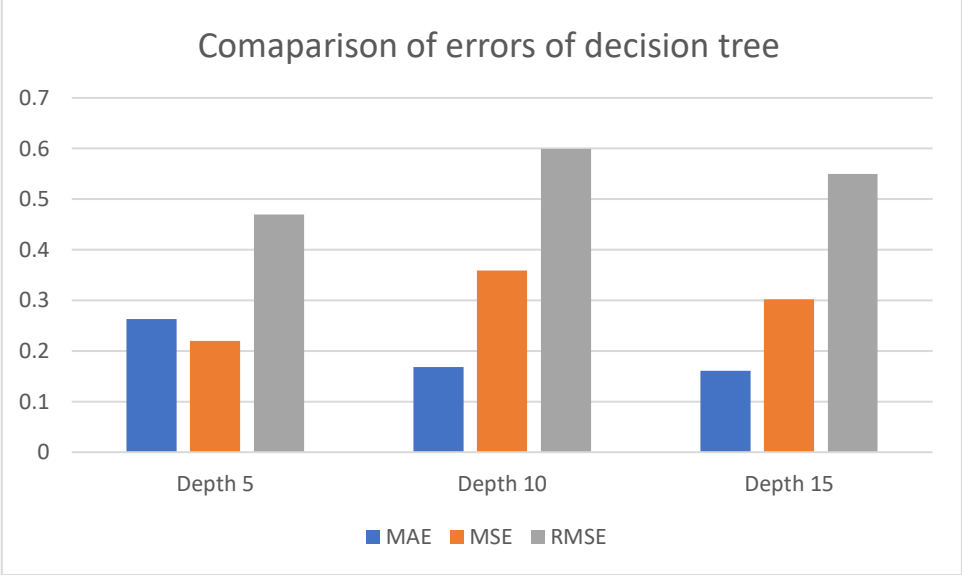

Figure 22 Errors Comparison of decision tree

Now we evaluate the LSTM model. Initially we take 50 epochs and batch size is 30. The errors is shown in **Table 11.**

Table 11 Errors of LSTM when epochs are 50, batch size in 30

Parameter	Value
MAE	0.19799209521163869
MSE	0.1226833595111617
RMSE	0.35026184421252865

Now we plot loss graph which is shown in **Figure 23**, where in x axis we plot the epochs values and in y axis we plot the loss values.

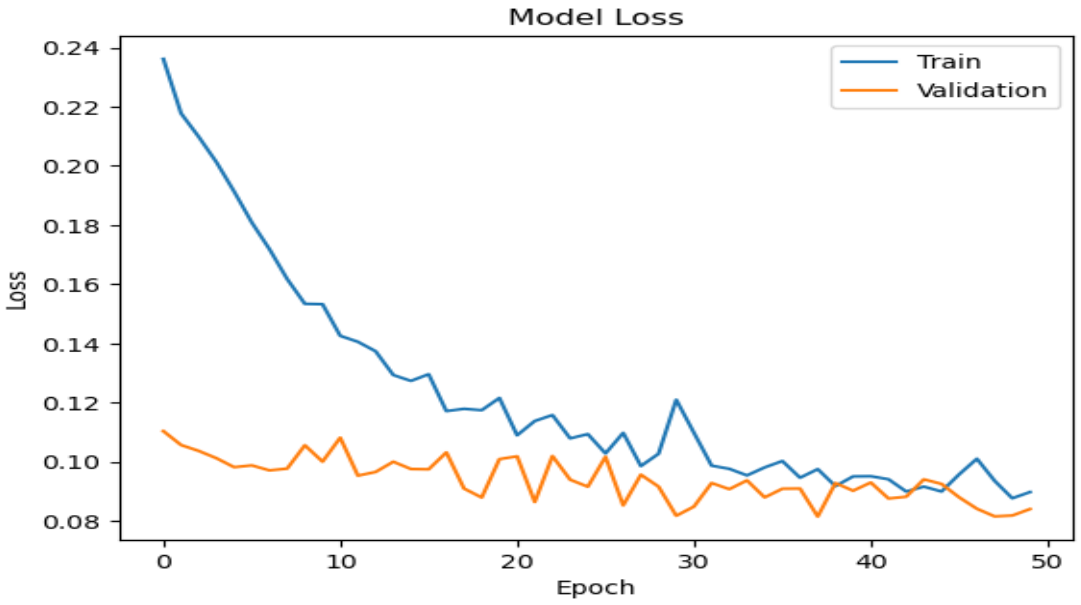

Figure 23 Epochs vs loss graph

Now we increase the batch size from 30 to 60 and we find the errors which is shown in **Table 12.**

Table 12 Errors of LSTM when batch size is 60, epochs are 50

Parameter	Value
MAE	0.0574214839295615
MSE	0.022088155858427407
RMSE	0.1486208459753456

Now we plot the model loss which is shown in **Figure 24**, where in x axis we plot the epochs values and in y axis we plot the loss values.

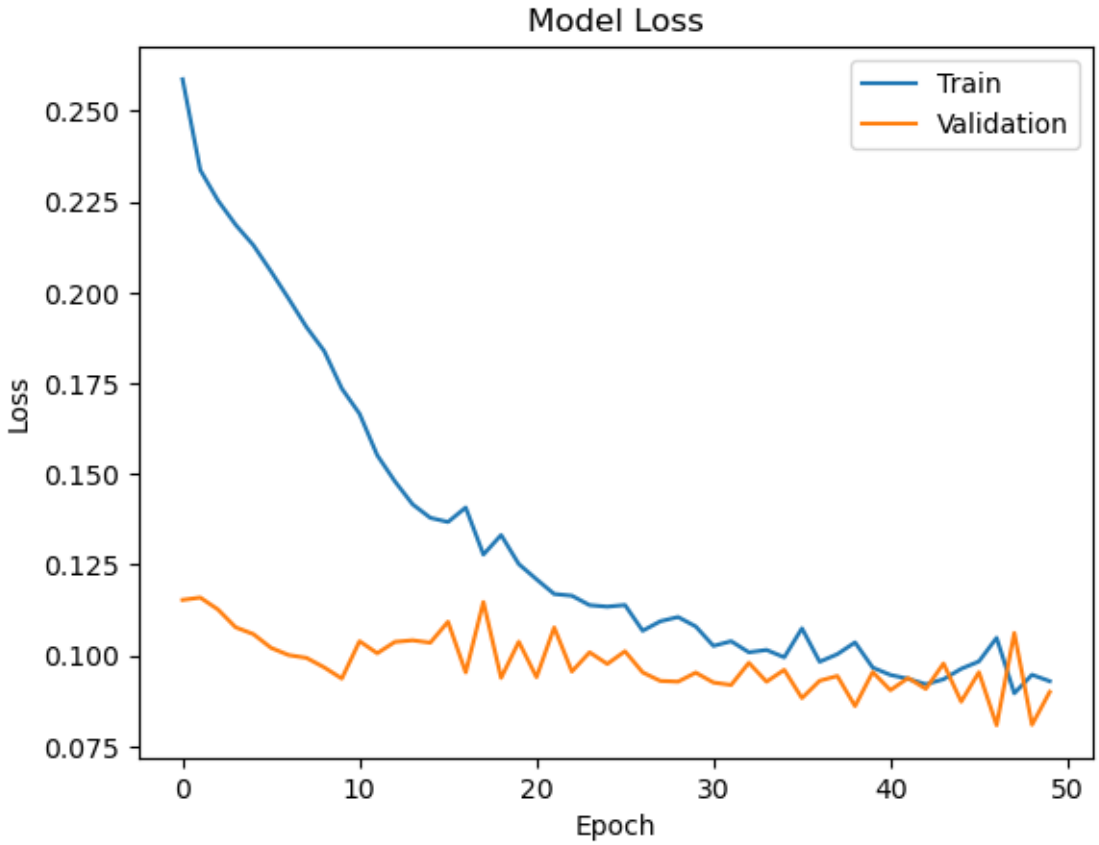

Figure 24 Loss graph of LSTM when batch size is 60, epochs are 50

Now we compare the errors of LSTM models when epochs is 50, but the batch sizes are 30 and 60. The errors are shown in **Table 13**.

Table 13 Comparison of LSTM models when epochs are 50

Parameter	Batch size = 30	Batch Size = 60
MAE	0.19799209521163869	0.0574214839295615
MSE	0.1226833595111617	0.022088155858427407
RMSE	0.35026184421252865	0.1486208459753456

We also compare the errors graphically, which is shown in **Figure 25**.

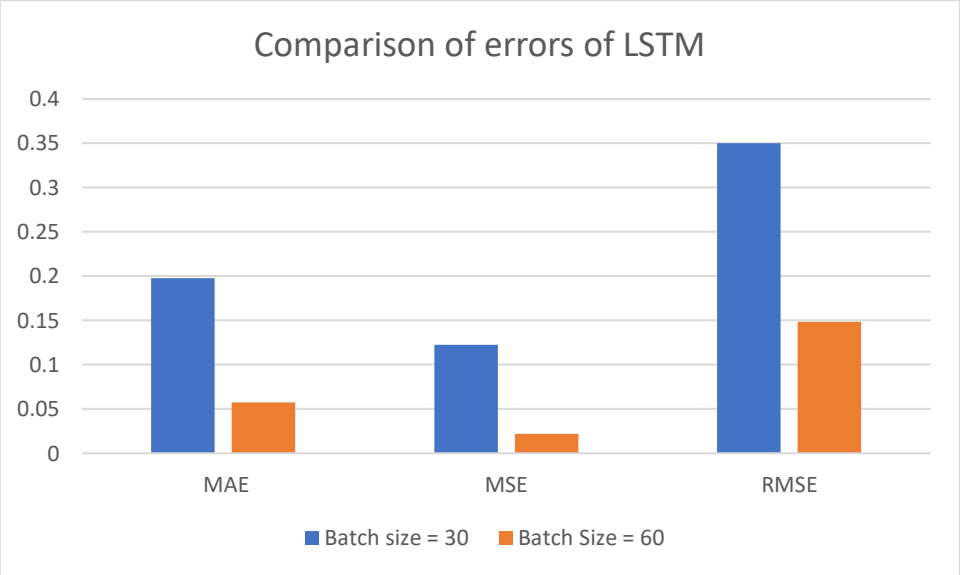

Figure 25 Comparison of LSTM error when epochs are 50

Now, we increase the epochs from 50 to 100.We initialize the batch size to 30. After evaluating, we get the errors which is shown in **Table 14.**

Table 14 LSTM errors when epochs are 100, batch size is 30

Parameter	Value
MAE	0.09982585260489786
MSE	0.03109546401366934
RMSE	0.1763390598071492

Now we plot the loss graph of this model which is shown in **Figure 26**, where in x axis we plot the epochs values and in y axis we plot the loss values.

