Prediction of Extreme Precipitation in the Upper Indus Basin Using Time Series Models

By

Aleesha Khan

Regn. No. 3301058

Supervisor

Dr. Tahir Mehmood

A thesis submitted in partial fulfillment of the requirements of the

degree

the degree of *Master of Science in*

Statistics

Department of Statistics

School of Natural Sciences (SNS)

National University of Sciences and Technology H-12

Islamabad, Pakistan

August 2023

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS thesis written by **Aleesha Khan** (Registration No **00000331058)**, of **School of Natural Sciences** has been vetted by undersigned, found complete in all respects as per NUST statutes/regulations, is free of plagiarism, errors, and mistakes and is accepted as partial fulfillment for award of MS/M.Phil degree. It is further certified that necessary amendments as pointed out by GEC members and external examiner of the scholar have also been incorporated in the said thesis.

Signature: ____________________

Name of Supervisor: Prof. Tahir Mahmood

Date: 08/08/2024

Signature (HoD): __________ (For HoD Mathematics)

Date: 08/08/2024

Signature (Dean/Principal): ____________________

Date: 09.08.2024

FORM TH-4

National University of Sciences & Technology

MS THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by Aleesha Khan, Regn No. 00000331058 Titled Prediction of Extreme Precipitation in the Upper Indus Basin Using Time Series Models be Accepted in partial fulfillment of the requirements for the award of MS degree.

Examination Committee Members

1. Name: DR. SHAKEEL AHMED Signature:

2. Name: DR. ZAMIR HUSSAIN Signature:

Supervisor's Name: PROF. TAHIR MAHMOO Signature:

For Head of Department

08/08/2024
Date

COUNTERSINGED

Date: 09.08.2024

Dean/Principal

I dedicate this thesis to my beloved parents.

Acknowledgments

Allah Almighty, the most beneficent and gracious, who created the whole universe, deserves all honor and glory. I am deeply grateful and indebted to Him for bestowing countless blessings upon me, including the courage and strength to complete my thesis effectively. I am indebted to my supervisor, Dr. Tahir Mehmood, for his continued guidance and endless support. His unassuming approach to research and science is a source of inspiration. This approach is reflected by his simple but clear writing style, which is something I hope to carry forward throughout my career. He is the true definition of a leader and the ultimate role model. The levels of patience, knowledge, and ingenuity are something I will always keep aspiring to. I would like to pay my gratitude to my GEC members Dr. Zameer Hussain and Dr. Shakeel Ahmed for their support and guidance in completing this thesis. Countless people supported my effort on this thesis, and my family deserves endless gratitude. I am grateful to my parents whose constant love and support keep me motivated and confident. My accomplishments and success are because they believed in me.

author

Abstract

The Indus River, a transboundary and one of the biggest rivers in the world originates in western Tibet, flows northwest through Ladakh, Gilgit-Baltistan, and then south and southeast across Pakistan and empties into the Arabian Sea. Tarbela Reservoir is one of the biggest being constructed on the Indus River and plays a pivotal role for agriculture, water storage, hydropower, and ecosystem stabilization. This study attempts to model and forecast the monthly extreme precipitation events in the Upper Indus Basin (UIB) using time series and machine learning approaches. The observed data about precipitation has been collected from the Pakistan Meteorological Department (PMD) for the duration of 1960-2013 and then the monthly extreme precipitation was calculated by using Excel. Two classes of time series models, i.e., Autoregressive Integrated Moving Average (ARIMA) whereas LSTM form of Recurrent Neural Network (RNN) form machine learning approach have been used. The performance of these models has been assessed by using various error statistics (e.g., RMSE, MBE, AIC, and BIC) as well as graphically. The findings of this study may help policymakers in water availability, hydropower, infrastructure loss, flooding, and agriculture.

Keywords: *Precipitation, Upper Indus Basin, Time Series Models, Long Term Short Memomry*

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations

ARIMA	Autoregressive Moving Average
RNN	Recurrent Neural Network
DNN	Deep Neural Network
LSTM	Long Short Term Memory
RMSE	Root Mean Square Error
NN	Neural Network
DL	Deep Learning
ML	Machine Learning
AI	Artificial Intelligence

CHAPTER 1

Introduction

To help readers better grasp the study, this chapter describes the thesis project's overall premise. The following is how this chapter is organized. The study of extreme precipitation occurrences in the Upper Indus Basin (UIB) and an outline of contemporary time-series analysis and forecast techniques are presented in section one. In the second section, the study provides a description of the research problem. The following two parts go into further detail about the objectives and aims of the study before presenting the chosen research methodology. After a brief discussion of delimitations, he thesis structure is presented as the chapter comes to a close.

1.1 Section 1

The "climate" is described by meteorologists as the 30-year of the local climate [12, 24]. The term "climate change" refers to shifts in changes in the climate over time, whether caused by natural occurrences or human action [34]. In the middle of the 20th century, scientists came to the conclusion that one of the main causes of climate change is human activity. Along with human activities, other climatic phenomena including solar cycles, earthquakes, and volcanic activity also contribute to environmental degradation [25]. The damaging climate change is predicted to have an impact to worsen in the near future[28]. Climate change, freshwater scarcity, and other factors are posing threats to food supply and hunger, and these problems will get worse in the near future [8]. Due to the rapidly worsening effects of climate change, action must be taken immediately to address the issue and promote climate change adaptation.

Of South Asia's major rivers, the Indus is the furthest to the west. It drains sections of China, India, Afghanistan, and Pakistan along its 3200 km overall length [14]. With an average annual water availability of 1329 m3, the water resources of the Indus basin are expected to support 215 million people [19]. Precipitation in the upper indus basin (UIB), which differs with elevation and in both space and time, is substantially controlled by complex mechanisms including an interaction between synoptic-scale circulations (such as the western disturbances and Indian summer monsoon) and KKH topography [11].

Rainfall is the most significant activity in the hydrological cycle, is and critical for maintaining the global balance of freshwater and saltwater resources [37]. influences due to its direct or indirect impacts on our society, it is arguably one of the most studied and investigated hydroclimatic phenomena [27]. considerable insights in finding the transmission dynamics of infectious disease and its surveillance Additionally, it has a direct impact on agriculture, water resources, ecosystems, and the spatiotemporal unpredictability of freshwater supply.

Precipitation amounts each month are an example of a time-series data, here we have used areas of UIB and predicted the future values of it. Balakot, Chitral, Gilgit, Murree, Muzaffarabad, Peshawar, and Skardu are seven of the stations with the highest precipitation rates. There are numerous analysis techniques available today to find patterns or forecast trends in such data. Statistical approaches (e.g., ARIMA model), (ML) or (DL) methods (e.g., LSTM), and numerous other techniques have been demonstrated to be useful in the analysis of time-series data, and some of them have already yielded significant insights into the dynamics of infectious disease transmission and its surveillance. considerable insights in finding the transmission dynamics of infectious disease and its surveillance [6]. The ARIMA model is frequently employed in the disciplines of finance, medicine, and meteorology [41, 28]. According to studies, ARIMA was also effective at forecasting precipitation. Although neural networks are a crucial tool for forecasting the weather, BP neural networks lack memory and feedback. Time series data's relationship between the past and the future cannot be reflected in the output, which only takes into account the current input. Due to its unique network topology, the LSTM network has been popular in recent years because it is good at handling time series data.

The ARIMA and LSTM based techniques were used in this work to model and predict the same relative time series. Comparatively, both approaches proved to be successful at forecasting relative precipitation.

CHAPTER 2

Literature Review

2.1 ARIMA

Few scholars have tried to explore time series analysis utilizing the Arima Model in the past. We will talk about some of the studies that used Arima for predicting in this section of our thesis. In a study, a model of monthly precipitation in Urmia city was developed using data on monthly rainfall from the past 68 years collected by the Iranian Meteorological Organization.We did this by using two different methods, ARIMA and gene expression programming (GEP). For the GEP and ARIMA modeling, ten and twenty-five unique patterns, respectively, were taken into consideration.Modifying the delay in the precipitation time series created the patterns. 80of the data was used for model training and 20for model testing in both procedures. To compare the model performances, three statistical measures—root mean square error (RMSE), coefficient of determination (R2), and mean absolute error (MAE)—have been used.[26].

In 2015, a researcher by the name of "S H BARI" predicted monthly precipitation for Sylhet City using a seasonal ARIMA model built using the Box and Jenkins method. Precipitation data from the Sylhet station from 1980 to 2010 were used to build and evaluate the model. Data on rainfall from 1980 to 2006 were used to develop the model, while data from 2007 to 2010 were used to evaluate the projections' accuracy.The four key historical procedures of identification, estimation, diagnostic testing, and forecasting were fitted out to create the model. The model's validity was assessed using Box and Jenkins' usual graphical residual description. As a second step in the validation procedure, predicted monthly rainfall quantities were verified using actual data sets. after making all necessary inquiries and anticipated observations, The model with the highest accuracy for predicting future precipitation with a 95confidence interval was the

ARIMA(0, 0, 1) (1, 1, 1)12. The optimal scheduling of crop management, urban planning, rainfall collection, and flood prediction are anticipated benefits of this long-term forecast for decision-makers.

The rainfall in an Indian coastal region can be forecast using a Time Series Modeler (TSM). This model was developed using a five-year dataset (2009–2013) with major attributes such as temperature, dew point, wind speed, maximum temperature, minimum temperature, visibility, and rainfall. This dataset has been trained and tested using the novel method of the Statistical Package for Social Studies (SPSS) TSM. Because the performance criteria for this model's evaluation are based on the significant values of the statistical performance measures, namely (MAD), (MSE), (MAPE), and (RMSE), s a result, a viable model for rainfall prediction is now possible. At 80The prediction accuracy range of the results generated by this model is largely satisfactory.

This model is based on the SPSS 20.0 TSM Arrima. [21].

Using historical data, the ARIMA time series model was used in a different study to estimate the price of gold in Indian browsers with the goal of lowering the risk associated with gold purchases. To advise the investor on the best times to acquire and sell yellow metal. Researchers, investors, and speculators are all on the lookout for alternative financial tools to diversify and reduce risk in their portfolios because the Indian economy is hampered by reasons including the changing political landscape, global cues, and rising inflation, among others. [22].

2.2 Machine Learning

This area starts with the concept of non-linear regression, machine learning, simple and multi-layer neural networks, different types of models used for predicting quantitative outcomes, and the selection of models based on their prediction power.

2.2.1 History of Machine Learning

Charles Babbage is a well-known name in history because he was the one who introduced the mechanism of the machine when no one knew that there will be a time when people can do their difficult tasks in one click. He invented the computer of huge sizes like a size of a room in the middle of 1830 and this faced huge criticism from the orthodox[18]. At that time all people were unaware of the power of this innovation and never knew it will become a massive part of human's daily life activities. Nowadays it would not be a surprising statement if we say, humans are machine driven and day after day humans are becoming more and more dependent on machines. We are all crowded by machines like a person can take a breath through a machine, from birth to death we are on machines. But there was a drawback in the invention of Charles Babbage that a computer can do only that task which human asked for. These program-based machines are only capable of doing those tasks for which human put information as input and get outputs but if human change anything or needs to know a little bit change of it they can't do that. They cannot exceed their defined limits. So study has machine learning, which is the scientific technology of making machines that can do tasks with more efficiency and in less time. This technology wanted to give the idea that humans and machines can do the same job but machines can do these tasks with high precision as defined these machines (computers) cannot do the task for which they do not have operating algorithms[32]. So, people work on artificial intelligence (AI) in machines, there has been so much hard work, so much struggle and it is expected that these machines would be capable of operating and doing and improving themselves in the coming future and they would not be dependent on humans.

2.2.2 ML vs DL

Artificial intelligence is the idea of making the smart intelligent powerful machines. ML is a subspace of AI which helps human to create AI-driven applications. ML is a basic type of science but science is not originated from it. Most people used to think these three concepts ML, DL, and neural network are related to each other but in reality, these are sub-sections of artificial intelligence[20]. ML comes first in this lane, DL is a sub-section of machine learning while neural network is a sub-section of deep learning. The algorithm learning techniques in both cases are the main difference between DL and ML. In ML human interference is more as compared to deep learning because it has a manual task system. So in this humans make sure that from the data algorithm is learning, built-in nature is seeing when network is doing deep learning process, this process is more automated. In machine learning the different mechanisms to put the values into the system can be seen and explain the system about the algorithm. So in the DL process, it automatically learns algorithms from the data, adjusts the weights automatically in every iteration, gives a suitable learning rate, etc[20]. The importance of ML is that the system can train any network without knowing the deep learning and the same for the DL but when the working is with deep learning then to know about the machine learning makes the network easier and understandable.

We used illustrations to explain how ML, DL, and AI are related. The primary area where the system imitates the brain is in artificial intelligence. AI's branch of ML, or DL, uses statistical methods to enable systems to get better and more advanced over time.

Deep learning has two types of models which are SL and USL. In SL,Data must be identified as input and output for the system. and the algorithms of supervised learning "learn" from the training data set and do iterations and adjust weights to minimize the error rate, but also keep in mind that labeled data is not always required for supervised learning. At the same time in unsupervised learning system do not need to have labeled input or output, it uses unlabeled or raw data sets. It does not make its algorithms. it depends on machine learning algorithms and makes groups of unlabeled data set. Deep learning is more powerful to handle unlabeled data sets (i.e., texts, images) and after that classify these data sets into groups or clusters according to their similar properties[18]. But in machine learning the situation is different if the system puts unlabeled or raw data e.g., of either texts or images, firstly, it needs to put information manually into an algorithm about classification, and then it becomes a banal and boring task to even control data in machine learning. Due to this problem or flaw of ML, DL and NN were introduced which have a massive number of uses in every area like speech recognition and language processing.