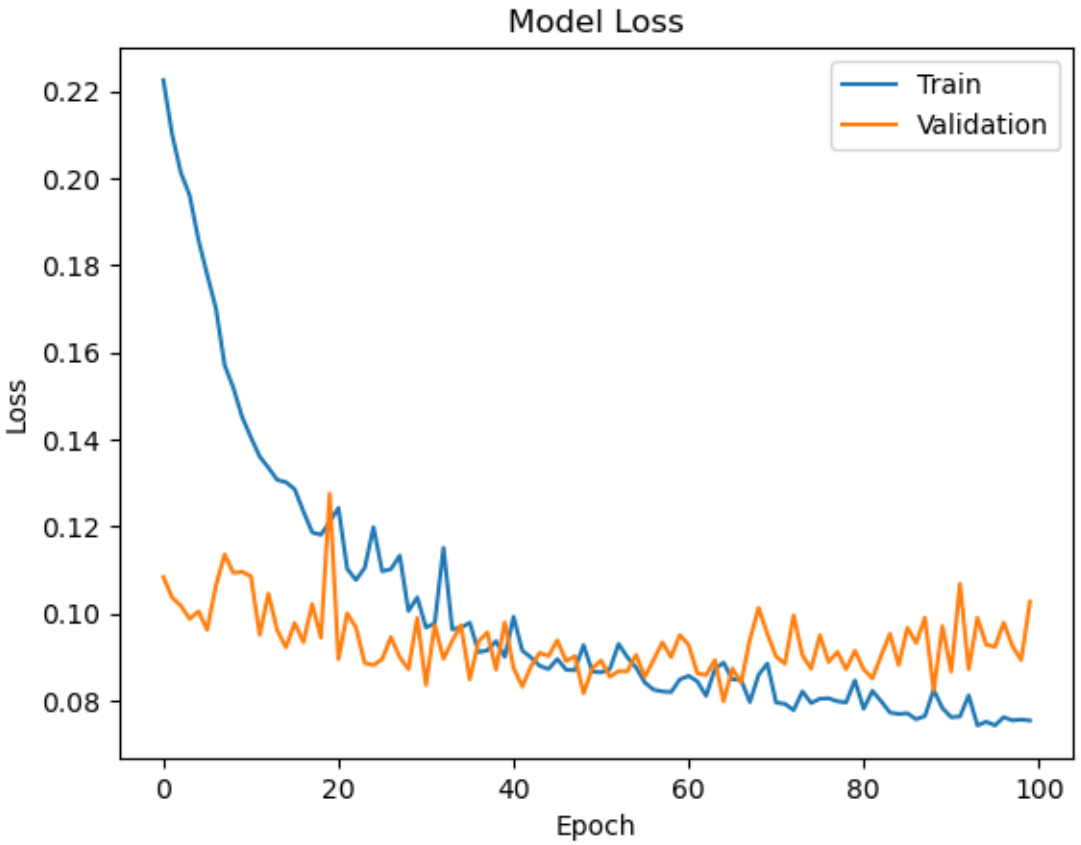

Figure 26 Loss graph when epochs are 100, batch size is 30

Now, we increase the batch size from 30 to 60 and after evaluating we get the errors which is shown in **Table 15**.

Table 15 LSTM errors when epochs are 100, batch size is 60

Parameter	Value
MAE	0.11261714556109262
MSE	0.03290016500761175
RMSE	0.18138402632980596

Now we plot the loss graph which is shown in **Figure 27**, where in x axis we plot the epochs values and in y axis we plot the loss values.

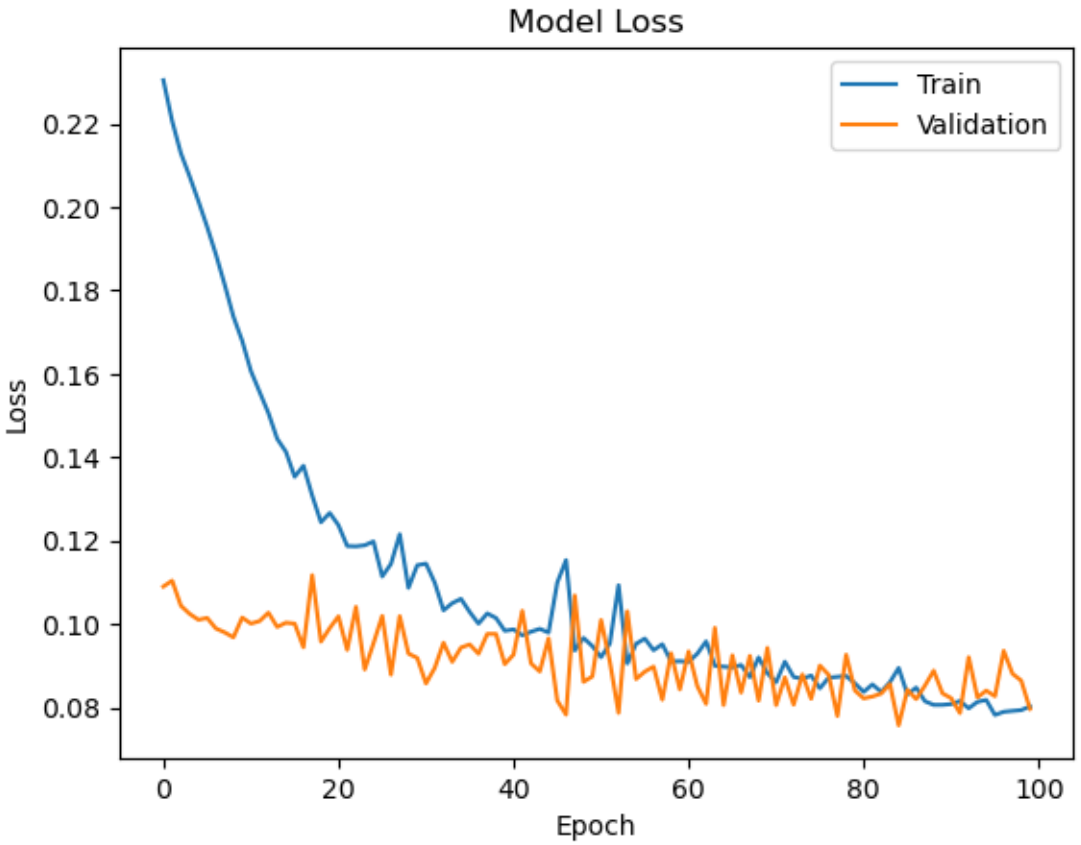

Figure 27 Loss graph when epochs are 100, batch size is 60

Now we compare the errors of LSTM models when epochs is 100, but the batch sizes are 30 and 60. The errors are shown in **Table 16**.

Table 16 Comparison of LSTM models when epochs are 100

Parameter	Batch size = 30	Batch Size = 60
MAE	0.09982585260489786	0.11261714556109262
MSE	0.03109546401366934	0.03290016500761175
RMSE	0.1763390598071492	0.18138402632980596

We also compare the errors graphically, which is shown in **Figure 28**.

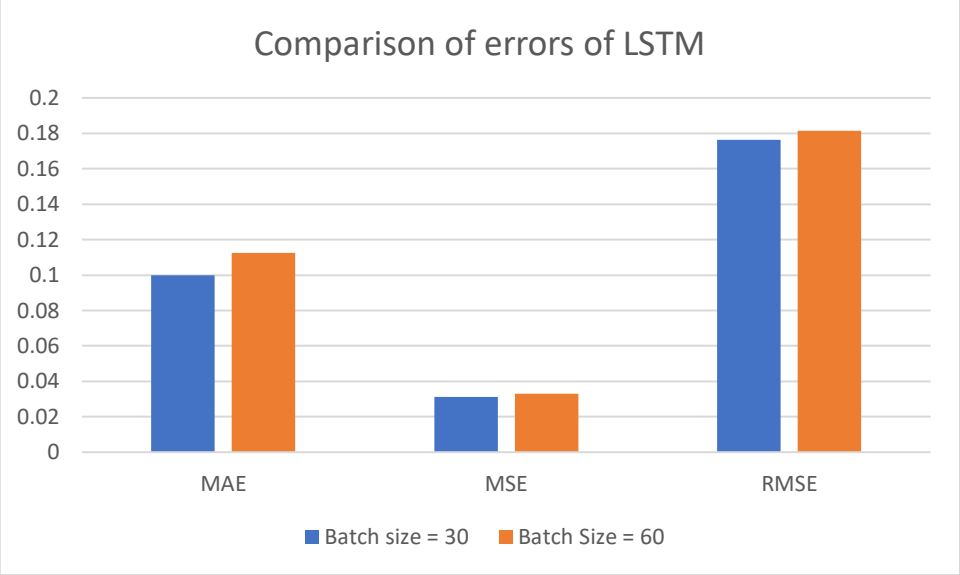

Figure 28 Comparison of LSTM errors when epochs are 100

- **Szeged Weather Dataset:**

  In this dataset, we first evaluate the LR model. For this we take 'Apparent Temperature' as independent variable or in x axis and we take 'humidity' as dependent variable or in y axis. After evaluating the model these are the errors what we get which is shown in **Table 17**.

Table 17 Errors of LR

Parameter	Value
MAE	0.08798530099071504
MSE	0.011213097936505228
RMSE	0.10589191629442367

And now we plot actual vs predicted values which is shown in **Figure 29**, where in x axis we plot Apparent temperature and in y axis we plot humidity values.