Machine learning algorithms have three steps to work i.e., a decision process, an error function, and last model optimization process.

- Most of the time machine learning is used either in the classification of data or prediction of data. Machine learning starts the process by inputting training data into the already chosen algorithm and it trains the system to make difference between two objects suppose a lion and a cat. There are billions of characteristics that made a lion different from a cat. From this, it concludes, that it requires learning every characteristic of an object. Every characteristic is important to machine so that it can make a difference between two objects and recognize them successfully. So the sum up is machine learning will take the data as input, then train this data manually, after that algorithm will learn from it and then do classification[17].

- Later an error function is defined which finds the error between actual and predicted outcomes.

- In the last estimate the model by weights adjustment and try to reduce the error. And this whole process will happen n number of times until the required accuracy is achieved[40].

2.2.3 Types of Machine Learning Methods

The classification of machine learning is in three subcategories i.e., supervised machine learning, unsupervised machine learning, and semi-supervised machine learning. Supervised machine learning defines the labeled data only. It does not work with raw data and unsupervised learning defines the raw or unlabeled data but semi-supervised learning has the qualities of both learning, supervised learning and unsupervised learning.The detailed work on these learning techniques are given below.

2.2.4 Supervised Learning

There are many different types of data sets and so study cannot define a general mechanism for everyone. Each data set has different properties so due to different types of data sets and their properties machine learning has been divided into different methods for various types of data.

Supervised machine learning gets information through specified or labeled data to observe the native nature of data and make future predictions easier. The condition which is mandatory for supervised machine learning is only to mention the type of data, whether the data is from labeled data or unlabeled data where a label can define as a data set that already has been attached. If the data is in raw form or not labeled then supervised machine learning is not a good approach to apply to data to make predictions. The predictions which network wants to make will not give accurate results with raw data in the supervised technique. Supervised learning data sets are created to train algorithm into classifying data and they use training data set which contain labeled input and give a correct output which helps the model to learn fast. The example which is best to explain supervised learning is the "text classification problem", in this the aim is to classify and predict the class label of a given text. Another example related to text classification is to classify or predict the sentiments of text pieces same as tweets and product reviews.

In given below 2.1 the whole mechanism of can be seen . There is a labeled data, in the next step it will go for the model training to ready the system, and go for testing and prediction of the data to check whether it predicts the right or wrong data. Different shapes can be seen in labeled data. Due to different geometrical shapes the data has been labeled according to that and after that algorithm learns from it and in the last the categories prediction of geometrical shapes correctly.

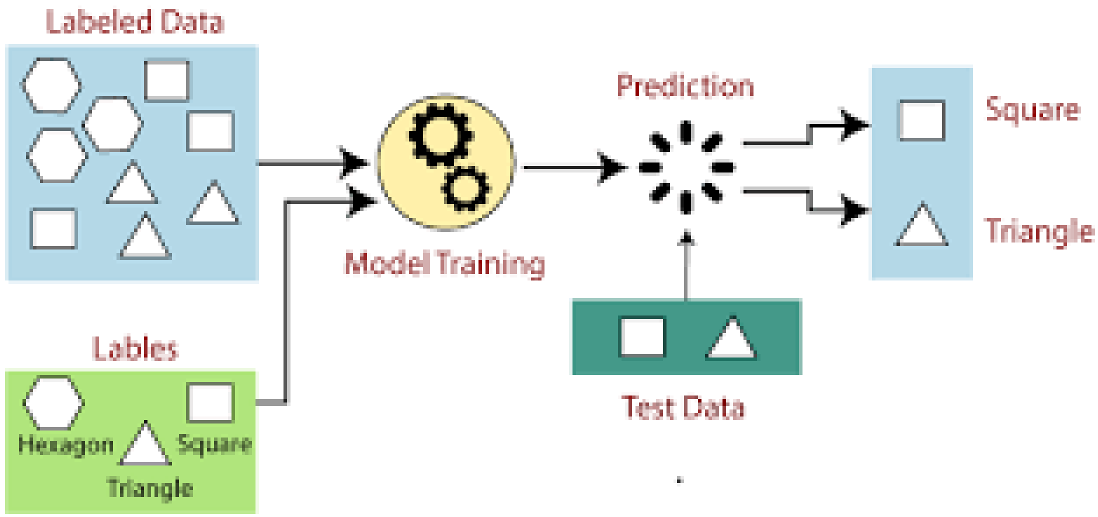

Figure 2.1: This figure illustrates the relationship between AI, ML, DL

Supervised learning has different types of uses in the context of statistical modeling.

- In regression analysis, there is immense use of supervised learning ways in which it observe and finda dependent variable's reaction to disparate independent variables and in the analysis make predictions that depend on resultant coefficients.
- When the system faces binary output data then It fits the data using a logistic function, this is familiar as logistic regression. The logistic regression function and simple regression are somehow similar to each other. Any technique and method has its pros and cons same as supervised machine learning way also has some positives and some negatives and the following are those.
- Historical data has a high weight in supervised learning so it makes it easy to do predictions because statistics and statistical analysis are almost dependent on past manners.
- The optimization power of supervised learning techniques is quite high because it depends on the memory of the data or also known as the memory-based models so it optimizes the data very well.
- The real world have a massive number of problems in which human can use supervised-machine learning methods to address these issues.
- Supervised machine learning algorithms are best than unsupervised but training these algorithms is not an easy task because the system has to select many samples from every

class while training therefore making time is a big arrangement[35].

2.2.5 Unsupervised Learning

The important point of supervised learning is network can only deal when the data set is labeled but when the unlabeled data set present then study goes for unsupervised machine learning. These are the extensions of machine learning and they are good for searching hidden patterns, making groups or clusters inside the data, customer segmentation, cross-selling strategies, and image recognition. Human interaction is not involved but has massive use in data analysis[31]. It is best in finding patterns but does not know exactly what is the objective. What is the main purpose behind all the processes? What does system need to do? What other is thinking? So it is very helpful in cyber-security where in the system attacker always go for the different methods. So the summary points are: It learns the patterns, designs, and structure from the unlabeled datasets, make predictions, and do analysis. In real life it is a more suitable technique because the system usually has raw values.

Where the data is available, the design of unsupervised machine learning can be seen., which is in a raw form go for interpretation so in this there is no training of data is happening then application of algorithm happens so the process of the method starts and in last system gets output.

Unsupervised machine learning has a massive amount of uses in many areas but in clustering, association, and dimension reduction it is best. Here are some of them:

- clustering is a purely statistical concept but can be used in data science. In clustering algorithm, use unlabeled data, process it, and transform it into groups with the help of their patterns. Cluster algorithms could further be classified into overlapping, hierarchical, and probabilistic types.

- Initially the data values are separate and then made into clusters based on their similarities and this until one cluster unit is achieved is hierarchical clustering. In probabilistic clustering, solve the complications of soft clustering or estimation, and the method of making and selecting groups is based on likelihood e.g., through the Gaussian mixture model. To examine the link between variables, association is utilized.. This method is valuable in industries like when companies need to check the relationship between their products and on this basis, they increase or decrease the sale of product and also works in recommendations of products on an online store where if someone buys a table then program recommend them a chair, lamp, diary, etc as relevant products[29].

- It is a common perception about data if the system has more data means more information but it can be a problem at some points where the problem of over-fitting and extra values, which are not related can be seen. So it makes it complicated for an algorithm to learn. For that situation, dimension reduction techniques are best. They eliminate some features of data to shrink the size of the data. This merely eliminates the extraneous information and maintains the data's integrity.[38].

2.2.6 Semi-Supervised Learning

Combining supervised and unsupervised machine learning is semi-supervised machine learning. If the data set is the combinationand a significant section of it is unlabeled then go for semi-supervised machine learning in which it gets the benefits of both supervised and unsupervised learning so it gives the bridge between two approaches[44]. Semi-supervised Machine Learning has various uses in real time some of them are given here: It is used for clustering the objects which helps to classify the class of that object, an outcome variable, cluster labels, or information about the relationship when it is known. The main advantage of the combination it is easier to understand and makes applications more simple. If the system works with semi-supervised learning the application of that algorithm becomes very easy. It is very helpful in the reduction of the annotated data.

The algorithm of semi-supervised has data in a raw form then the system goes for the training of the model so that it can work accurately for the network Then its algorithm works and the system gets the desired output or predictions from the network. An example of semi-supervised learning is a text documents classifier. This helps to remove the noisy items from the data which makes the computations more efficient and also best for the model. It is a more suitable algorithm and the computation is very simple and the system gives the result with very high accuracy. There are different algorithms which are based on semi-supervised technique some of them are listed below:

- Self training; It is the re-sampling method which again and again labels the unlabeled training samples.

- Graph Based semi-supervised machine learning; The more significant sub-section of semi-supervised learning and nowadays one of the most famous algorithms. The steps to make it are as follows:

- Graph development
- One the subset of nodes, system has to infuse the seed marks
- The unlabeled nodes in the graph have to be mentioned with labels.
- Low-Density Separation; In this system has decision boundaries and these decision boundaries should be in between the low-density region.

2.3 LSTM

Few researches were made for the forecasting of different things by using LSTM. Here are some of the research:

The financial market's anomaly prevents simple designs from making more accurate predictions of future asset values in the stock market. The current main trend in scientific research is ML, which involves teaching computers to perform tasks that would normally need human intelligence. Simple models cannot anticipate future asset values in the stock market with greater accuracy due to the anomaly of the financial market. Machine learning, which involves teaching computers to carry out tasks that typically require human intellect, is the dominating scientific research trend at the moment.[36].

Even while (LSTM) is widely accustomed to assessing and forecasting time series-based events, It is challenging to resolve extremely long-term dependencies, likely because LSTM he number of errors increases as the length of the sequence increases. Recent research has revealed that adding characteristics across a range of time scales can aid in the improvement of RNN's long-term reliance when taking into account the importance of historical data in traffic flow forecasts. The attention mechanism served as a model for the RNN. We present a more effective technique that relates the attention mechanism that transfers high-impact traffic flow data from very long sequence time steps to the current time step. We simultaneously to improve prediction outcomes, and smooth out some data that is outside of the average range. [30].

For the purpose of month-by-month precipitation forecasting, an (LSTM) NN and (ARIMA) are recommended. It was developed using monthly precipitation data from the past for Luoyang

City from 1973 to 2021. The frequency components of the model produced by EEMD decomposition were split entering the realm of high-frequency and low-frequency parts using the Permutation Entropy (PE) approach. The low-frequency sequence is predicted by the ARIMA model, whereas the high-frequency sequence is predicted by the LSTM model. The outputs of the two models are combined to produce monthly precipitation projections. Finally, a variety of assessment markers are used to evaluate the anticipated performance. It also suggests that the model has a high level of confidence in its predictions of future precipitation. [43].

STM and Arima are both used for precipitation in a number of countries. In order to estimate precipitation month by month, a linked model based on (EEMD), (LSTM), and (ARIMA) has been developed. The model was created using monthly precipitation data from the past for Luoyang City from 1973 to 2021. Different frequency model components produced by EEMD the breakdown was split into high-frequency series sections. and low-frequency series parts using the Permutation Entropy (PE) approach. The outputs of the two models are combined to produce monthly precipitation projections. Finally, a variety of assessment markers are used to evaluate the anticipated performance.

2.4 ARIMA-LSTM

:

In the University of California, a study, (LSTM) network and the ARIMA model were used to compare how well they predicted the price of bitcoin in US dollars. Pycurl gathered current price information from Keras and TensorFlow is utilized to put the LSTM model into action. The major goal of the ARIMA model used in this study is to provide a typical time series forecasting comparison. As predicted, Only capable of forecasting accurately for short time intervals, and the final result is time-dependent. With additional, necessary time for model training, notably via CPU, the LSTM could perform better [33].

Today, using historical data to forecast time series is quite crucial. It is used in numerous industries, including banking, healthcare, and meteorology. For any offline or online firm, financial data analysis of profits is essential. It aids in forecasting future values and aids in understanding sales, earnings, and losses. The techniques (ARIMA), (SARIMA), and the DL method (LSTM) Neural Network model in the prediction of time series, have been selected for this efficient investigation. Not for SARIMA and LSTM, but for ARIMA, it has been transformed into a stationary dataset. Results indicate that in building the model which is best, LSTM outperforms pair of statistical models [42].

Due to its application in producing clean energy and the ability to integrate wind speed forecasts into the electricity grid has emerged as the the most fascinating subject for scholars working on renewable energy. Currently, there are a variety of time series forecasting techniques and models available. Deep learning techniques have improved to the point where it is now possible to build multi-step prediction models that are more sophisticated than shallow neural networks (SNNs). To be accurate and sufficient, the problem of long-term wind speed forecast must be resolved. To do so, find the most precise and effective time series forecasting model, this study compares the (ARIMA) to various (ANNs), (RNNs), and (LSTM), a particular kind of RNN model. In order to calculate the results, the root mean square error (RMSE) method is applied. The comparison demonstrates that the LSTM method is more exact than the ARIMA method. [39].

2.5 Problem statement

As it is evident from the review of the literature available Arima and LSTM are still not used for the precipitation prediction in Pakistan. Our problem statement was to predict future values for precipitation in the stations of the Upper Indus basin.

CHAPTER 3

Data Explanation

3.1 Data Source

The precipitation data we used in this research has been collected from the Pakistan Meteorological Department (PMD) for the duration of 1961-2013. Data consists of the daily precipitation value of the different cities of Pakistan. So for 63 years, the values for each city was $19,700$.

3.1.1 Pre Processing

In this research,h we have used monthly extreme precipitation values for each city. Initially, we had 19,700 values for each city. Then for the monthly values, we used excel and extracted extreme monthly values for each city. we have found 648 values for the monthly data.