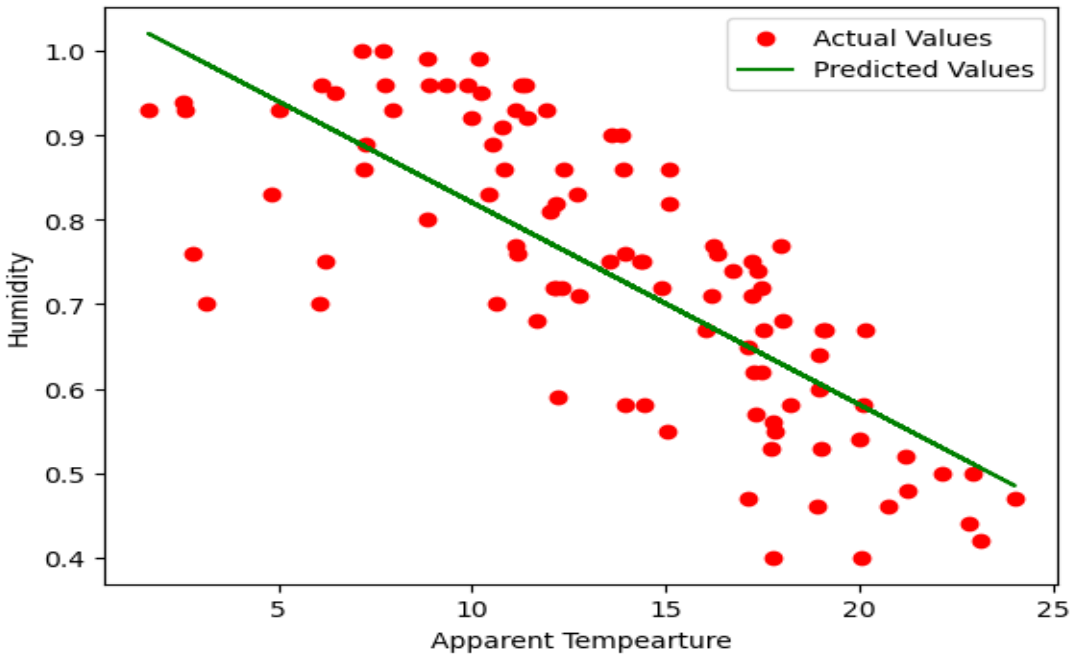

Figure 29 Actual vs predicted values using LR

Next, we evaluate the MLR model where we take 'Apparent temperature' and 'temperature' in independent variable or in x axis and we take 'Humidity' in dependent variable or in y axis. After evaluating the model these are the errors what we get which is shown in **Table 18**.

Table 18 Errors of MLR

Parameter	Value
MAE	0.07555854326917352
MSE	0.008387437418538262
RMSE	0.09158295375526093

Now we plot actual vs predicted values which is shown in **Figure 30**, where in x axis we plot Actual humidity values and in y axis we plot predicted humidity values.

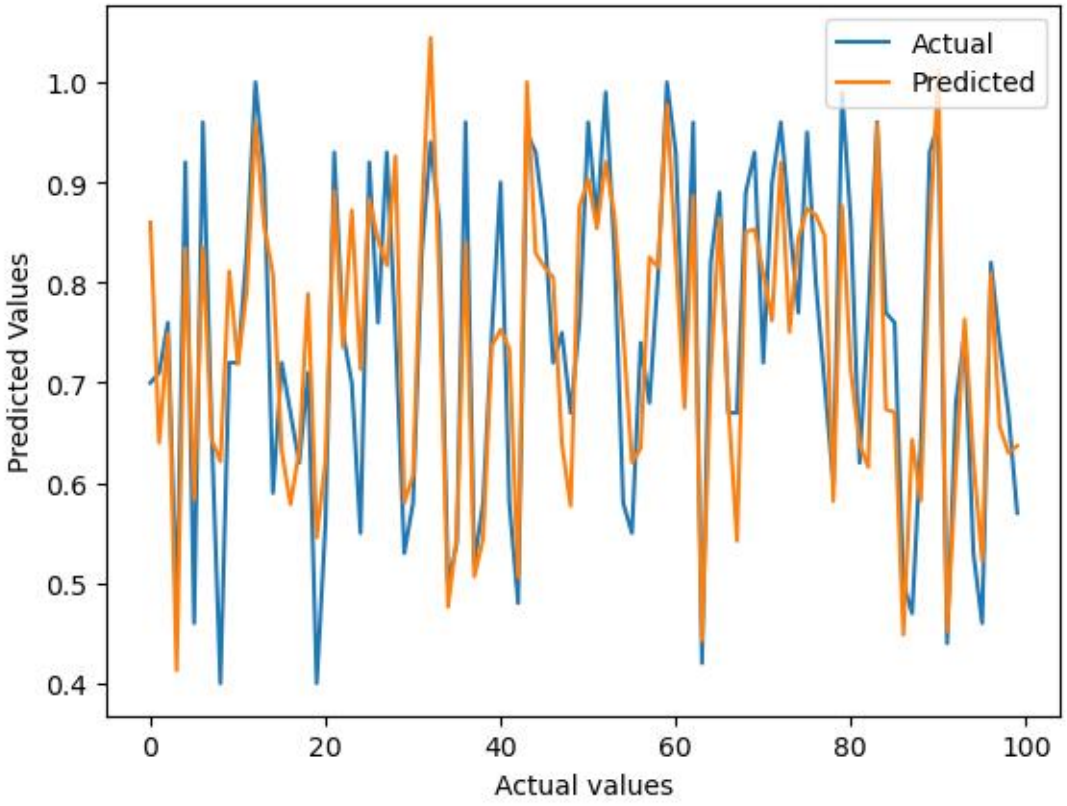

Figure 30 Actual vs predicted values using MLR

Now we compare the errors of LR and MLR models which is shown in **Table 19.**

Table 19 Comparison of errors between LR and MLR

Parameter	LR errors	MLR errors
MAE	0.08798530099071504	0.07555854326917352
MSE	0.011213097936505228	0.008387437418538262
RMSE	0.10589191629442367	0.09158295375526093

And we also plot these errors in graphically which is shown in **Figure 31.**

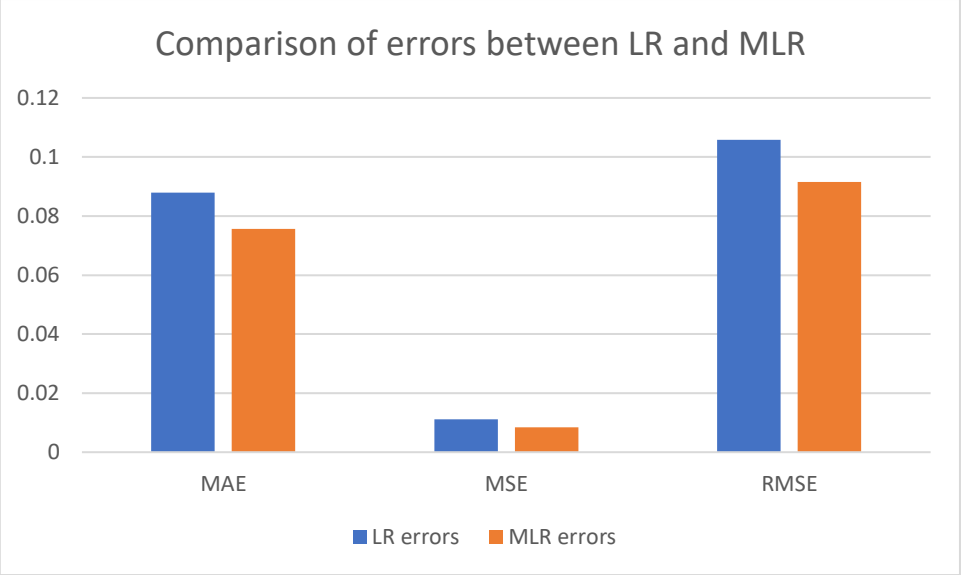

Figure 31 Comparison of errors between LR and MLR

Now, we evaluate the decision tree model where we use 'MSE' as criterion and we initialize max depth as 5. After evaluating we get the errors which is shown in **Table 20.**

Table 20 Errors of decision tree where depth is 5

Parameter	Value
MAE	0.08732048183829332
MSE	0.010958921369993968
RMSE	0.10468486695790355

Now, we plot the actual vs predicted values which is shown in **Figure 32**, where in x axis we plot the actual values and in y axis we plot the predicted values.

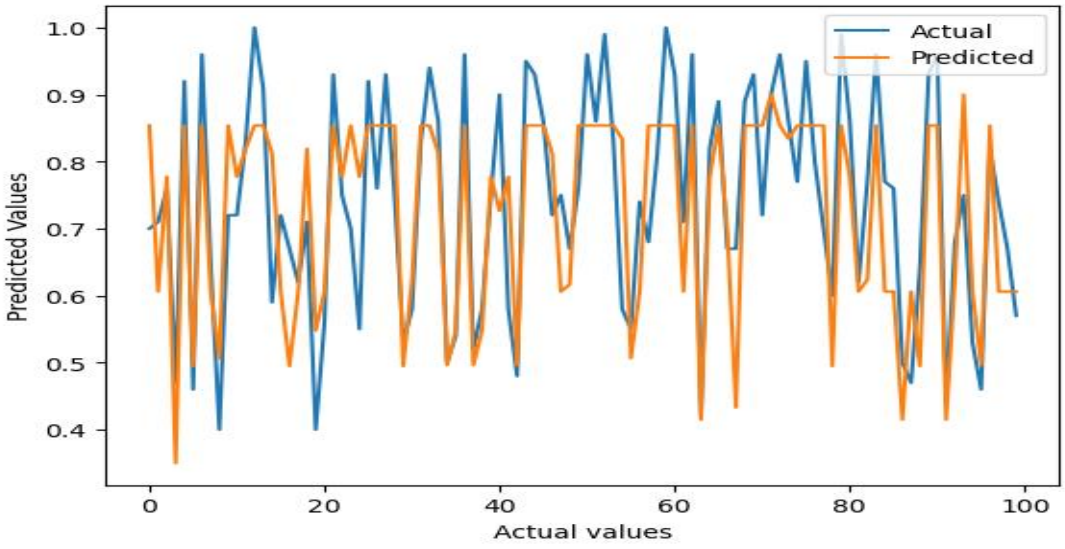

Figure 32 Actual vs predicted values of decision tree when depth is 5

Now we increase the max depth of the decision tree from 5 to 10 and calculate the errors which is shown in **Table 21**.

Table 21 Decision tree errors when depth is 10

Parameter	Value
MAE	0.08677373902413375
MSE	0.012435721921410401
RMSE	0.1115155680674694

Now, we plot the actual vs predicted values which is shown in **Figure 33,** where in x axis we plot the actual values and in y axis we plot the predicted values.

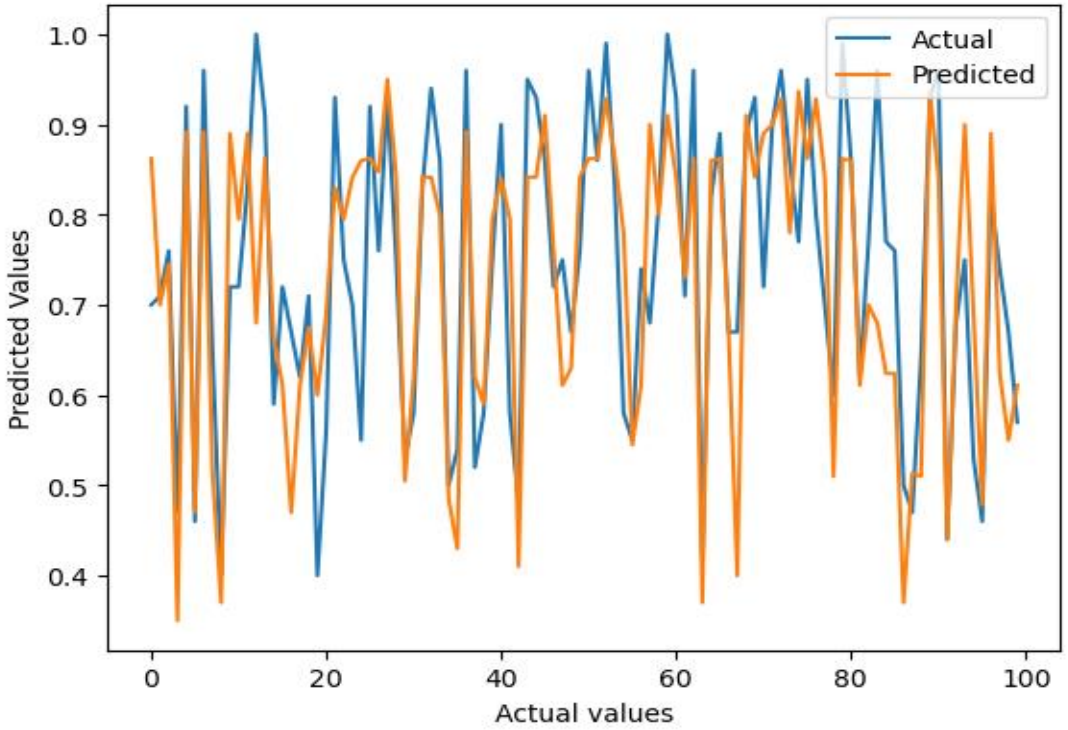

Figure 33 Decision tree results where depth is 10

Now we increase the max depth of the decision tree from 10 to 15 and calculate the errors which is shown in **Table 22**.

Table 22 Decision Tree errors where depth is 15

Parameter	Value
MAE	0.08957666666666668
MSE	0.014417645605963519
RMSE	0.12007350084828675

Now, we plot the actual vs predicted values which is shown in **Figure 34,** where in x axis we plot the actual values and in y axis we plot the predicted values.

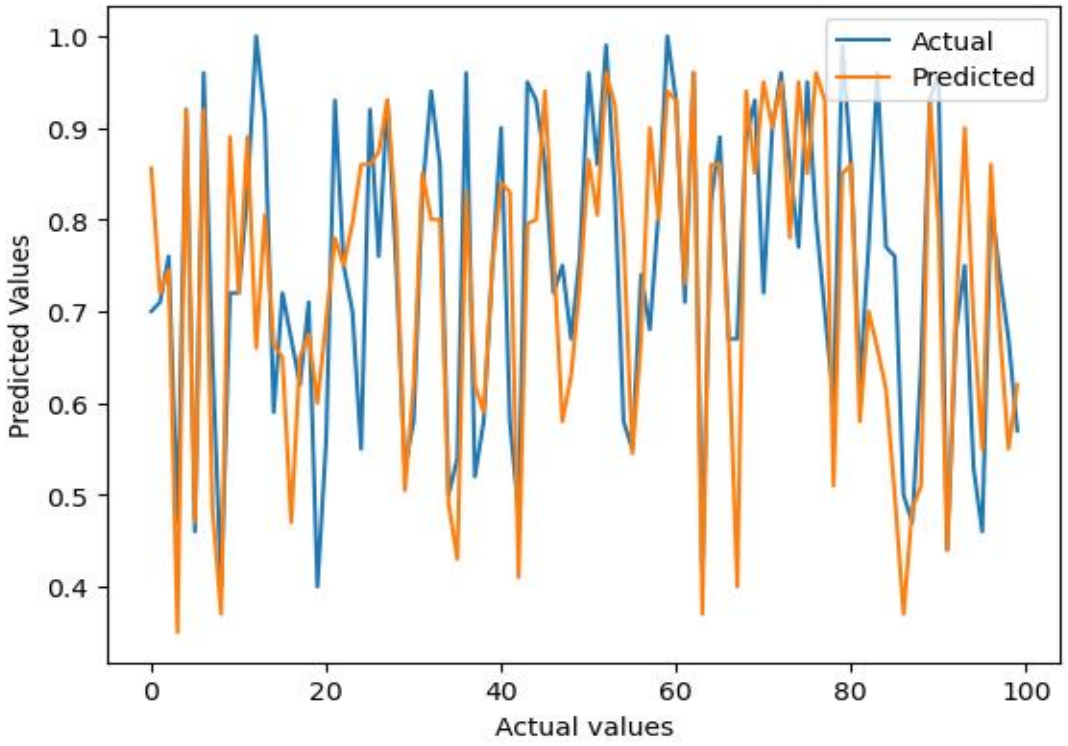

Figure 34 Decision tree results where depth is 15

Now, we compare the errors of decision tree which is shown in **Table 23**.

Table 23 Comparison of decision tree errors

Parameter	Depth 5	Depth 10	Depth 15
MAE	0.0873204818382 9332	0.0867737390241 3375	0.0895766666666 6668
MSE	0.0109589213699 93968	0.0124357219214 10401	0.0144176456059 63519
RMSE	0.1046848669579 0355	0.1115155680674 694	0.1200735008482 8675

Now, we plot this comparison which is shown in **Figure 35.**

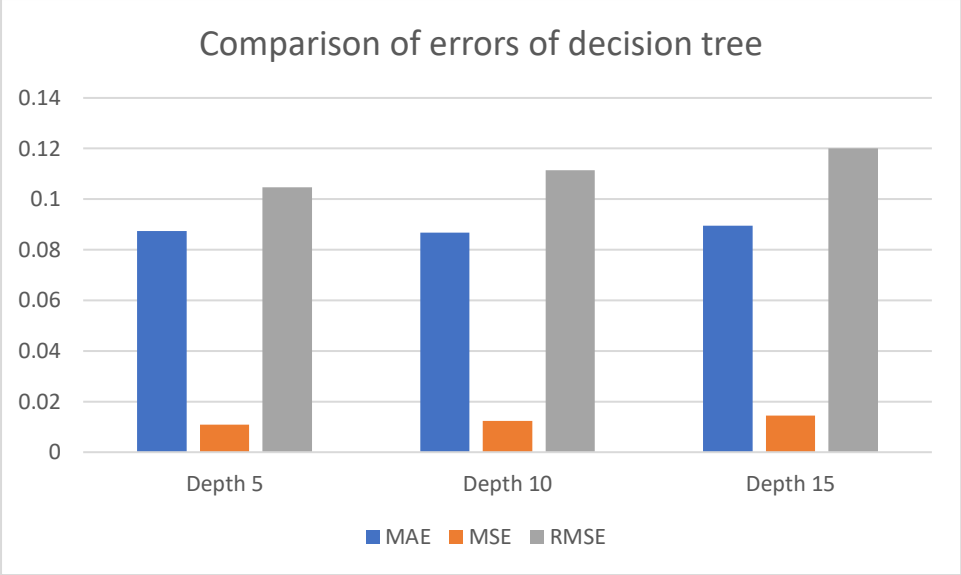

Figure 35 Comparison of errors of decision tree

Now we evaluate the LSTM model. Initially we take 50 epochs and batch size is 30. After evaluating, errors are shown in **Table 24**.

Table 24 Errors of LSTM when epochs are 50 and batch size are 30

Parameter	Value
MAE	0.08373510398864745
MSE	0.009729148637090063
RMSE	0.09863644679878764

Now we plot the loss graph of this model which is shown in **Figure 36**, where in x axis we plot the epochs values and in y axis we plot the loss values.

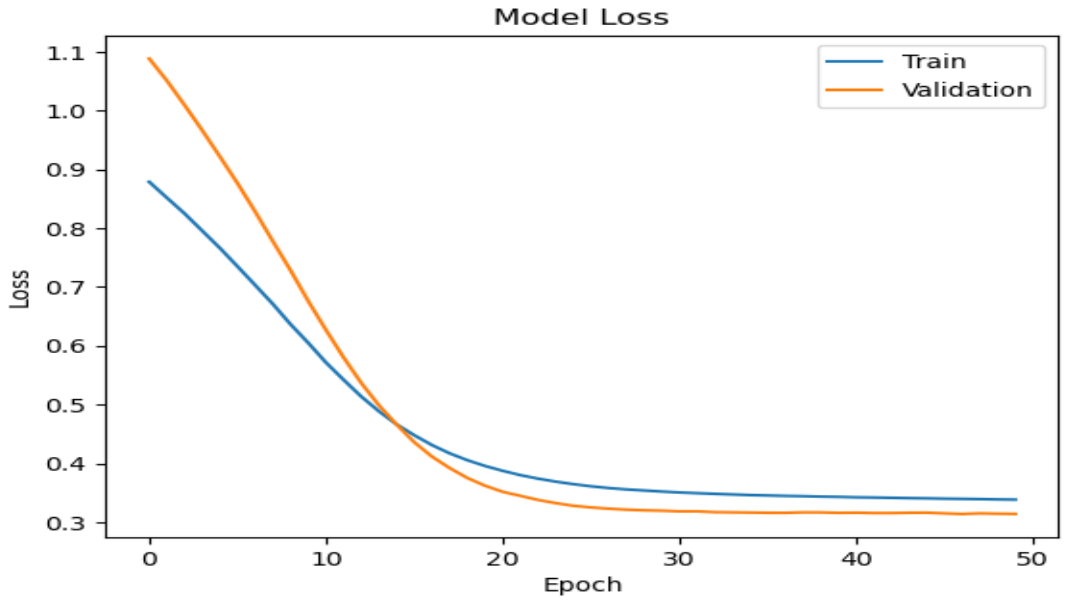

Figure 36 Loss graph of LSTM when epochs are 50 and Batch size is 30

Next, we increase the batch size from 30 to 60 and calculate the errors, which is shown in **Table 25.**

Table 25 Errors of LSTM when epochs are 50 and batch size is 60

Parameter	Value
MAE	0.14425150604117368
MSE	0.03244506834060803
RMSE	0.1801251463305666

Now we plot the loss graph of this model which is shown in **Figure 37,** where in x axis we plot the epochs values and in y axis we plot the loss values