3.1.2 Study Area

As mentioned earlier, our study attempts to model and forecast the monthly extreme precipitation events in the (UIB) using time series with machine learning approaches. Areas in the UIB are Gilgit, Chilas, Balakot, Murree and Peshawar which can be seen in the map 3.1 :

3.1.3 Balakot

38 kilometers separate the city of Mansehra from Balakot, which is located in Khyber-Pakhtunkhwa. the Kunhar river, which runs out of the city and joins the River Jhelum at Muzaffarabad Azad Kashmir and has its source in Lake Lulusar. Balakot experiences warm summers and chilly winters.

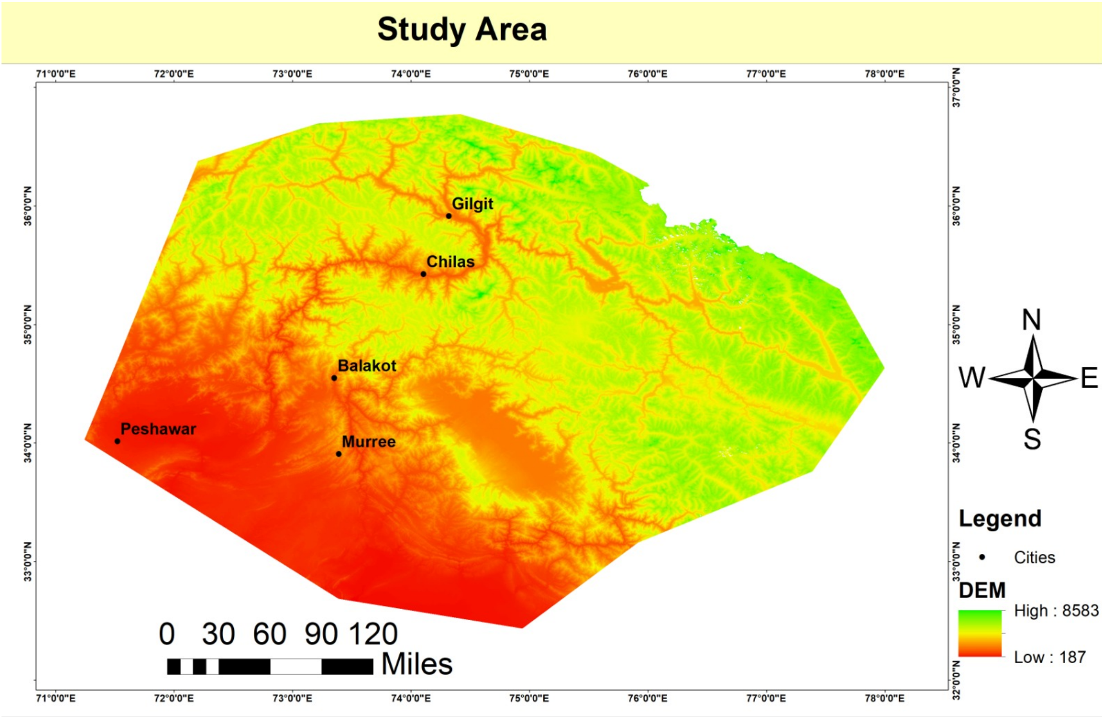

Figure 3.1: Study Area Map

3.1.4 Murree

The subdivision of Rawalpindi District that is located at a very high altitude in the outer Himalayas is called Murree.

3.1.5 Gilgit

The northern region of Pakistan is where the Gilgit valley is located. The Gilgit city serves as the valley's capital. With an elevation of 1500 mm, it is one of Pakistan's highest locations.

3.1.6 Peshawar

Peshawar serves as the provincial capital of Pakistan's Khyber Pakhtunkhwa. Very scorching summers and similarly chilly winters are the norm in Peshawar.

3.1.7 Chilas

It is the divisional capital of Diamer Division and is located on the Indus River. Very hot in summer and similarly chilly in winter.

CHAPTER 4

Methodology

4.1 Time Series Analysis

Time series analysis is a statistical approach used to analyze data that has been collected or recorded over time. It deals with time-ordered data point analysis, modeling, and forecasting. Numerous disciplines, including economics, finance, meteorology, engineering, and more frequently use time series data. Here is a thorough explanation of the fundamental ideas and procedures of time series analysis:

4.1.1 Time Series data

Data from time series are observations or measurements that were made at regular intervals. The chronological arrangement of these data makes time the independent variable.

4.1.2 Components of Time Series

- Trend: The overall direction or pattern of the data is represented by a trend, which is a long-term, sustained movement in the data. An upward, declining, or flat trend is possible.
- Seasonality: A recurring, periodic pattern in the data, such as daily, monthly, or annual cycles.
- Cyclic Patterns: Unlike seasonality, cyclic patterns are long-term oscillations that are less predictable. They frequently come from commercial or economic cycles.
- Irregularity/ Noise: Random changes or unanticipated fluctuations that cannot be traced

to the aforementioned elements are considered irregularity or noise.

4.1.3 Stionarity

If a time series statistical characteristics, such as its mean, variance, and autocorrelation, remain stable throughout time, it is said to be stationary. ARIMA is one of several time series models that make this assumption. To become stationary, non-stationary data may require differencing.

4.1.4 Data Visualisation

- Time Plots: Data points are displayed against time to show patterns, seasonality, and abnormalities using the fundamental visualization method known as a time plot.
- Decomposition:Decomposing a time series into its trend, seasonality, and residual components will allow you to examine each one separately.

4.1.5 Autocorrelation and Partialautocorrelation

- ACF: Autocorrelation is a statistical technique that assesses the link between a time series and its lag values at different points of time delays.
- PACF: More precise type of autocorrelation known as PACF examines the correlation between a time series and its lagged values while accounting for the impact of intermediate delays.

In ARIMA modeling, ACF and PACF charts assist in determining the relative importance of the moving average (MA) and autoregressive (AR) components.

4.1.6 Model Building

ARIMA combines moving average (MA) and autoregressive (AR) components with differencing to manage non-stationarity.

Holt-Winters and other techniques can identify patterns and seasonality in data. Time-varying dynamics of a system can be captured by models like the Kalman filter.

4.1.7 Model Evaluation

- Holdout Samples: A portion of the time series data is withheld to evaluate the predicting precision of the model.
- Residual Analysis: Examine the residuals (the discrepancies between the actual and anticipated values) to see whether the model is adequate.
- Metrics for Forecast Accuracy: Metrics for forecasting accuracy include (MAE), (MSE), and (RMSE).

4.1.8 Forecasting

The model may be used to forecast or predict the future once it has been developed and validated.

4.1.9 Seasonal Decomposition of Time Series (STL)

A method for dividing time series data into trends, seasonal, and residual components as an alternative to ARIMA.

4.1.10 Advanced Methods

- GARCH: For modeling volatility in financial time series, use GARCH (Generalized Autoregressive Conditional Heteroskedasticity).
- Machine Learning: Time series analysis and forecasting are done using cutting-edge methods including recurrent neural networks (RNNs), LSTM networks, and Prophet.

4.1.11 Software and Tools

The study is facilitated by a number of software packages and libraries, including R, Python (using libraries like statsmodels, scikit-learn, and Prophet), and specialist time series analysis applications.

Understanding and predicting data with temporal connections may be done effectively using time series analysis. The kind of data being used and the precise forecasting objectives determine the approach and model to be used. It is fundamental to comprehending and forecasting temporal trends in many different domains.

4.2 Auto regressive Moving Average

Time series data are utilized in an autoregressive integrated moving average, or ARIMA, statistical analysis model to either improve the interpretation of the data set or anticipate future developments. Autoregressive models forecast upcoming values with the help of previous values. For example, an ARIMA model can estimate a forecast a stock's price based on its performance or forecast a company's profitability based on prior periods.

- (ARIMA) model forecasts future values using historical data
- ARIMA moving averages are used to blur time series data..
- They are commonly used in technical analysis to estimate future asset price fluctuations.
- Autoregressive models use the implicit assumption that the future will repeat itself.
- The underlying assumption of the future will resemble the past in autoregressive models.

4.2.1 Understating of Arima

An Arima model is a sort of regression analysis that evaluates the significance of just one variable that is reliant in relation to a set of modifying a variable. It estimates later securities or financial market motions made by analyzing variations in rather than the actual value, series values. The figure shows the Arima model's structure 4.1:

$$ARIMA(P,D,A) = AR(P) + Difference(D) + MA(Q) \tag{1}$$

Here is the prediction model:

$$x_t = \alpha_0 + \alpha_1 X_{t-1} + \alpha_2 X_{t-2} + \cdots + \alpha_p X_{t-p} + \varepsilon_t + \gamma_1 \varepsilon_{t-1} + \gamma_2 \varepsilon_{t-2} + \cdots + \gamma_p \varepsilon_{t-p} \tag{2}$$

Each of an Arima model's parts can be simply comprehended as follows:

- Autoregressive (AR): A model in which one of the variables changes and its own regression prior to, or lagged, outcomes.
- Integrated (I): is representative of the changing of fresh data in order to make the time series stagnant (i.e., To replace the data values, the difference between the data and prior values is used).

- Moving Average (MA): includes the relationship between a moving average model residual error applied to lagged observations and an observation.

It can be easily understood by the following figure:

4.2.2 Paramters

Each ARIMA component serves as a parameter and makes use of standard terminology. The typical nomenclature for ARIMA models is ARIMA with P, D, and Q, where integer values are utilized the following parameters can be configured:

- P is the order of lag is another name in terms of the number of observations with a lag.
- The degree of distinction is defined as tthe number of times the raw observations were made are differentiable or D.
- Q is the moving average window size, often known as the moving average order.

For example, a linear regression model includes the quantity as well as the type of phrases. If the parameter's value is zero, it means the component should be avoided. The method allows the ARIMA model to be built in order to accomplish the same function as an AR MA just a straightforward AR, I, or MA model.

4.2.3 ARIMA AND STATIONARY DATA

An ARIMA model is differenced with the data in order to make it stationary. It exemplifies stationary shows that the data are stable throughout time. Because most economic and market data exhibit patterns, with certain exceptions. aims to get rid of any patterns or seasonal structures that may be present.

The regression model may suffer from seasonality, or when data display recurrent,over the course of a year, predictable trends. If a pattern emerges and Stationary does not exist. immediately apparent, many of the computations performed during the procedure cannot be reversed and did not achieve the desired results.

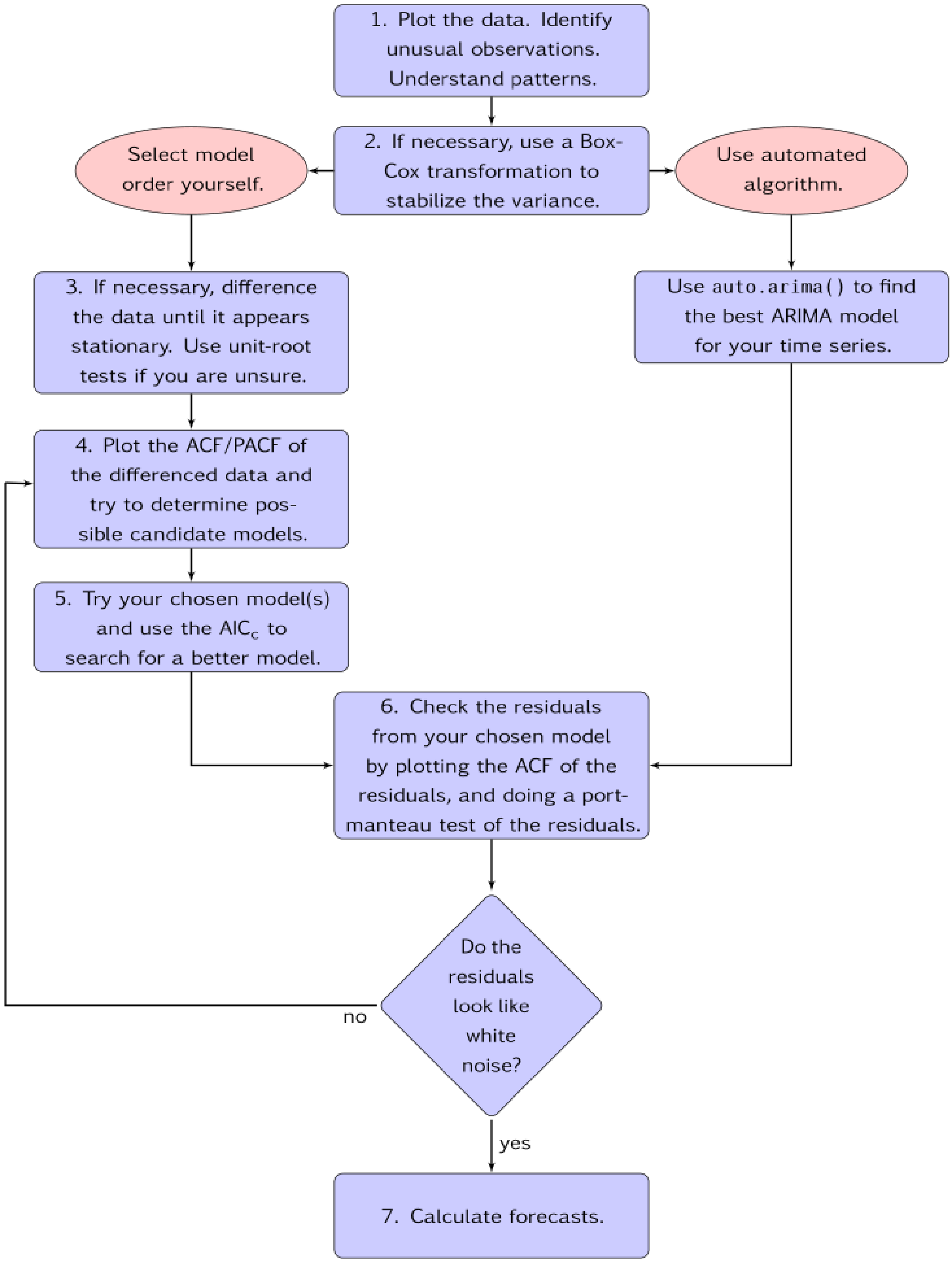

Figure 4.1: Modeling flow cart for Arima model

4.2.4 Arima Model Building

You set about creating an ARIMA investing model by downloading as much price information as possible. The autocorrelations can be used to find the smallest order of difference (D) once the patterns in the data have been identified. The series is divided if lag-1 is autocorrelation is 0 or negative. You might have to differentiate it again if the lag-1 is more than zero (https://people.duke.edu/ rnau/411arim2.htm).