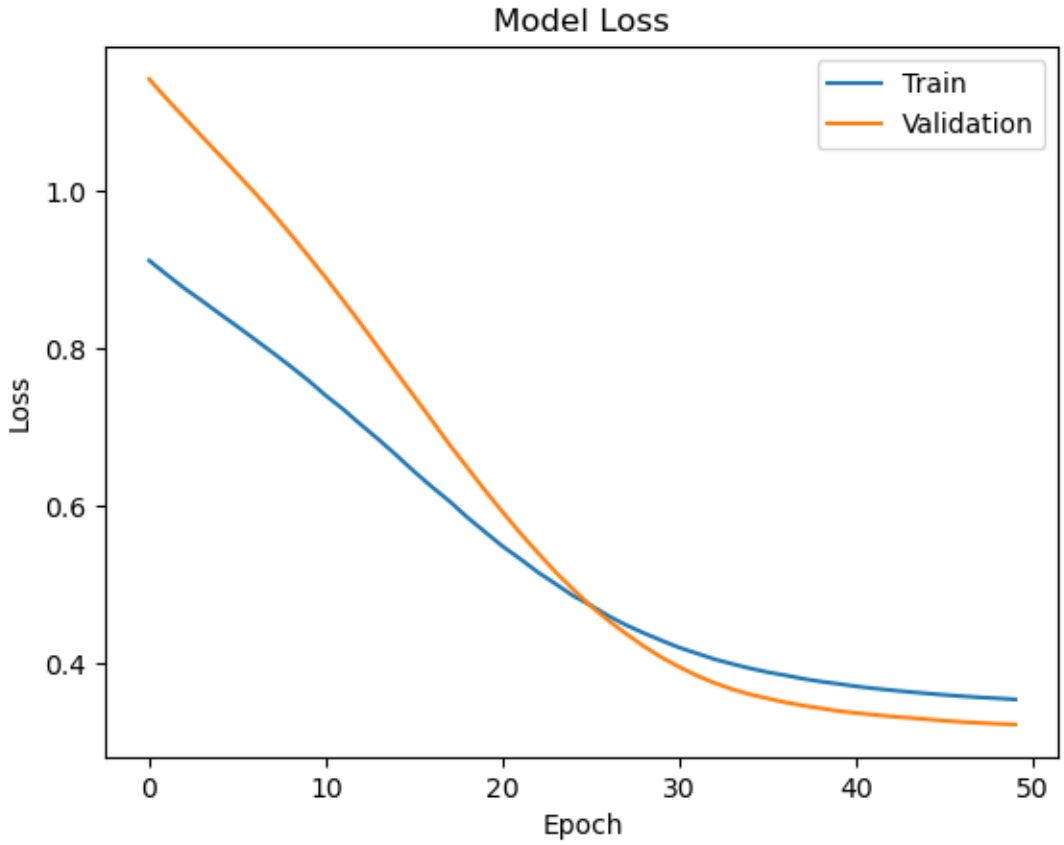

Figure 37 Loss graph of LSTM when epochs are 50 and batch size is 60

Now we compare the errors of LSTM models when epochs is 50, but the batch sizes are 30 and 60. The result is shown in **Table 26**.

Table 26 Comparison of LSTM models when epochs are 50

Parameter	Batch size = 30	Batch Size = 60
MAE	0.08373510398864745	0.14425150604117368
MSE	0.009729148637090063	0.03244506834060803
RMSE	0.09863644679878764	0.1801251463305666

And we also plot the errors in graphically which is shown in **Figure 38**

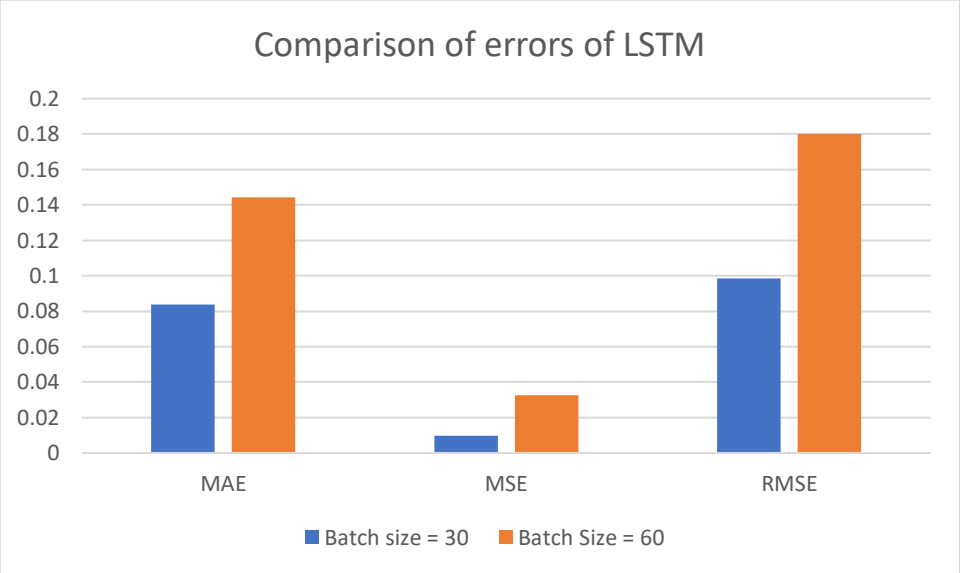

Figure 38 Comparison of LSTM errors when epochs are 50

Now, we increase the epochs from 50 to 100.We initialize the batch size to 30. After evaluating, we calculate the errors which is shown in **Table 27**.

Table 27 Errors of LSTM when epochs are 100 and batch size is 30

Parameter	Value
MAE	0.1637308046048152
MSE	0.04513037894842329
RMSE	0.21243911821607453

Now we plot the loss graph of this model which is shown in **Figure 39,** where in x axis we plot the epochs values and in y axis we plot the loss values.

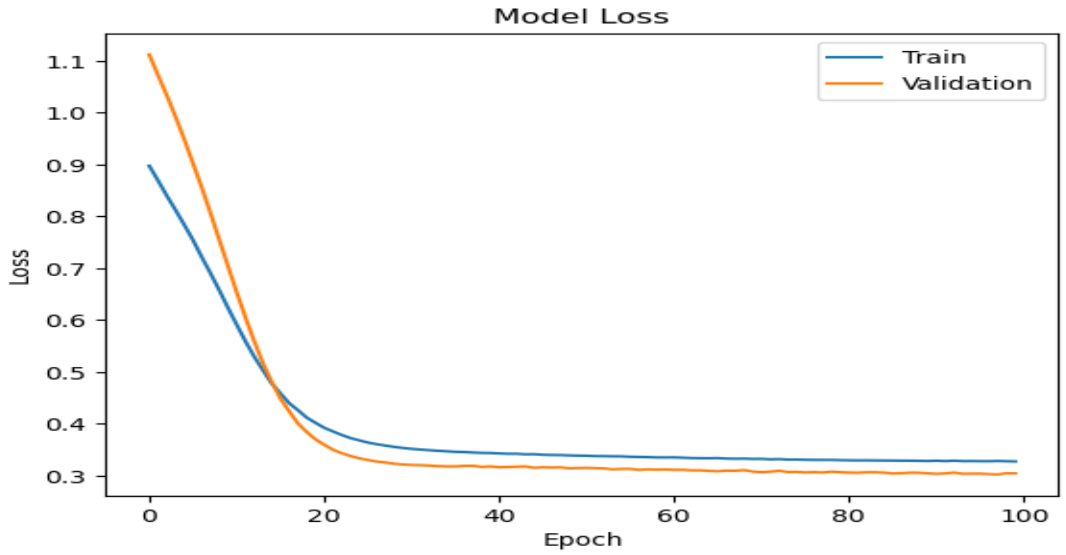

Figure 39 Loss graph of LSTM when epochs are 100 batch size is 30

Next, we increase the batch size from 30 to 60 and calculate the errors, which is shown in **Table 28**.

Table 28 Errors of LSTM when batch size is 60 and epochs are 100

Parameter	Value
MAE	0.16572093225241447
MSE	0.04550590559599581
RMSE	0.21332113255839377

Now we plot the loss graph of this model which is shown in **Figure 40,** where in x axis we plot the epochs values and in y axis we plot the loss values.

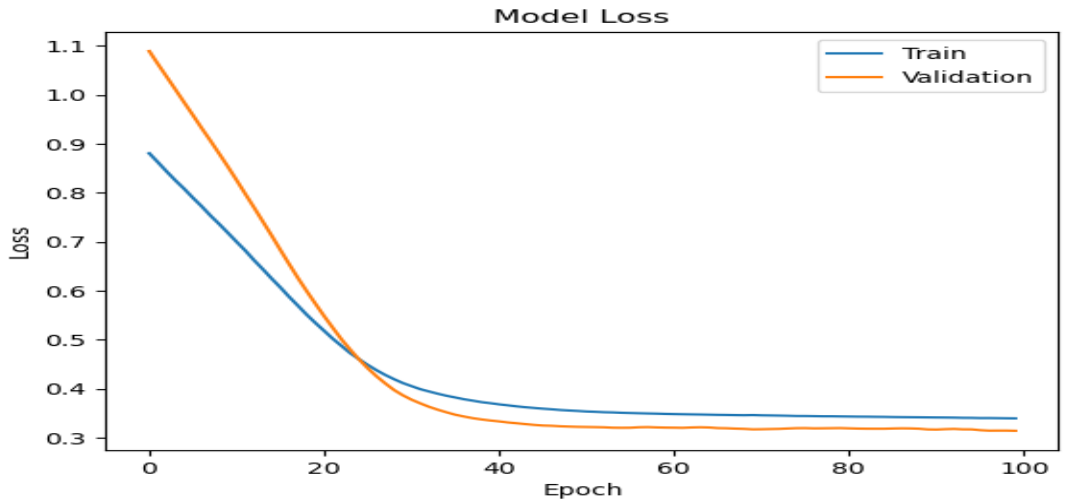

Figure 40 Loss graph of LSTM when epochs are 100 and batch size is 60

Now we compare the errors of LSTM models when epochs is 100, but the batch sizes are 30 and 60. The result is shown in **Table 29**.

Table 29 Comparison of LSTM models when epochs are 100

Parameter	Batch size = 30	Batch Size = 60
MAE	0.1637308046048152	0.16572093225241447
MSE	0.04513037894842329	0.04550590559599581
RMSE	0.21243911821607453	0.21332113255839377

And we also plot the errors in graphically which is shown in **Figure 41**.

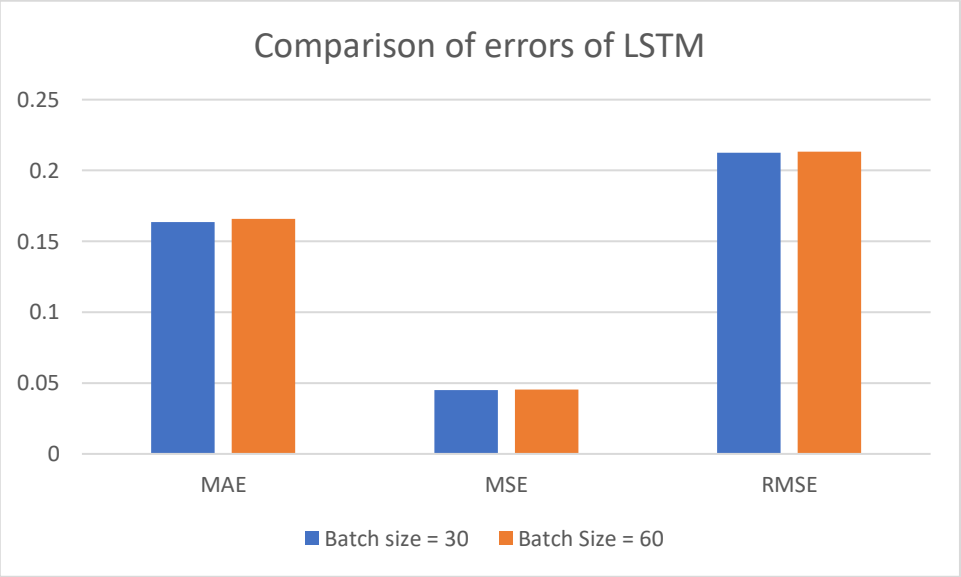

Figure 41 Comparison of LSTM errors when epochs are 100

- ## AQI Dataset:

In this dataset, we first evaluate the LR model. For this we take 'spi' as independent variable or in x axis and we take 'AQI' as dependent variable or in y axis. After evaluating the model these are the errors what we get which is shown in **Table 30.**

Table 30 Errors of LR

Parameter	Value
MAE	15.174897128477522
MSE	1175.412179930889
RMSE	34.28428473704663

And now we plot actual vs predicted values which is shown **Figure 42**, where in x axis we plot spi and in y axis we plot AQI values.

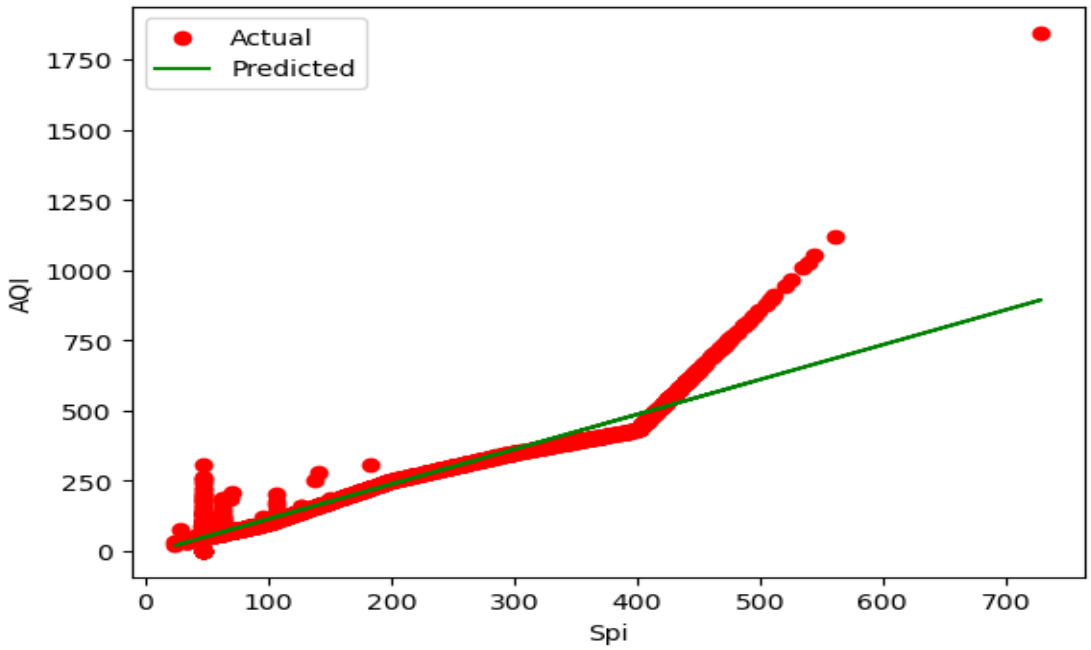

Figure 42 Actual vs predicted values of LR

Next, we evaluate the multiple linear regression (MLR) model where we take spi, noi, soi, etc. in independent variable or in x axis and we take 'AQI' in dependent variable or in y axis. After evaluating the model these are the errors what we get which is shown in **Table 31**.

Table 31 Errors of MLR

Parameter	Value
MAE	6.050586015859769
MSE	140.75750072201072
RMSE	11.864126631236314

Now we plot actual vs predicted values which is shown in **Figure 43**, where in x axis we plot Actual AQI values and in y axis we plot predicted AQI values.

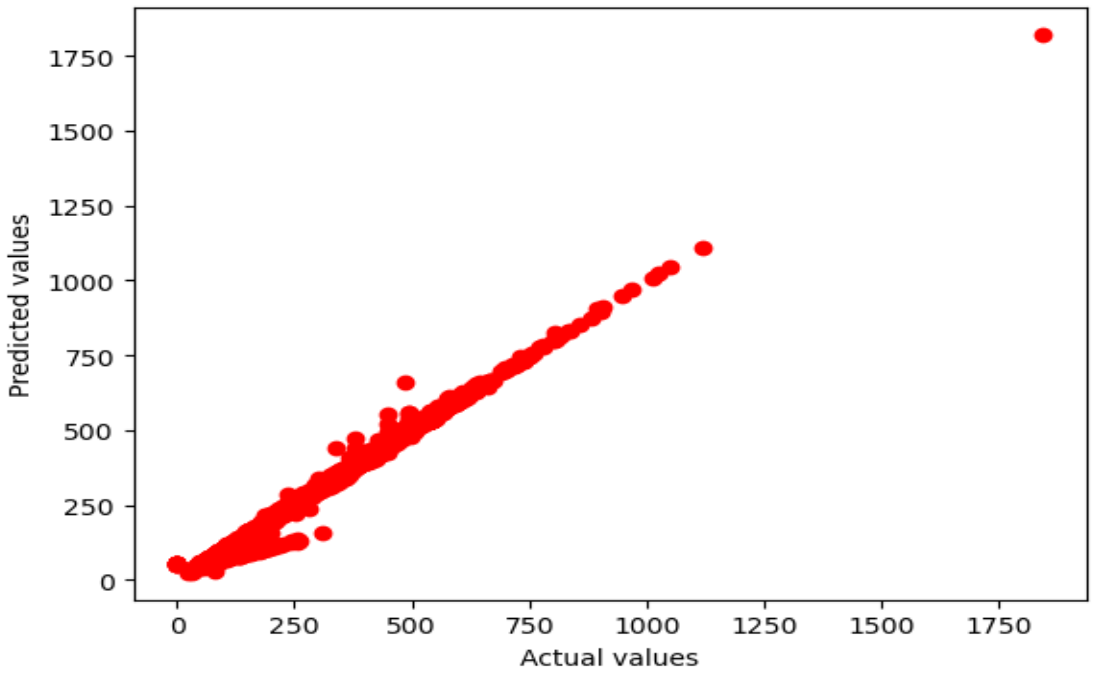

Figure 43 Actual vs predicted graph of MLR

Now we compare the errors of LR and MLR models which is shown in **Table 32.**

Table 32 Comparison of errors between LR and MLR

Parameter	LR errors	MLR errors
MAE	15.174897128477522	6.050586015859769
MSE	1175.412179930889	140.75750072201072
RMSE	34.28428473704663	11.864126631236314

And we also plot the errors in graphically which is shown in **Figure 44.**

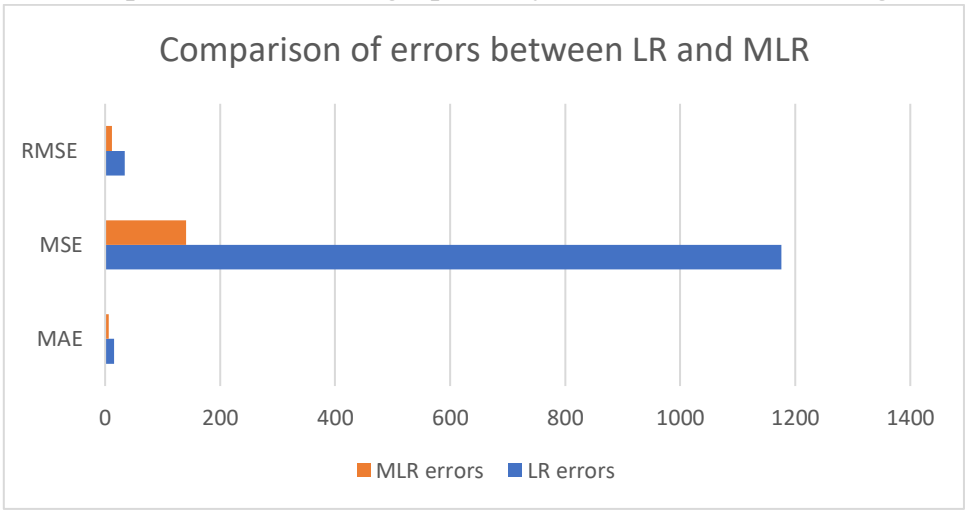

Figure 44 Comparison of errors between LR and MLR

Now, we evaluate the decision tree model where we use 'MSE' as criterion and we initialize max depth as 5. After evaluating we get these errors which is shown in **Table 33**.

Table 33 Errors of decision tree when depth is 5

Parameter	Value
MAE	6.308411673872369
MSE	282.9707467507989
RMSE	16.82173435620712

Now, we plot the actual vs predicted values which is shown in **Figure 45** where in x axis we plot the actual values and in y axis we plot the predicted values.