The regression (P) and moving average (Q) order are then determined by contrasting autocorrelations and PCA. Upon having the required data, you can select the model to use.

4.2.5 Pros and Cons

There are numerous reasons to exercise care even if Arima models have their benefits as well as successful in making predictions based on previous experience data. Contrary to make an investment warning that states, "Past performance is not an indicator of future performance...," Arima models rely on historical data to forecast the future by assuming that there is some residual value in previous values. influence on present or future values.

Other ARIMA features that exhibit both positive and negative traits are listed in the following table:

4.2.6 What is ARIMA used for?

Future events can be predicted or foreseen using a preceding time series and the ARIMA approachIt is founded on the statistical concept of serial correlation, which implies that Previous data points have an impact on succeeding data points.

4.2.7 Difference between AR and MA:

Moving average and autoregressive properties are combined in ARIMA. For example, an AR(2) process bases the current value on the two values preceding it, It is based on the value that came before in an AR(1) autoregressive process.A moving average method is used to assess data points and reduce the influence of outliers by averaging different subsets of the whole data set. They can predict values with incorporates trends, cycles, seasonality, and other non-static types of information by integrating these techniques.

4.2.8 How does Arima forecasting work

The prediction is accomplished by accumulating data from time series for the variable of interest. Statistical software will determine the correct no. of the amount of differencing or delays to apply to the data, as well as the stationarity of the data.

4.2.9 Bottom line

Based on historical data, the ARIMA model is used as a predicting tool to create forecasts about potential coming behavior. In technical analysis, it is used to project an asset's future performance.

For long-term forecasting, projections longer than six months in the future, ARIMA modeling is often unsuitable since it depends on past data and characteristics influenced by human intellect. As a result of this, it functions in conjunction in addition to additional technological analysis tools to provide a more comprehensive picture of the performance of an asset.

4.3 Deep Neural Network

Neurons in our brains connect to one another and send signals when they are active, which is how neural networks were first conceptualized. This structure of neural network is represented by a graph, similar as in graph theory of mathematics in the form of pairwise interlinked functions.. In this figure, working of neurons has been shown. There is a single input layer and in it, there are three input neurons which are x1, x2, and x3 which take values then the arrow between them shows the activation functions which are working for the process there are three weights w1, w2, and w3. Firstly, the first neuron which is x1 goes with the weight and bias to the hidden layer then moves towards the output then again the first neuron is now attached to the second weight and again goes to the end. This process goes for the second input neuron and then goes for every neuron in the network. And in the last system gets the error rate for the networks.

The primary goal is to develop an NN that, with the aid of weight and learning rate adjustments, minimizes the mean square error loss function. The step size at each iteration or epoch, while the network is working to reduce the cross entropy LF, is indicated by the tuning parameter known as learning rate, which has a range of 0 to 1. For certain jobs, multiple algorithms exist. Based on historical data, the ARIMA model is used as a prediction tool to create predictions about potential future behavior. It is used in technical analysis to project an asset's upcoming performance. In figure 4.2 a neural network consisting of 2 hidden layers is shown. The circles

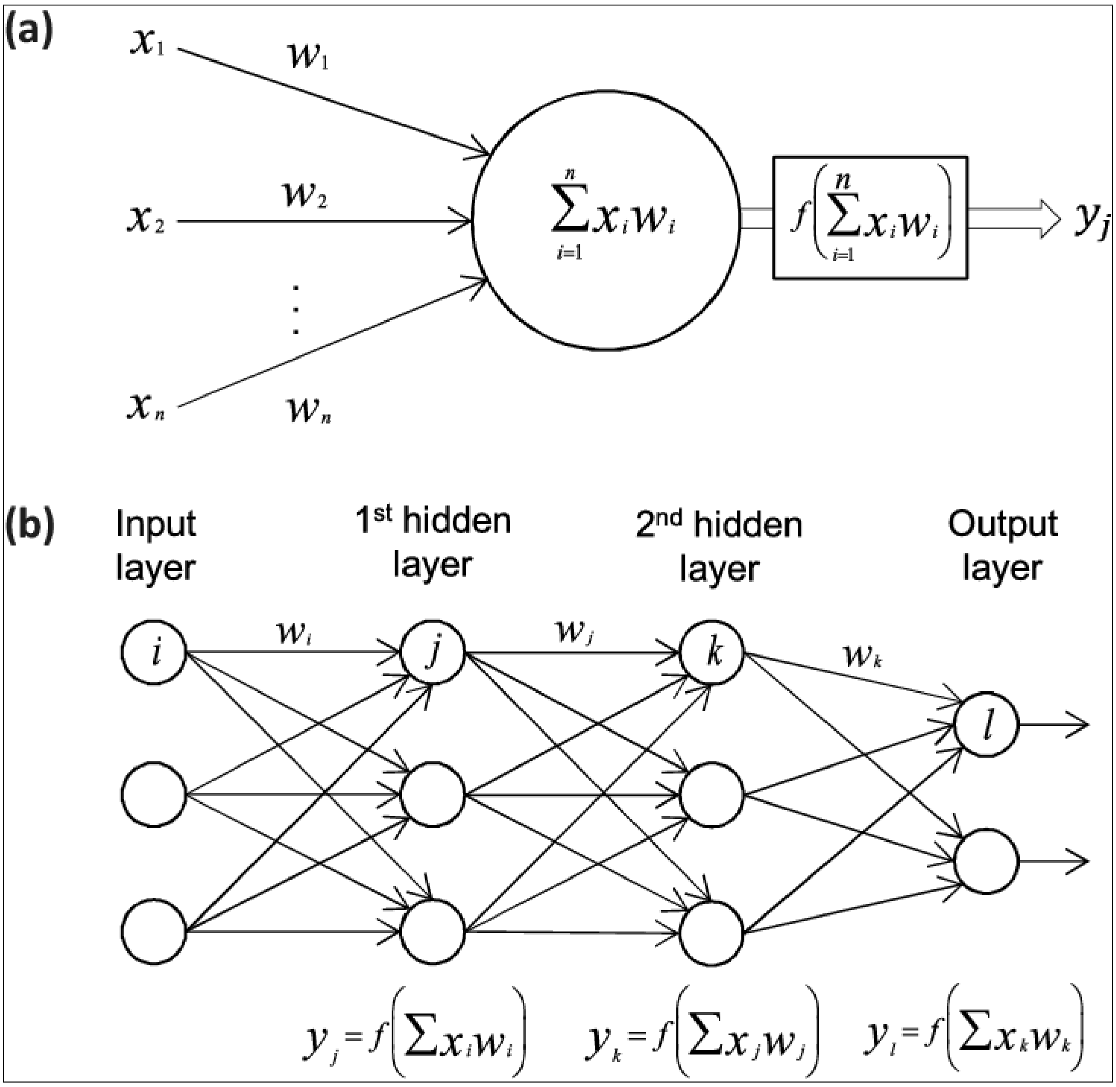

Figure 4.2: Working of Neurons

are the nodes known as neurons. . The simplest layout of an NN with a single hidden layer is this arrangement of layered units, with inputs connecting to inner neurons that are connected to the output node. Neural networks are made up of a combination of neurons set in layers. Every neuron is a mathematical operation that takes values as inputs, multiplies them with their weights then sends the sum through an activation function to another neuron. Due to the fact that they don't directly interact with input or output data, these inner neurons are referred to be hidden. Deep neural networks are any neural networks with more than one hidden layer. Our mathematical model requires a regulating function to describe the activity of the neuron node's activity. The (Relu) is the most modern and cutting-edge option, whereas the sigmoid function is the most historically suited for this purpose. Hyperbolic tangents are another possibility for the activation of a neuron. The graphical representation of these functions is represented in the figure. In this study, it is used for the prediction purpose. A universal approximation theorem establishes the ability of neural networks to adequately represent any continuous function. In Michael Nielson's book, it is described through graph of how to appropriately approximate the the continuous function. Mathematical equations are also mentioned in Cybenko and Hornik's work.

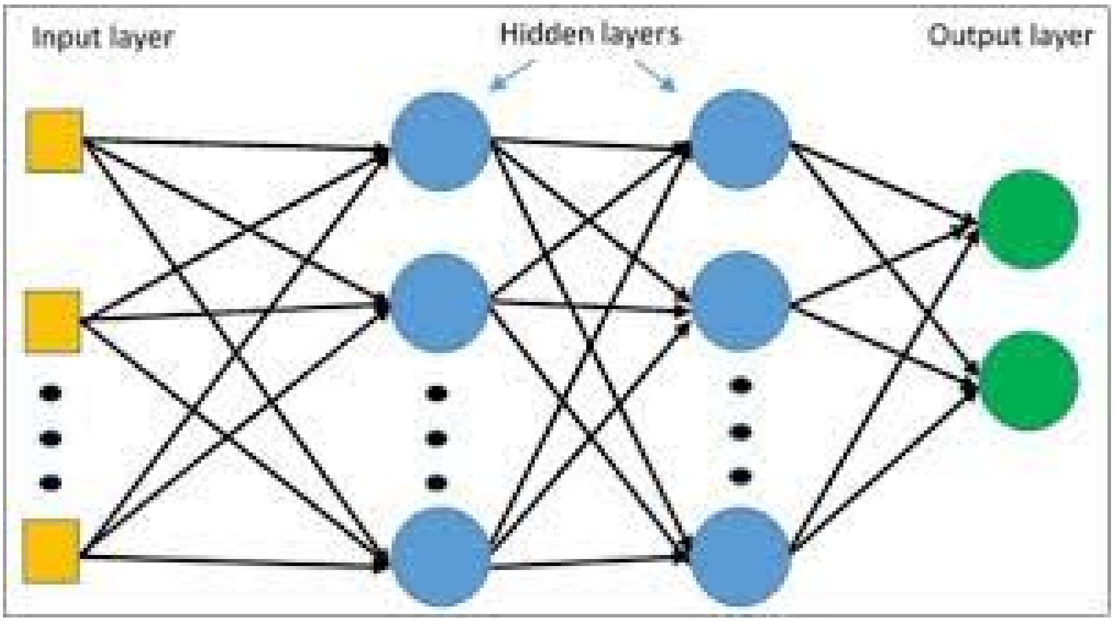

Figure 4.3

Think about a NN with a single hidden layer made up of 10 neurons. Five neurons make up the input layer, but just one neuron makes up the output layer. The NN model is as follows:

$$N = \sum W_{2i} a_i + b_2 \tag{4.3.1}$$

$$a_i = \sigma \left(\sum W_{1j} X_j + b_1 \right) \tag{4.3.2}$$

In the above equations, W_1 and W_2 are the weights associated with the connection of nodes. b_1 and b_2 are called the bias associated with the neural network. W_1 are the links between the IL to the first HL are weighted. W_2 are the weights associated with the connections from the first HL to the OL. Form of W_1 implies (5, 10) as a matrix since there are 5 inputs and 10 hidden neurons. The form of W_2 is (10,1) as a matrix since there are 10 hidden layer neurons and 1 output layer neuron. The network consists of 62 parameters adding two bias values. Since random weights are used to establish neural networks, the output is initially random. The final output comes from the universal approximation theorem. The model that emerges when a neural network learns from data is stored in the nodes' activity. It follows that neural networks are opaque systems.

4.3.1 Feed Forward Nueral Network Mapping

It was the most basic one-direction network, sometimes known as a uni-direction network. With feed-forward, there is no need for any loops or cycles because inputs are simply taken, passed through hidden layer nodes, and then sent to output nodes. This network mechanism is very popular because the computation power is good as it calculates relatively fast and almost could solve any accountable calculations. Feed-forwards are more accurate in the prediction of things they work amazingly with historical data.

- Let's suppose there are two input values that are associated with ten weights and two bias functions and the system wants binary output. So initial biases will be B1=B2= 1 or 0. Here the system can fix the biases according to the nature of the data. A bias permits the network to either shift the ability to activate to the left or right. The first step is to find the product of weights and inputs in hidden nodes.

$$H1 = I1W1 + I2W3 + B1W5 \tag{4.3.3}$$

$$H2 = I1W2 + I2W4 + B1W6 \tag{4.3.4}$$

4.3.2 ACTIVATION FUNCTION

Similar to the nervous system, every neuron in the DNN is related to each other neuron in the layer below it[1]. A neuron's interactions with other cells are all given weight. The forecasters serve as inputs for the IL, multiplied by the number of corresponding weights. The total of all these factors, in addition to the bias, is then delivered to the neurons of the first HL, serving as an input for the layer below's activation function. In order to convey a signal to the following layer, a node must have sufficient information, which is determined by the activation function. Rectified linear unit function (ReLU) is utilized as an activation function in the current work[23] for HL. The formula for the RELU is:

$$f(z) = \begin{cases} 0, & for\ z < 0, \\ z, & for\ z \geq 0 \end{cases} \tag{4.3.5}$$

The output layer in the current investigation has a linear activation function that is defined as follows:

$$f(z) = z \tag{4.3.6}$$

DNN selects a collection of samples collected during the forward pass, modifies the Wt used to connect neurons, and forecasts. The effectiveness of the DNN is assessed utilizing the loss function and the chosen performance metric. DNN computes the slope of the loss function in relation to the weights as it scans the layers in the backward pass. Back-propagation is the name of this technique. Back-propagation requires the goal function and optimization procedure, two parameters.

4.3.3 OBJECTIVE FUNCTION

The aim function, also referred to as the loss or error, is minimized during training. The objective function calculates the difference between the forecasted and actual values error. Mean-square error (MSE) is employed as the loss function in the current investigation. The MSE loss function's mathematical formula is:

$$MSE = \frac{1}{M}\sum_{j=1}^{M}(x_j - \hat{x}_j) \tag{4.3.7}$$

where x_i and $\hat{x}_i$ represent both the actual and expected response values for the j_{th} sample respectively. M is DNN employs a sample size of.

4.3.4 OPTIMIZATION ALORITHM

The root mean square propagation learning algorithm (RMSprop) is used to train the current DNN. The RMSprop approach works by keeping a moving average of the most recent mean square gradients of a certain weight[15].

$$MS(w,t) = \gamma \times MS(w,t-1) + (1-\gamma)(\nabla E(w))^2 \tag{4.3.8}$$

$$RMSprop : w_{new} = w_{old} - \frac{\alpha}{\sqrt{MeanSquare(w,t)}}\nabla E(w_{old}) \tag{4.3.9}$$

The default value of $gamma$ is set to 0.9, w_new is the updated weight, w_old is the prior weight, E is the output loss determined by the objective function, $alpha$ is the learning rate, and t is the time step. The learning rates in the current study are set at 0.001, 0.002, and 0.005 for various DNN architectures.

4.3.5 EPOCHS AND BATCH SIZE

An epoch is the number of full forward and backward iterations (forward and backward passes) that the entire training data set goes through while going from the model. The amount of observations the model processes before making weight adjustments is known as the batch size. Epochs are set to 100 in the current study, whereas batch dimension is at 32.

One of the problems with neural networks is that while the loss error is small when the model is trained on training data, the model's accuracy is not, It is significant when the same network is fed testing data. Overfitting is the term for this. Validation split and dropout regularization are two strategies that can be used to address the overfitting issue.

4.3.6 VALIDATION SPLIT

Our model has been modified with the split validation parameter to lessen overfitting. The validation split in the current study is set at 20The 60training set and 20validation set are separated from the 80 percent training dataset by the validation split. The DNN model is trained using a 60training set, and throughout training, both training and validation loss are tracked for each epoch. Internal validation technique is what this is known as. The DNN model's training and validation losses should ideally converge.

4.3.7 DROPOUT REGULIZER

Another method to lessen overfitting and increase generalization is dropout[16]. It essentially causes some neurons in a layer to have their outputs set to zero during training, reducing their contributions to neuron activation in the subsequent layers during the forward pass, and not updating during the retrograde pass, the weights of these neurons are increased. To avoid intricate co-adaptations among neurons, this is done[15]. Dropout rates are set at 40for the third hidden layer, 30for the second hidden layer, and 20for the third hidden layer in the current study.

4.3.8 METRIC

(MAE) is employed as the performance statistic to assess how well the DNN model performed during the training process. . Similarly to the loss function track record, the MAE of the training and validation sets is also tracked.

4.4 Neural Network

Neural networking is called the mimicry of human brain system and it is a unit in neuroscience. Back in history, the people used computers in which machines do only those tasks that was mentioned to do but due to neural networking now machines are able to work like the human brain works, how the human brain commands, operates different tasks, and much more. Neural networking due to its capability of doing tasks like the brain has many benefits, people are using it in many fields like in medical the professionals are using it in neuroscience and solving many neurological complications and it helps to do processes smoothly as study knows neurons are complexly attached with each other[13]. A NN is a compound set of input layers, an output layer, hidden layers, weights, neurons, etc, and these units or layers are combined in an exact way that copies the human brain[9]. In neural networking, the study has various algorithms to find the underlying relationships in a data set and have a set of neurons that could be either natural or artificial[7]. Classification of neural network models relies on the nature of the data set. According to the nature of the data collection it has (ANN), (CNN), (RNN), and more[4].

4.4.1 Understanding Of NN

The basic concept of NN is discussed below. As we mentioned earlier in neural networking system works like a human brain working.

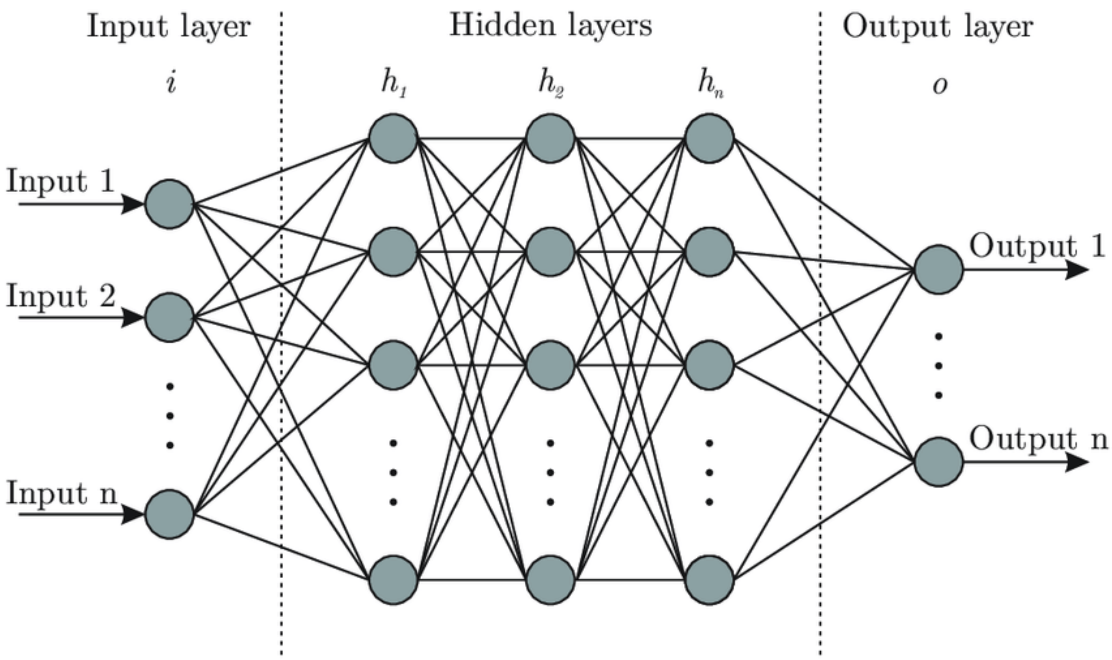

Figure 4.4: This figure illustrates the structure of a simple neural network

In the given 4.4 a straightforward neural network's structure can be seen. This network depends on two inputs which are x1 and x2 then these inputs go to the hidden layers but in between they had some weights and some activation functions which activated the function to further procedure. This model is a single perceptron model because it has a single hidden layer with two nodes which are h1 and h2. And in last we have our output value which is represented by o1. It is very different from the computers like computers can not mimic the working of the brain but neural networking is mimicry of brain work. Neural networks are made up of basically three components which are input which system gets from its features and also called them independent variables because inputs are always based on system's desire there are neural network's structure consists of hidden layers, which make up the majority of the network, and outputs, which are the conclusions or results that the network derives from the process. . The whole process seems simple but this is not as simple as it looks there are a lot of components between them like there are some weights attached to them, and some activation functions work here to further proceed the process, then there is a bias which is usually 0.5 but it also differs according to the wish and system, some transfer functions which transfer the information through nodes, etc. NN has a combination of different neurons which are attached through different interconnected nodes and make a layer of those. So by attaching the nodes the system makes a network that works properly. In networks Inputs layer has one or more neurons depending on desire like the weight of a person is a single neuron input but the height of all the students is not a single neuron input same as for output neurons they are mostly one but could be more than one based on specific needs, it is known as a logistic outcome if two outputs are present for example with height input the system wants to predict the IQ level or age of the students [2].

The main process part is where the network has hidden layers. Hidden layers could be one, two, three, or more than these but there is one concept for that when there is only one hidden layer in the system then that is called a single perceptron But if there will be two or more than two hidden layers then call it multi-perceptron and they used to apply a non-linear transformation with the help of activation function and weights which adjust on the independent variables. Further, will see the concept and implementation of weights mean how to implement the weights on the network and also how to adjust those weights to minimize the error rate, activation function, etc[10].

4.4.2 Working of Neurons

Neural networks are made up of the combination of neurons set in layers[3]. Every neuron is a mathematical operation in which takes values as inputs, multiplies them with their weights then sends the sum through an activation function to another neuron. Here is the sample of mathematical computation of a neuron by which it gets the output.

Here in equation bj is biases of the function which are very important because in modeling of fitting the data weight goes up and down and if there is bias then there is no need for a line to pass through the origin and it is same like the intercept in a regression model. It gives space for the activation function to go either in an upward direction or in downwards and also it provides flexibility which is good in machine learning. In the neural network, there is an assigned magnitude of error with the calculation of weights and inputs which is called the cross entropy loss function. The objective to have the least error so that the accuracy for the model should be high. The model's accuracy just depends on its error rate so try to minimize the error rate as much as system can.

In given 4.5 the working of neurons has been shown. There is a single input layer and in it, there are three input neurons which are x1, x2, and x3 which take values then the arrow between them shows the activation functions which are working for the process there are three weights w1, w2, and w3. firstly, the first neuron which is x1 goes with the weight and bias to the hidden layer then moves towards the output then again the first neuron is now attached to the second weight and again goes to the end. And this process goes for the second input neuron and then goes for every neuron in the network. And in last system gets the error rate for the networks.

The main objective is to build a NN that minimizes cross-entropy loss function with the help of adjustment of weights and LR. Learning rate is a tuning parameter that ranges from 0 to 1 and tells each iteration's step size or epoch although cross-entropy loss function is becoming minimized by the network [5]. There are different algorithms for different tasks.

4.5 Recurrent Neural Network

Predicting measures in total derived from vector sequences of varying lengths and "time" components is essential in ML. RNNs are able to learn historical dependencies. Previous-term reliance is of utmost importance whenever forecasting weather conditions. From that standpoint, is valuable because it is an artificial neural network that is typically used for developing predic-

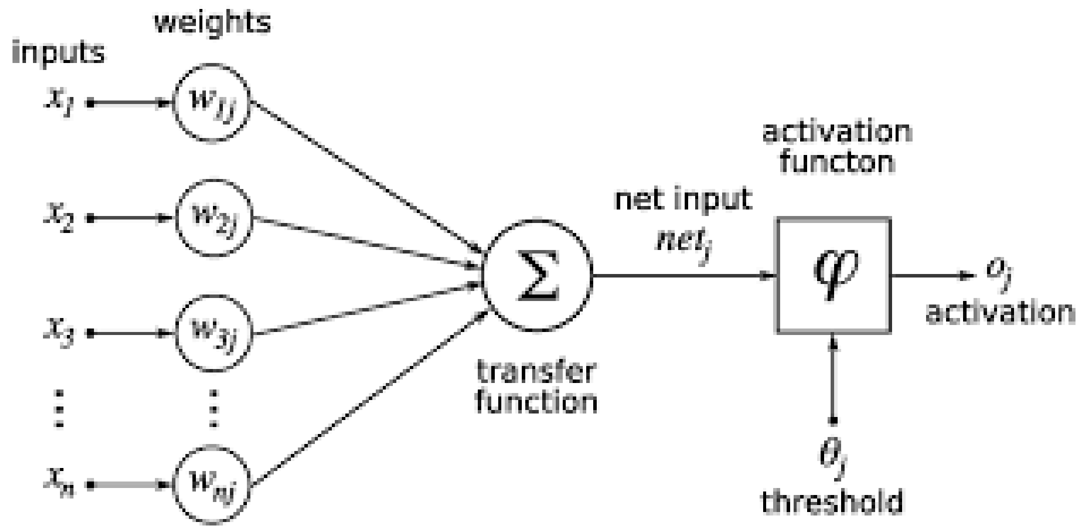

Figure 4.5: This figure illustrates the working of neurons in neural network

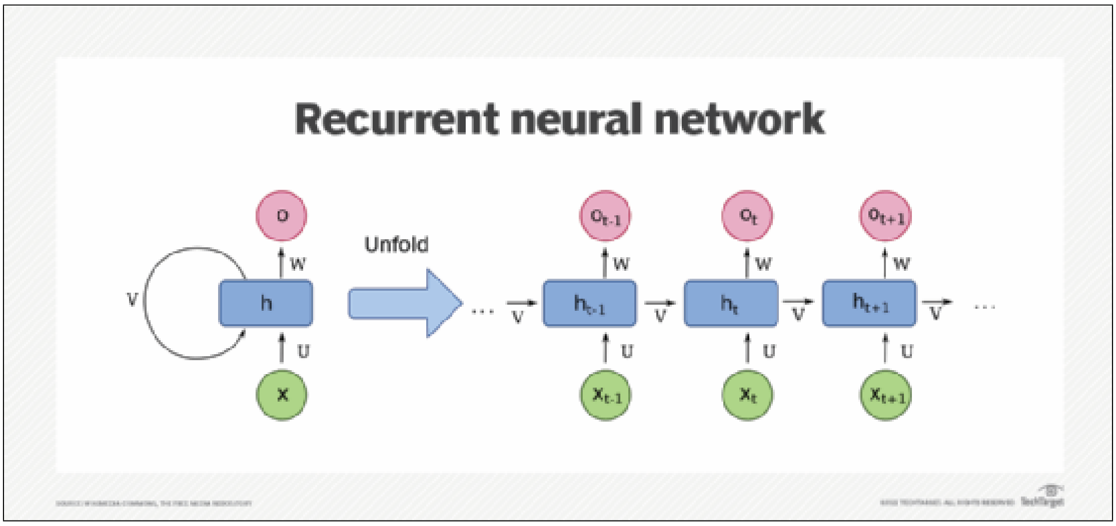

Figure 4.6: Structure of RNN

tion networks utilizing long-term time series datasets. At time t, the generic RNN predicts the following:

$$H_t = g(W_h[H_{t-1}], X_t + b_h) \tag{4.5.1}$$

$$O_t = f(W_o * [H_t], +b_o) \tag{4.5.2}$$

The figure depicts the standard structure of an RNN. 4.6: Where W_h and W_o are applied to the HL and OL, b_h and b_o is the opinion for the HL and OL, g is the activation function used in the HL, and f is the forecast output function. Ht is the hidden state. Ot is the predicted output. The RNN's prediction performance is mostly determined by the activation function, which forecasts output depending regarding outcome of the hidden state and employs the current input and prior state as input vectors. This model has been used in numerous research, and it has undergone alterations that are described in the section on related work.