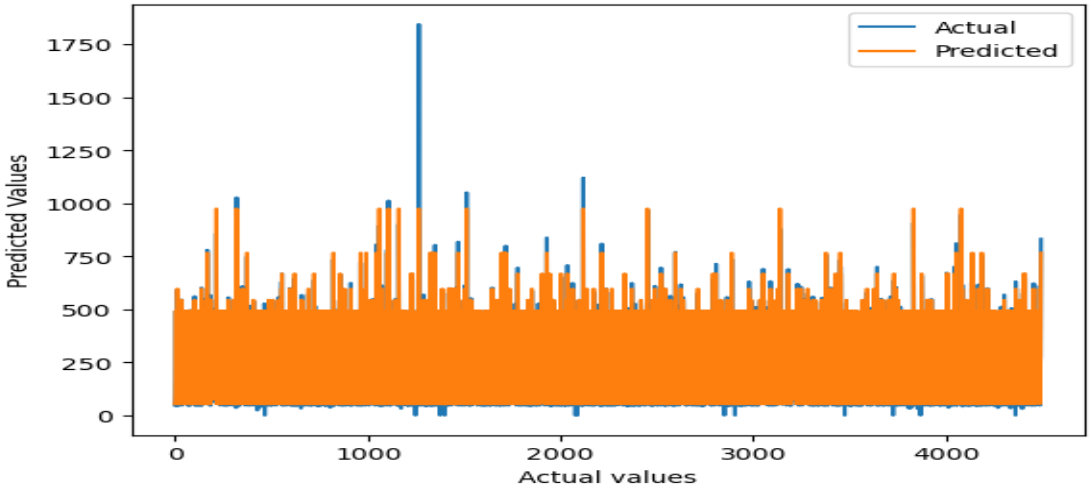

Figure 45 Actual vs predicted graph of decision tree when depth is 5

Now we increase the max depth of the decision tree from 5 to 10 and calculate the errors which is shown in **Table 34**.

Table 34 Errors of decision tree when depth is 10

Parameter	Value
MAE	0.46407865056296216
MSE	64.28417925436828
RMSE	8.01774153077837

Now, we plot the actual vs predicted values which is shown in **Figure 46**, where in x axis we plot the actual values and in y axis we plot the predicted values.

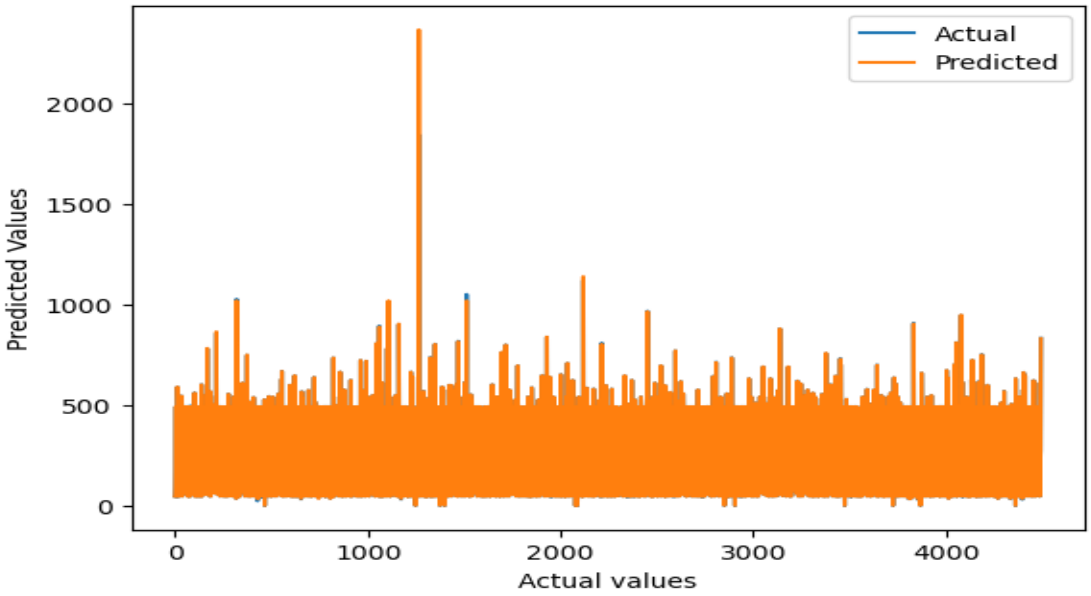

Figure 46 Actual vs predicted values of decision tree when depth is 10

Now we increase the max depth of the decision tree from 10 to 15 and calculate the errors which is shown in **Table 35**.

Table 35 Errors of decision tree when depth is 15

Parameter	Value
MAE	0.25740557686322485
MSE	64.21864192184026
RMSE	8.013653469038966

Now, we plot the actual vs predicted values which is shown in **Figure 47**, where in x axis we plot the actual values and in y axis we plot the predicted values.

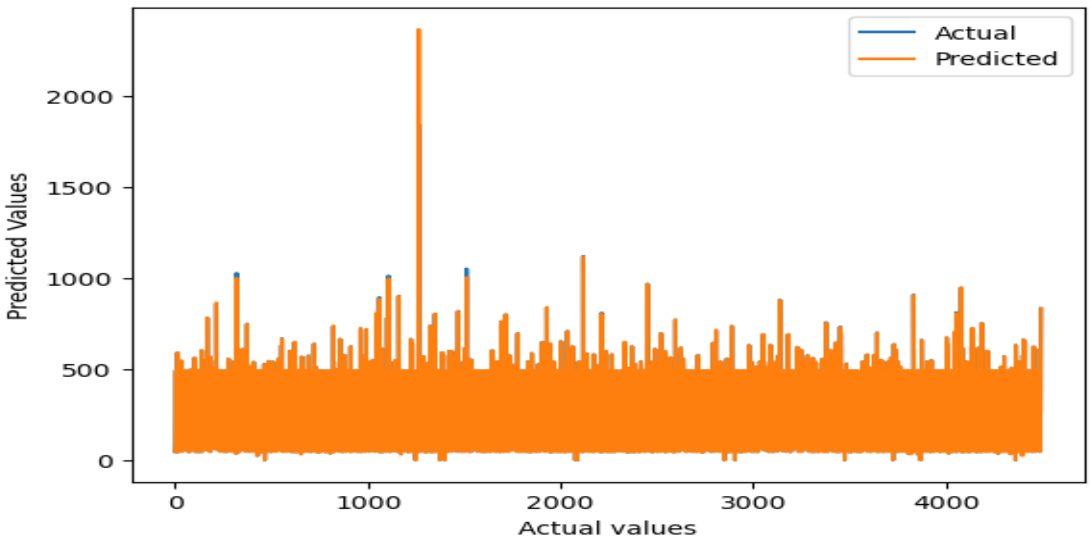

Figure 47 Actual vs predicted values of decision tree when depth is 15

Now, we compare the errors of decision tree which is shown in **Table 36.**

Table 36 Comparison of errors of decision tree

Parameter	Depth 5	Depth 10	Depth 15
MAE	6.308411673872369	0.46407865056296216	0.25740557686322485
MSE	282.9707467507989	64.28417925436828	64.21864192184026
RMSE	16.82173435620712	8.017741530778337	8.013653469038966

Now, we plot this comparison which is shown in **Figure 48**.

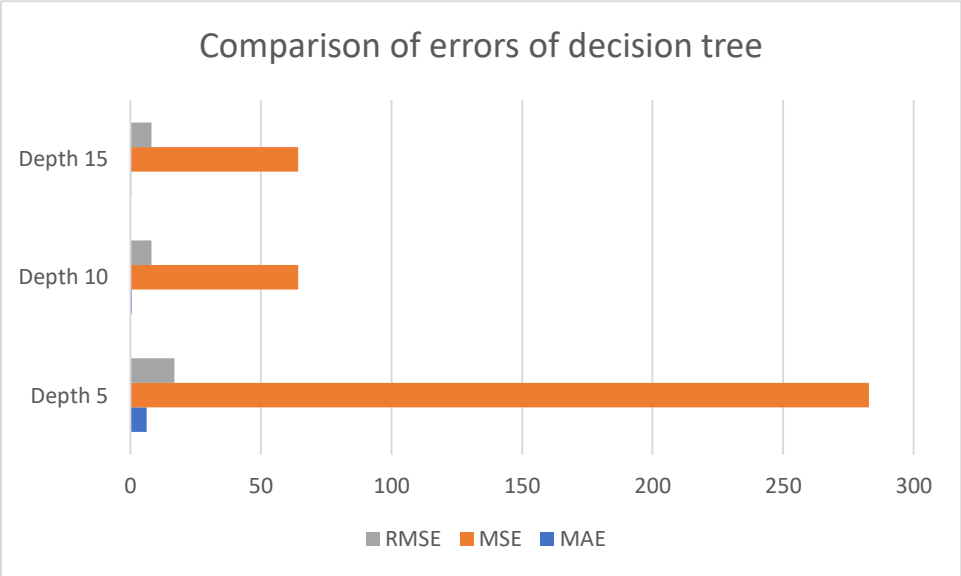

Figure 48 Comparison of errors of decision tree

Now we evaluate the LSTM model. Initially we take 50 epochs and batch size is 30. After evaluating, the errors are shown in **Table 37**.

Table 37 Errors of LSTM when epochs are 50 and batch size is 30

Parameter	Value
MAE	6.198745442562173
MSE	72.56003591510026
RMSE	8.518217883753636

Now we plot the loss graph of this model which is shown in **Figure 49**, where in x axis we plot the epochs values and in y axis we plot the loss values.

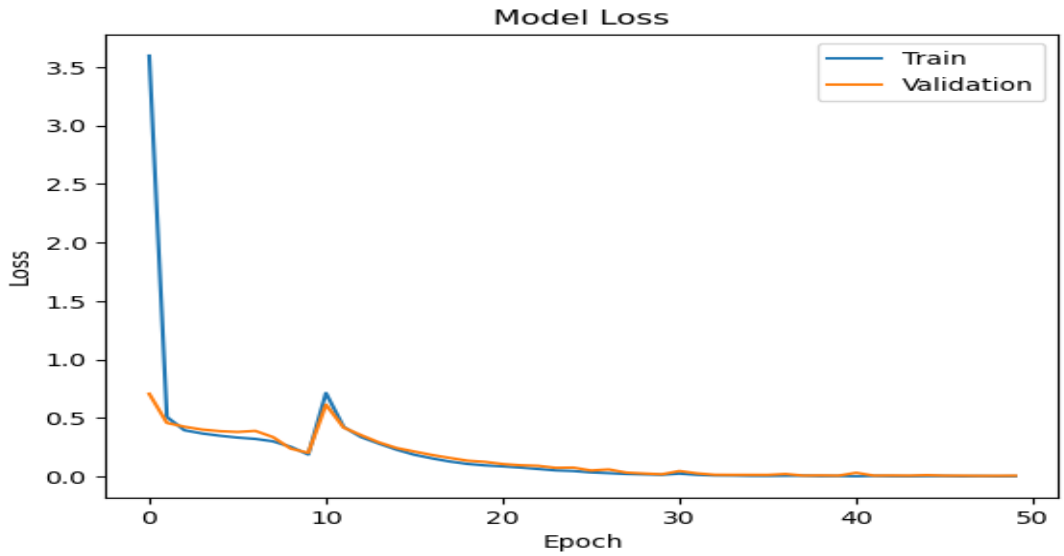

Figure 49 Loss graph of LSTM where epochs are 50 and batch size is 30

Next, we increase the batch size from 30 to 60 and calculate the errors, which is shown in **Table 38**.