Elman networks are classified as RNNs and can contain a single layer or multiple levels of concealment . RNNs were developed to collect temporal contextual data together with time series data. RNNs have feedback loops incorporated into their architecture, in contrast to traditional FFNNs.

The formula below illustrates the error in a neural network.:

$$\Delta = \frac{\partial E}{\partial W} \tag{4.5.3}$$

RNNs are frequently trained using the (BPTT) method, which continually back-propagates to a specified number of states n to change the network's weight in an effort to reduce error. In order to account for the change in weight, the sum of the errors with corresponding weights is found as follows:

$$\Delta = \sum_{i=1}^{n} \frac{\partial E_t}{\partial W_t} \tag{4.5.4}$$

$$\Delta = \frac{\partial E}{\partial W_1} + \frac{\partial E}{\partial W_2} + ... + \frac{\partial E}{\partial W_n} \tag{4.5.5}$$

Later time step errors are difficult to detect to revert to a previous occasion step with the necessary network settings adjustments. The LSTM unit was developed to address this problem by significantly reducing each stage's loss.

4.6 Roll of Activation Function

As was said in the section prior to this one, activation functions are the main determinant of forecasting and overall network performance. Selecting an activation function for the suitable application is therefore a challenging challenge. A disparity in weight changes is initially significant because of the high learning rate experienced during the training phase. The weight eventually approaches 0, indicating that there was no difference for the change in input in output, and learning then slows down. The relationship between weight update and learning rate at t:

$$W_t = L_t \times E_t \tag{4.6.1}$$

While bias values used on the network discover an optimum match in relation to the curve together with relation to the data by moving, weight tuning aids neural networks in reshaping the steepness of the curve. Let's go over the advantages and disadvantages of each activation function in more depth. A common function for artificial neural networks' hidden layers is sigmoid, which maps any input value range between 0 and 1. A significant shift in input is reflected as a modest shift in output since the operation condenses even with a wide variety of input values. Therefore, this function, which has the following representation, can be utilized in applications that require computing probability as an outcome. This function can therefore be used in scenarios where the computation of:

$$f(Z) = \frac{1}{1+e^{-Z}} \tag{4.6.2}$$

A gradient of that function computed during back-propagation goes off exponentially, resulting in extremely modest differentiates and adjustments in the NN. When a gradient is disappearing, neurons often At some point, you should stop learning. The sigmoid function's derivative is:

$$\frac{d(f(Z))}{d(Z)} = \frac{e^Z}{(1+e^Z)^2} \tag{4.6.3}$$

Another essential feature is that the sigmoidal curve is centered on 0.5 rather than the origin. Because of this, transition-related computations on the curve are difficult, with the use of the *tanh* function is more complex as long as that of the sigmoid. The *tanh* activation function, referred to as the hyperbolic tangent, is superior to the *sigmoid* in a few ways. It is defined by::

$$f(Z) = \frac{e^Z - e^{-Z}}{(e^Z + e^Z)} \tag{4.6.4}$$

It minimizes the vr in comparison to the sigmoid because derivatives are strong and the hyperbolic tangent curve is steeper. After differentiating the above function we get:

$$\frac{d(f(Z))}{d(Z)} = 1 - (f(Z))^2 \tag{4.6.5}$$

As one delves deeper into networks, the vanishing gradient issue with the tanh function persists. As a result, in their look for a more effective activation function to address the gradient problem, many academics have recently utilized a (Rectified Linear Unit) activation function.

The previously mentioned Rectified Linear Unit function has the range that is [0 to] in contrast to the sigmoid and tanh functions. Even minor input values result in major output changes it converts all negative numbers to 0 and all positive values to Z. The differentiation of Relu is one for all Z values greater than zero and zero for all Z values less than zero; however, they never exist if Z equals zero.

This talent, enables Relu to mitigate vanishing gradients along with a higher learning rate for all Z values, enabling him to survive. There is, however, a problem with the Relu activation option. It should be observed that since all negative x values are set to zero, it is prevented from ceasing to learn, which will result in the neurons degenerating following a given total number of continuous inputs without a slope. Regardless of whether one of the input data tends to attain the slope amid several successive negative or zero values, learning could still take place in the network since neurons are still active.

4.7 PRPBLEM OF LONG TERM DEPENDENCIES

The ability of RNNs to connect prior information to the task at hand, for instance by using previous video frames to help interpret the present frame, is one of their attractions. If RNNs could do this, it would be incredibly beneficial. but are they able to? It differs.

Looking at the most recent data is sometimes all that is required to finish the task at hand. Think of a language model that tries to anticipate the next word by using the words that came before it. The final word in "the clouds are in the sky" can be deduced without the use of any extra context because it is obvious that sky will come after it. Since the needed region is small in this case, RNNs can be trained using historical data to account for the mismatch between the pertinent information and the relevant information.

But sometimes we need more background information. Try to figure out the last words of the statement "I grew up in France... I speak fluent French." The following word is most certainly the name of a language, according to current evidence, however in order to determine which language it is, we need more information about France's history. There is definitely a chance that there will be a very wide gap between the necessary information and the point at which it is needed.

Unfortunately, when that gap widens, RNNs become less and less capable of learning to correlate the data.

RNNs are, in theory, capable of handling these "long-term dependencies." A human might carefully select the parameters for these kinds of toy issues. Sadly, it appears that RNNs can't actually learn them. Thankfully, LSTMs don't have this problem.

4.7.1 RNN vs LSTM

Think about the situation when you need to edit some information in a calendar. An RNN applies a function to the existing data to accomplish this, entirely altering it. While LSTM just performs minor adjustments to the data through cell state-based addition or multiplication. This is how LSTM selectively forgets and recalls information, outperforming RNNs.

Now imagine that you want to process data that has periodic patterns, such as forecasting colored powder sales that surge around the Indian holiday of Holi. Looking back at the sales figures from the prior year is a wise move. Therefore, you must understand which information should be deleted and which should be kept for future use. Otherwise, you'll need a pretty sharp memory. Theoretically, recurrent neural networks appear to be effective in this. They are unnecessary because of their two drawbacks, exploding gradient and vanishing gradient.

To address this issue, LSTM introduces memory chunks known as cell states. One may consider the designed cells to have differentiable memory.

4.8 LSTM

It is a particular RNN architecture that was created with modeling temporal sequences in mind. LSTM is more accurate than traditional RNNs because of its long-range reliance. Error backflow is an issue caused by the RNN architecture's back-propagation algorithm [11]. An LSTM's basic structure is referred to as a memory cell since it explicitly remembers and propagates unit outputs over a number of time steps. To store data regarding temporal situations, the LSTM memory cell uses cell states. The research looked at, LSTM-based NN was trained to predict radiation in the backdrop using TS meteorological statistics. The problem of slop3 relates to the level of difficulty in understanding the long-term links in RNN structure from a mathematical perspective. An LSTM's basic structure is referred to as a memory cell since it explicitly remembers and propagates unit outputs over a number of time steps. To store data regarding temporal situations, the LSTM memory cell uses cell states.

An individual node's output is determined by its activation function. In LSTM networks, the activation functions sigmoid and hyperbolic tangent (*tanh*) are widely utilized. The suggested LSTM's real design for the input and forget gates is created using the *sigmoid* function, and the candidate vector for updating the cell state vector is implemented using the *tanh* function:

$$I_t = sigmoid(W_t[X(t), h_{t-1}] + b_t) \tag{4.8.1}$$

$$F_t = sigmoid(W_f[h_{t-1}, x(t)] + b_f) \tag{4.8.2}$$

$$O_t = sigmoid(W_o[h_{t-1}, X(t)] + b_o) \tag{4.8.3}$$

$$C'_t = tanh(W_c[h_{t-1}, x(t)] + b_c) \tag{4.8.4}$$

$$C_t = F_t \cdot C_{t-1} \cdot I_t \cdot C'_t \tag{4.8.5}$$

$$h_t = O_t \times tanh(C_t \tag{4.8.6}$$

in that place W represents the weight, the bias is represented by b, ht1 is representative of the previous state hidden vector, and X(t) represents the input vector for each gate. **??**.2 shows the LSTM network's core structural representation.

4.9 Intensified LSTM

The true LSTM architecture, as indicated in the prior section, incorporates the *sigmoid* and *tanh* functions, albeit there are a few difficulties with these activation functions. These difficulties were previously discussed in the earlier section. It is critical to retain the gradient at a specific degree of back-propagation to prevent the problem of vanishing gradients, which keeps neurons active during the training phase of learning.

The actual LSTM employs three sigmoid functions—the forget gate, the input gate, and the output gate—as well as two tanh functions—the candidate vector and the output gate—but multiplying the input value with the forget gate and output gate is ineffective because the forget gate

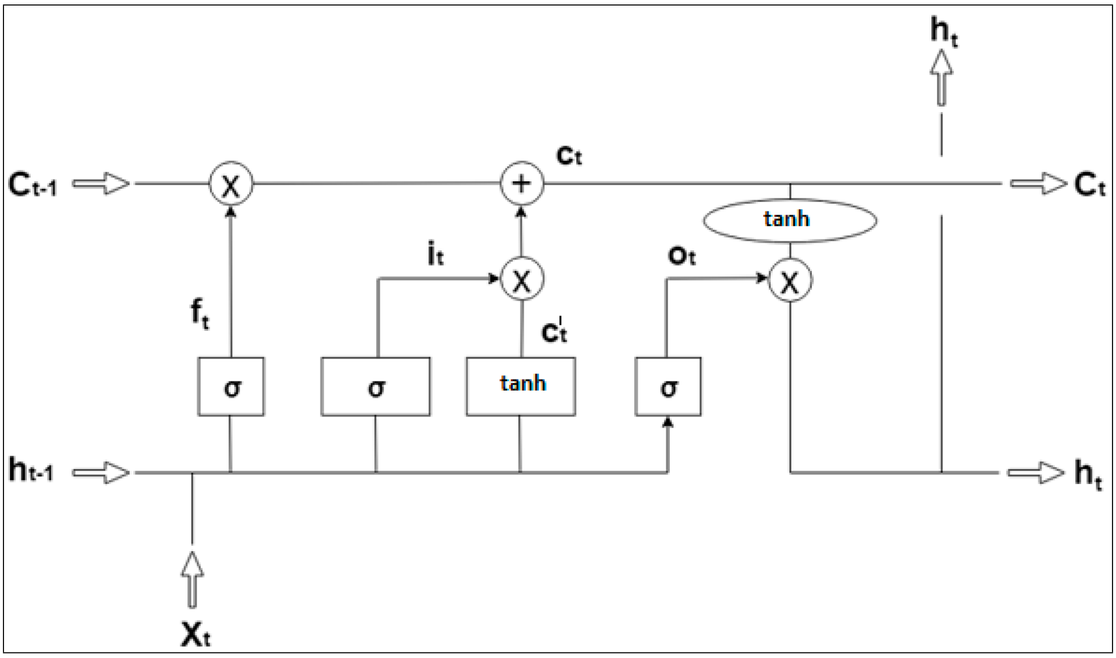

Figure 4.7: Representation of LSTM

determines whether or not to retain the current input and the output gate produces the predicted value, which has already been processed using cell state information. The input gate and candidate vector are obviously significant in updating the cell state vector, from which the LSTM learns and conducts analysis to anticipate the output value based on the current input.The input value is then multiplied by the sigmoid function in the input gate and the tanh function in the candidate vector. Figure 3.2 depicts the proposed Intensified LSTM structural representation. It is clear that the LSTM structure is more sophisticated in capturing the recursive relationship between the input and hidden layer. The output ht can be accessed in the same way as a traditional RNN can.

The forget gate evaluates whether or not the current input should be saved in the cell state by examining the impact of the current input on the prior state (data is saved via input and candidate vector). When not needed, the sigmoid function returns 0, deactivating the forget gate; when needed, it returns 1, activating the gate, as illustrated in Equation (3.4.1). Based on this selection, all of the other gates in the LSTM are active or disabled for a certain input.

$$f(t) = \left\{ \begin{array}{cc} 1 & \text{store the data} \\ 0 & \text{discard the data} \end{array} \right\} \tag{4.9.1}$$

When it is concluded that the current input should be retained, the network attempts to under-

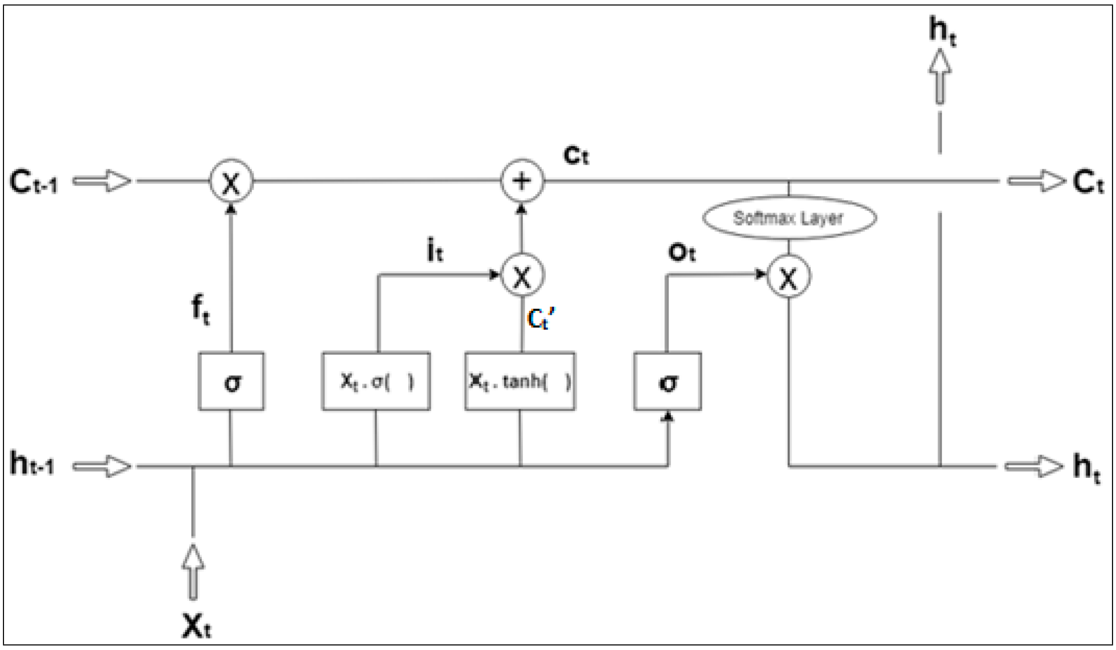

Figure 4.8: Schematic representation of Intensified LSTM

stand the new information brought over by comparing it to the previous state. Based on this, the input gate's sigmoid function is first examined, supplying a value between 0 and 1 indicating the range of additional information available in the current input. The outcome is then multiplied and reported, as shown in Equation (3.4.2). As a result, the input gate's value range is now [0, infinite], which prevents the gradient of the input vector from fading for the same range.

$$I_t = X(t) \times sigmoid(W_t[X(t), h_{t-1}] + b_t) \tag{4.9.2}$$

The current input and prior hidden state are used to activate the $tanh$ function, yielding between 1 and $+1$ indicating the current cell state vector data to be updated. As in the Equation, to provide a candidate vector between $[infty, +infty]$. This enhances learning since it allows the cell state vector to account for a little change in the time series.

$$C_t' = X(t) \times tanh(W_c[X(t), h_{t-1}] + b_c) \tag{4.9.3}$$

The approach described above slows propagation. Furthermore, A soft floor for the weights serves as implicit regularises by prohibiting weight learning for large magnitudes. Therefore, a gradient network is constantly present to minimize data loss and inaccuracy while learning new information. SoftMax regression is used to compute the probability of the distribution of a single event across several occurrences. An estimated probability distribution can then be utilized to determine the desired output for the supplied inputs. Mathematically is it shown as:

$$O_t = softmax(W_o[h_{t-1}, X(t)] + b_o) \tag{4.9.4}$$

$$h_t = O_t \times softmax(C_t) \tag{4.9.5}$$

All of the processed found values are computed by the output layers, and it continues until the EV of the concealed outcome equals 0. As a result, the prediction policy keeps a big record of historical rainfall as a reference. The storage of a big amount of data could have a substantial influence on forecast validity. Losses are reduced as a result of the enhanced learning rate produced by the time reduction. Because of the learning rate and losses, the RMSE number lowers automatically. Dry days are regarded as 0's in the precipitation dataset. The RMSE generated for precipitation prediction is greater when compared to other forecast models that account for elements such as pressure, temperature, and so on, as a result of the inclusion of zeros in the precipitation. However, Intensified LSTM has a smaller RMSE than earlier validation models. In addition, an optimizer is employed to enhance forecast accuracy.

4.10 Evaluation Index

We chose the(RMSE) to assess the model results since it was necessary to forecast the precipitation. The distinction between expected and actual figures can be reflected by the root mean square error.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1} m\,(x_i - \hat{x}_i)\,2} \tag{11}$$

In the above equation, the model prediction value is represented by xi, the actual value is represented by xi, and the projected sample size is shown by n. The formula shows that the RMSE value decreased in proportion to how closely the projected value matched the actual value, showing that the model prediction was more accurate.

CHAPTER 5

Results and Conclusion

The analysis for the 5 stations in this research was performed using one statistical model i.e. ARIMA and one machine learning model i.e. LSTM. Below are the outputs and results obtained from these models.

5.1 Results from Arima model

The time series was used for the stationarity and white noise tests. If the series was non-stationary, a differential approach would be used to convert it to a stationary sequence. If it was a white noise sequence, it means it carried no relevant information and had no modeling meaning..

5.1.1 Non Stationary data into Stationary data

After checking the stationary of all the regions we have found the data was non stationary is mentioned in the following figure 5.1.

So we first took the log and then the difference of the data then we found 5.2.

5.1.2 Forecasted precipitation by using the best fit ARIMA model AND the Residual Plots

In the figures 5.3,5.5,5.7,5.9 and 5.11, it can be seen that Arima model can easily and smoothly predict the future values for as many months as you wish. In the figures 5.4,5.6,5.8,**??** and 5.12, we can see that the residuals for Arima are spread over the plane with their mean equal to zero, the ACF plot shows the lagged values for each Arima model. Box. Ljung test aids the regression results as it verifies that there is no trend in the series.

5.2 LSTM

5.2.1 Data Processing

The dataset is loaded from a CSV file. 'MM/DD/YYYY' format is be used to convert the 'Date' column to datetime format. Resize the precipitation data to have a single feature and convert it to a numpy array. To get the data into the [0, 1] range, apply Min-Max scaling.

5.2.2 Model Trraining

Created a function called train-and-predict-city that accepts a dataset and a city's name as inputs.

Filtered the dataset to only contain the information for the chosen city. Separate the data into training and testing sets, with the length of the training data set being set to "2012-01-12."

5.2.3 Model testing and Prediction

Created sequences of the precipitation values for the last 60 days (x-train) and their matching next-day values (y-train) to prepare the training data. Made a 128-unit LSTM model first, then a second 64-unit LSTM layer. Used the Adam optimizer and mean squared error loss while compiling the model. Utilize the training data to train the model.

5.2.4 Evaluation and Visualisation

Evaluation measures have been calculated, such as (MSE), (MAE), (RMSE), and (MAPE). The training data, precipitation values (both real and anticipated), and precipitation values were plotted. displayed the plot along with the projected values, validation data, and training data.

5.2.5 Reporting

You can use this code to train an LSTM model on historical precipitation data for a specific city and predict future precipitation values. The training and testing phases include data pre-processing, building the LSTM model, training it, and thenassessing the model's performance on testing data. The training and testing phases include data preprocessing, building the LSTM model, training it, and then evaluating the model's performance on the testing data set. To evaluate the model's accuracy, the algorithm creates charts that visually contrast real and anticipated precipitation values. Metrics for evaluation offer numerical measurements of how effectively the model predicts precipitation values. This code essentially shows how to employ LSTM models for time series precipitation value predictions for a particular city.

In figures 5.13,5.14,5.15,5.16 and 5.17, Blue color shows the training process involves feeding the training data through the LSTM model. During training, the LSTM modifies its internal weights to recognize patterns and dependencies in the input sequences. It continues for multiple iterations until the model converges or reaches a predefined stopping point. Following model training, it is evaluated on the set of test data which can be seen with the orange color and the green color shows the predicted precipitation in each area which is quite reasonable and smooth.

5.3 Model Performance of Arima AND LSTM

In table 5.1, we can that the values of RMSE and the values of MAE are much less, which means the error value is much less for all the five regions of UIB. And on the other hand table 5.2 gives us how well the lstm model performed using RMSE and MAE, it tells us that the error value for the regions is very high.