Table 38 Errors of LSTM when epochs are and batch size is 60

Parameter	Value
MAE	46822.83126937647
MSE	3411838066.9282775
RMSE	58410.94132890068

Now we plot the loss graph of this model which is shown in **Figure 50**, where in x axis we plot the epochs values and in y axis we plot the loss values.

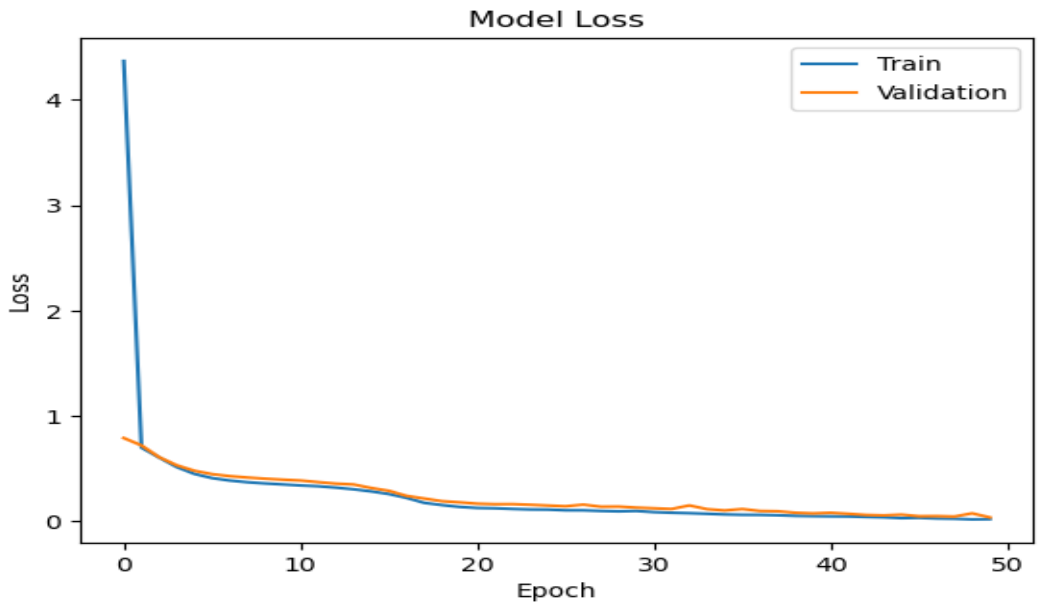

Figure 50 Loss graph of LSTM when epochs are 50 and batch size is 60

Now we compare the errors of LSTM models when epochs is 50, but the batch sizes are 30 and 60. The result is shown in **Table 39**.

Table 39 Comparison of LSTM model when epochs are 50

Parameter	Batch size = 30	Batch Size = 60
MAE	6.198745442562173	46822.83126937647
MSE	72.56003591510026	3411838066.9282775
RMSE	8.518217883753636	58410.94132890068

If we see **Table 39**, we can see that, if we increase the batch size the error values are also increased and difference between the errors are very huge. So, we conclude this method here.

Chapter 5

5. Conclusion and Future work

In this thesis, we conducted a comprehensive time series analysis on weather data and air quality index (AQI) to gain insights and develop predictive models. The primary objectives were to understand the relationship between weather variables and AQI, develop accurate forecasting models, and evaluate their performance. Through this study, we have achieved the following key findings:

- **Correlation between Weather and AQI**: We observed significant correlations between weather variables such as temperature, humidity, wind speed, and AQI. These findings highlight the strong influence of weather conditions on air quality.
- **Prediction Models**: We developed and evaluated several prediction models, including simple linear regression, multiple linear regression, decision tree, and LSTM. Our experiments showed that these models can effectively predict AQI based on historical weather data, with LSTM demonstrating the highest accuracy and predictive power.
- **Accuracy and Performance:** The prediction models exhibited varying degrees of accuracy and performance. The evaluation metrics such as mean MSE provided insights into the effectiveness of the models.
- **Visualization**: Visualizations played a crucial role in understanding the patterns, trends, and relationships in the data. Through plots, charts, and graphs, we were able to gain insights into the time-dependent behavior of weather variables and their impact on AQI.
- **Practical Implications:** The findings of this study can contribute to air quality management and public health initiatives. By accurately predicting AQI based on weather conditions, stakeholders can take proactive measures to mitigate air pollution and its adverse effects.

While my project on time series analysis for AQI and weather prediction utilizing LR, MLR, decision tree, and LSTM models provides valuable insights, however, it is important to acknowledge the limitations of this project, which may affect the accuracy and generalizability of the results.

- **Model Complexity:** While the models used in this project offer valuable insights, they may have limitations in capturing complex relationships and nonlinear patterns present in the data.
- **Time and hardware requirement in LSTM:** Lower hardware resource and software optimization can create slower processing of LSTM epochs.

While this project has provided valuable insights into time series analysis for weather and AQI prediction, there are several areas that can be explored further to enhance the research:

- **Incorporating Additional Variables**: Consider including other relevant variables such as, traffic data, and geographical features such as rainfall patterns, wind direction, elevation, or proximity to bodies of water to improve the accuracy of the prediction models. The inclusion of these factors may provide a more comprehensive understanding of the air quality and weather predictions.
- **Advanced Modeling Techniques**: Explore advanced machine learning techniques such as deep learning architectures (e.g., convolutional neural networks) or ensemble models to further enhance the predictive performance. These methods may capture complex interactions and nonlinear relationships present in the data.
- **Real-Time Predictions**: Develop real-time prediction models that can forecast AQI based on live weather data. Integrating real-time data sources and implementing efficient algorithms can enable timely air quality predictions, allowing for proactive decision-making and prompt interventions.
- **Long-Term Trends**: Investigate long-term trends in weather and AQI data to assess the impact of climate change on air quality. Analyzing historical data and identifying patterns in long-term trends can offer valuable insights into the changing dynamics of air pollution, rainfall patterns, and temperature variations.

References

[1] Z. C. Lipton, D. C. Kale, C. Elkan and R. Wetzell, *Learning to diagnose with LSTM recurrent neural networks.,* p. 18, 2015.

[2] Z. Che, S. Purushotham, K. Cho, D. Sontag and Y. Liu, *Recurrent Neural Networks for Multivariate Time Series with Missing Values,* vol. 8, no. 1, p. 6085, 2018.

[3] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and Practice, 2nd ed., Australia, 2018.

[4] J. Shao, Y. Liu, H. Wu and J. Zhang, "Air quality index prediction based on a convolutional neural network with residual connections (ResNet-AQI)," *Atmospheric Environment,* vol. 214, p. 116834, 2019.

[5] Z. Chen, H. Li and H. Du, "A hybrid model for the prediction of air quality index," *PLoS ONE,* vol. 12, no. 6, 2017.

[6] G. E. P. Box, G. M. Jenkins, G. C. Reinsel and G. M. Ljung, Time Series Analysis: Forecasting and Control, 5 ed., Wiley, 2015.

[7] C. Chatfield, "The Analysis of Time Series: An Introduction", 6 ed., Chapman and Hall/CRC, 2016, p. 352.

[8] H. Lütkepohl, New Introduction to Multiple Time Series Analysis, 1 ed., Heidelberg: Springer Berlin, 2005, p. 764.

[9] R. H. Shumway and D. S. Stoffer, Time Series Analysis and Its Applications, 4 ed., Springer, 2017, p. 562.

[10] B. Chatterjee, S. Acharya, T. Bhattacharyya, S. Mirjalili and R. Sarkar, "Stock market prediction using Altruistic Dragonfly Algorithm," *PLoS ONE,* vol. 18, no. 4, 2023.

[11] D. Mukherjee, R. Mondal, P. K. Singh, R. Sarkar and D. Bhattacharjee, "EnsemConvNet: a deep learning approach for human activity recognition using smartphone sensors for healthcare applications," *Multimedia Tools and Applications,* vol. 79, p. 31663–31690, 2020.

[12] N. N. Maltare and S. Vahora, "Air Quality Index prediction using machine learning for Ahmedabad city," *Digital Chemical Engineering,* vol. 7, p. 100093, 2023.

[13] K. B. Shaban, A. Kadri and E. Rezk, "Urban Air Pollution Monitoring System With Forecasting Models," *IEEE Sensors Journal,* vol. 16, no. 8, pp. 2598-2606, 2016.

[14] A. Bekkar, B. Hssina, S. Douzi and K. Douzi, "Air-pollution prediction in smart city, deep learning approach," *Journal of BIg Data,* vol. 8, no. 161, 2021.

[15] L. D. Monache, F. Eckel, D. L. Rife, B. Nagarajan and K. Searight, "Probabilistic Weather Prediction with an Analog Ensemble," *Monthly Weather Review,* vol. 141, no. 10, p. 3498–3516, 2013.

[16] A. L. Balogun and A. Tella, "Modelling and investigating the impacts of climatic variables on ozone concentration in Malaysia using correlation analysis with random forest, decision tree regression, linear regression, and support vector regression," *Chemosphere,* vol. 299, no. 134250, 2022.

[17] C. Sammut and G. I. Webb, "Mean Absolute Error. In," *Encyclopedia of Machine Learning,* vol. 1, p. 653, 2010.

[18] J. Nevitt and G. R. Hancock, "Improving the Root Mean Square Error of Approximation for Nonnormal Conditions in Structural Equation Modeling," *The Journal of Experimental Education,* vol. 68, no. 3, pp. 251-268, 2000.

[19] "Austin, Texas," Wikimedia Foundation, Inc., , [Online]. Available: https://en.wikipedia.org/wiki/Austin,_Texas. [Accessed 19 May 2023].

[20] "Szeged," Wikimedia Foundation, Inc.,, [Online]. Available: https://en.wikipedia.org/wiki/Szeged. [Accessed 19 May 2023].

[21] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago and O. Tabona, "A survey on missing data in machine learning," *Journal of Big data,* vol. 8, no. 140, 2021.