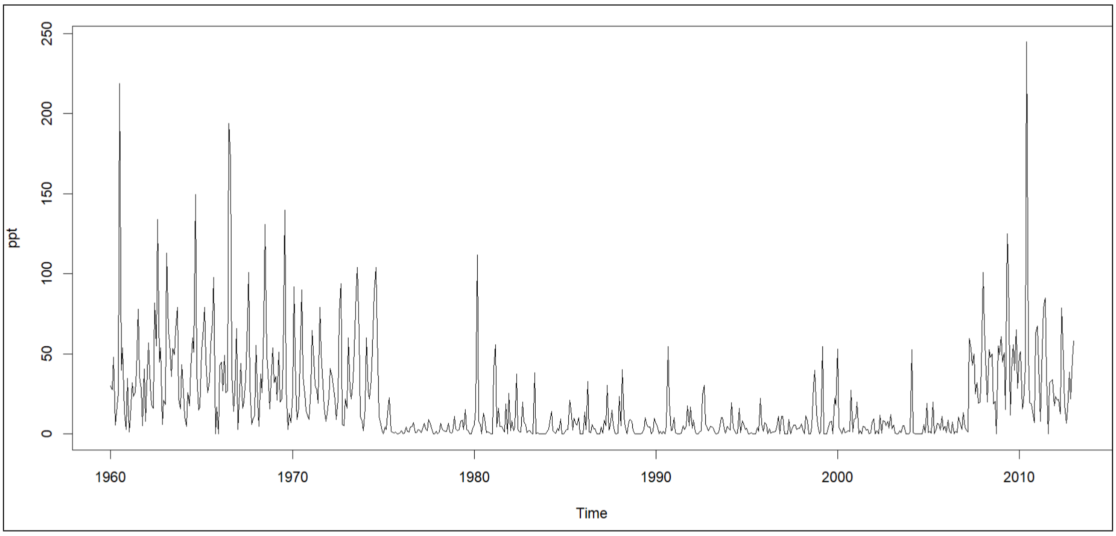

Figure 5.1: Non stationary graph

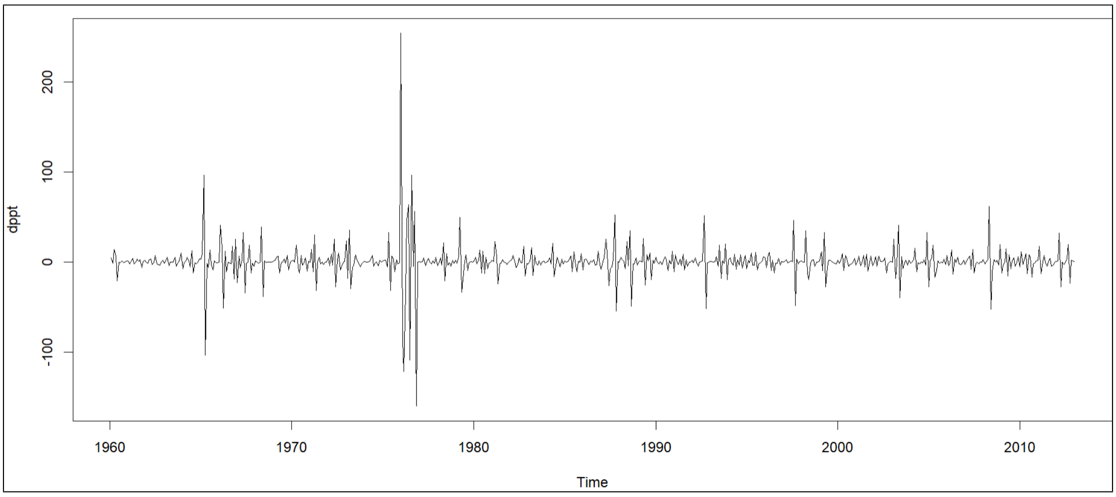

Figure 5.2: stationary graph

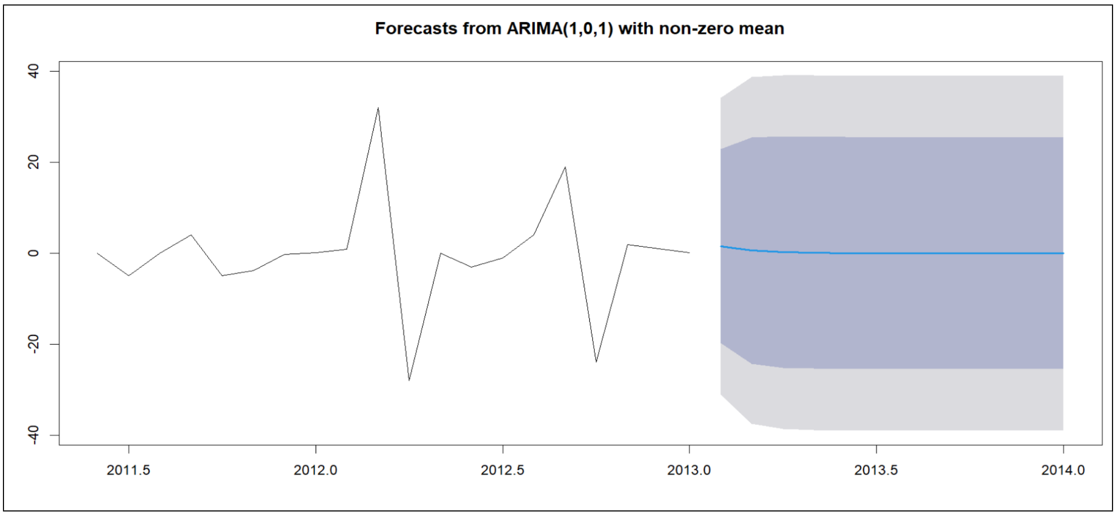

Figure 5.3: Prediction of Monthly Extreme Precipitation in Gilgit

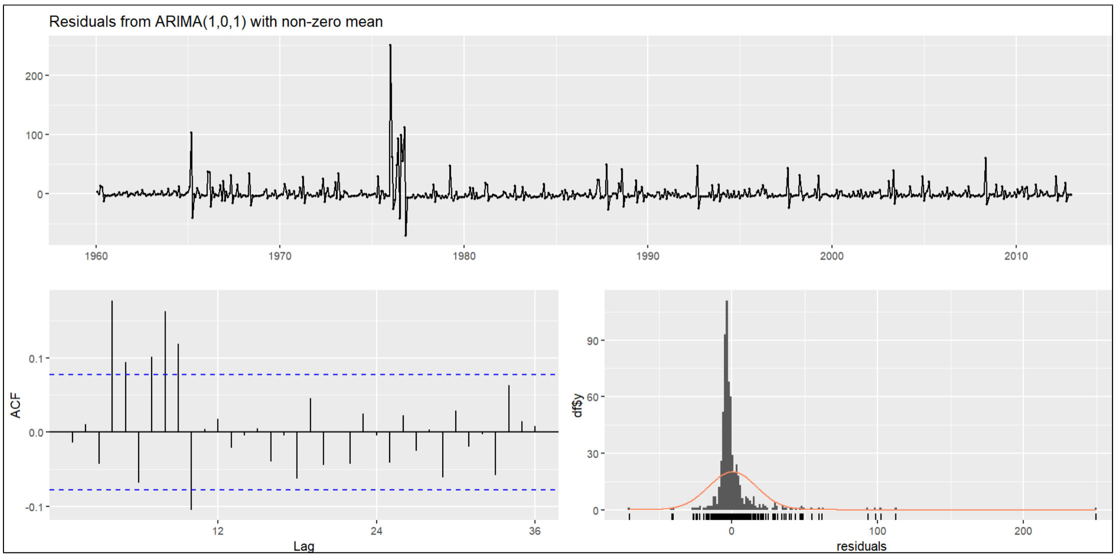

Figure 5.4: Residual plot for Gilgit Region

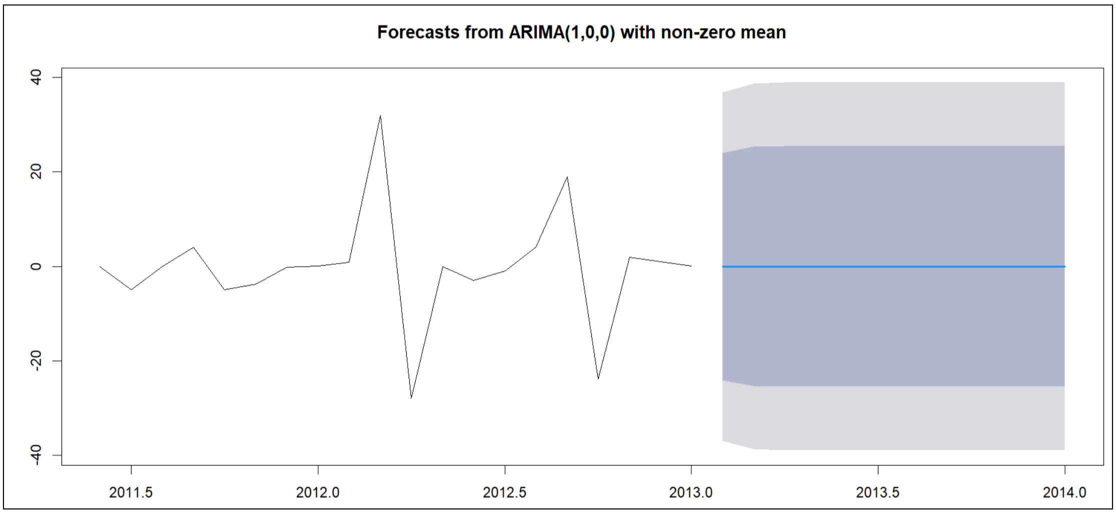

Figure 5.5: Prediction of Monthly Extreme Precipitation in Balakot

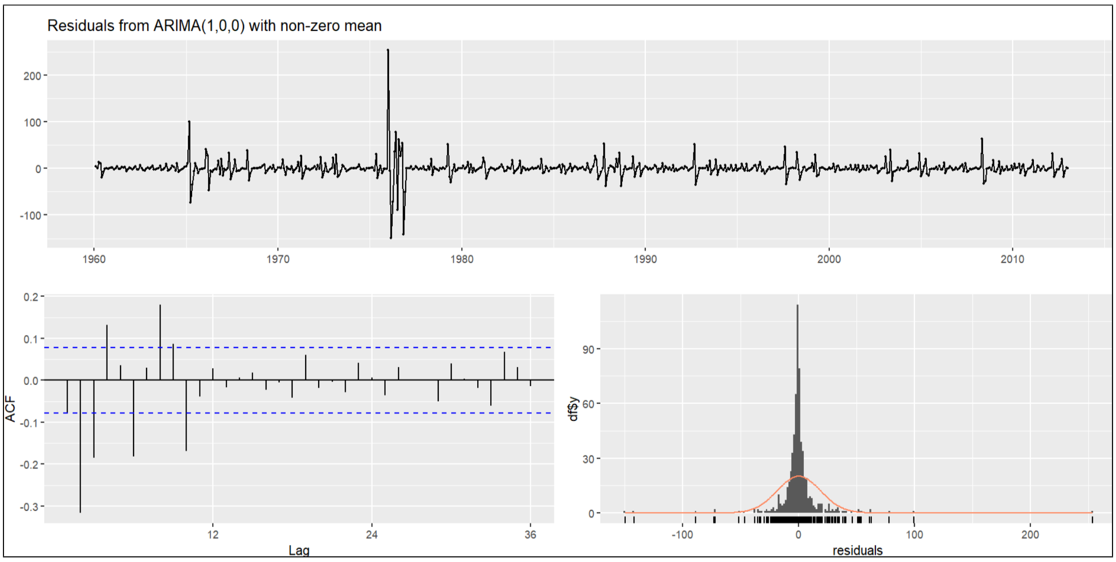

Figure 5.6: Residual plot for balakot Region

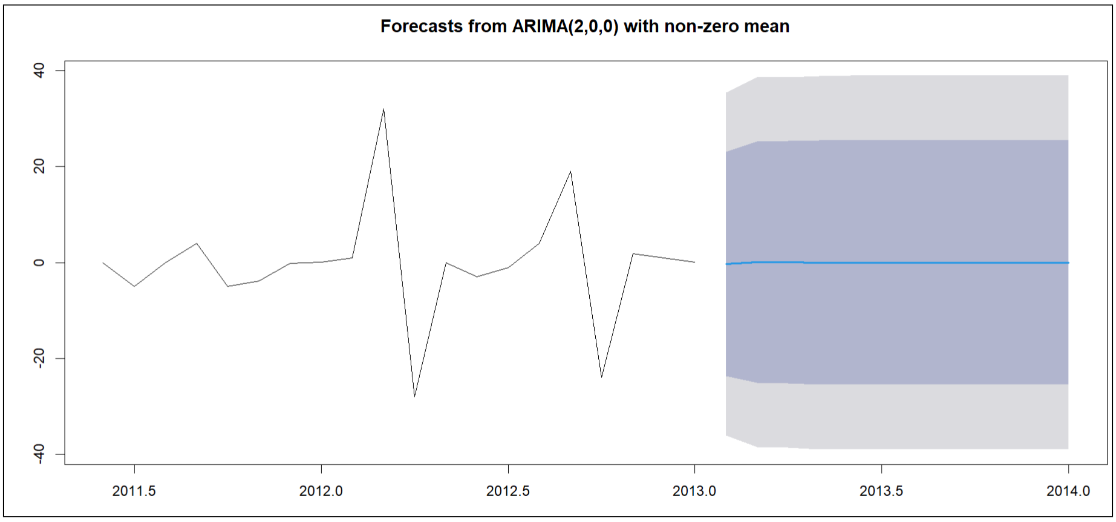

Figure 5.7: Prediction of Monthly Extreme Precipitation in Chilas

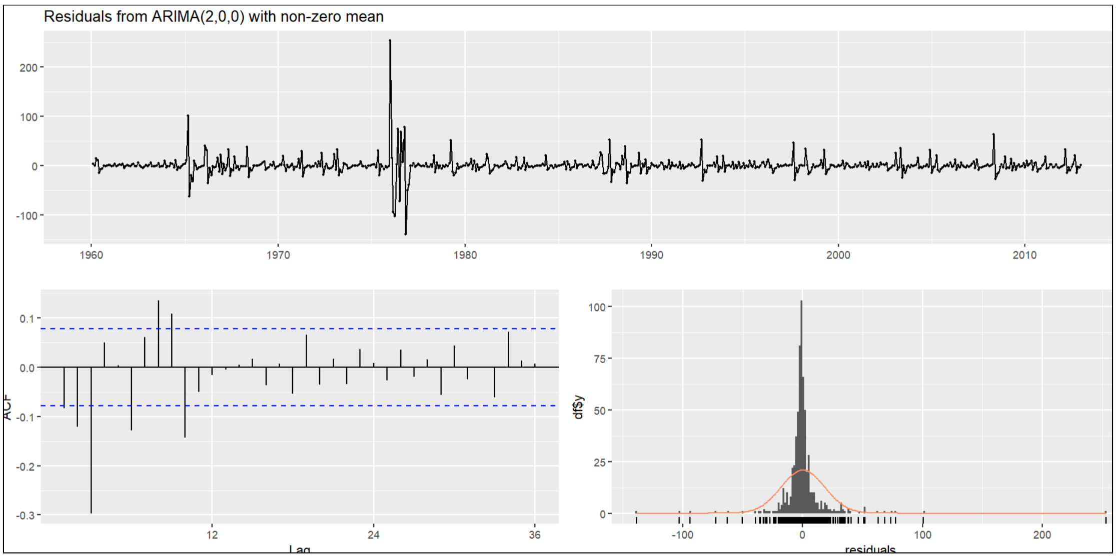

Figure 5.8: Residual plot for Chilas Region

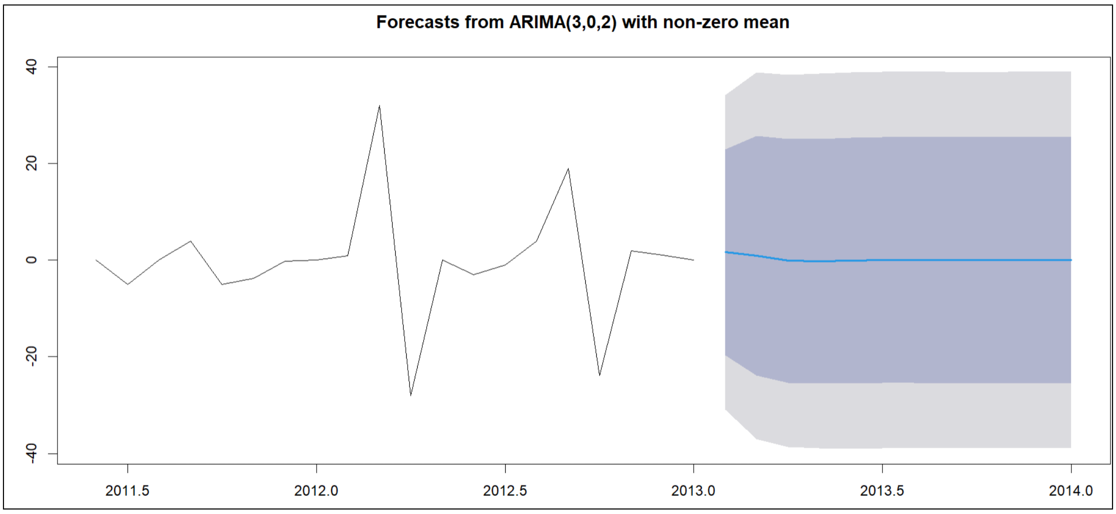

Figure 5.9: Prediction of Monthly Extreme Precipitation in Murree

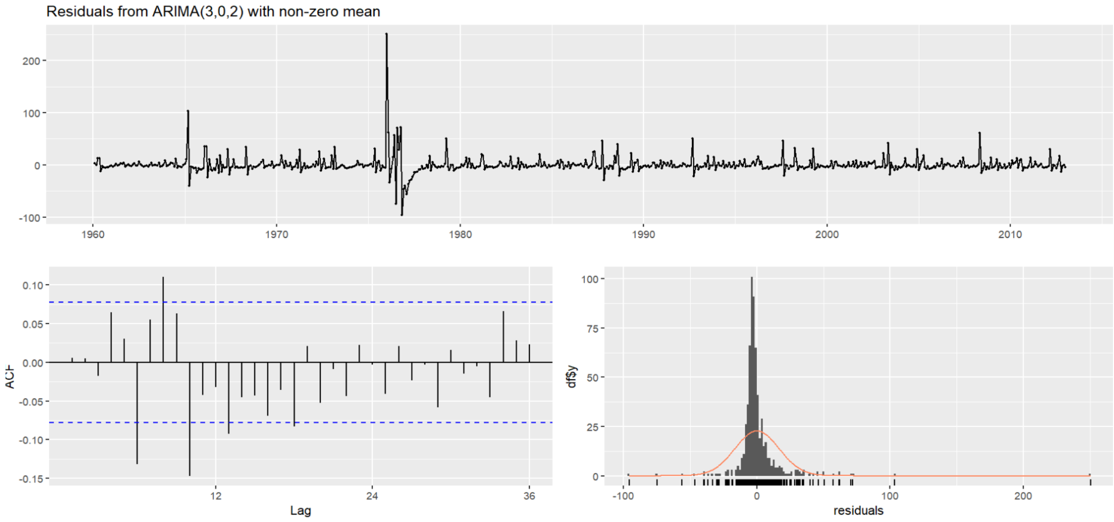

Figure 5.10: Caption

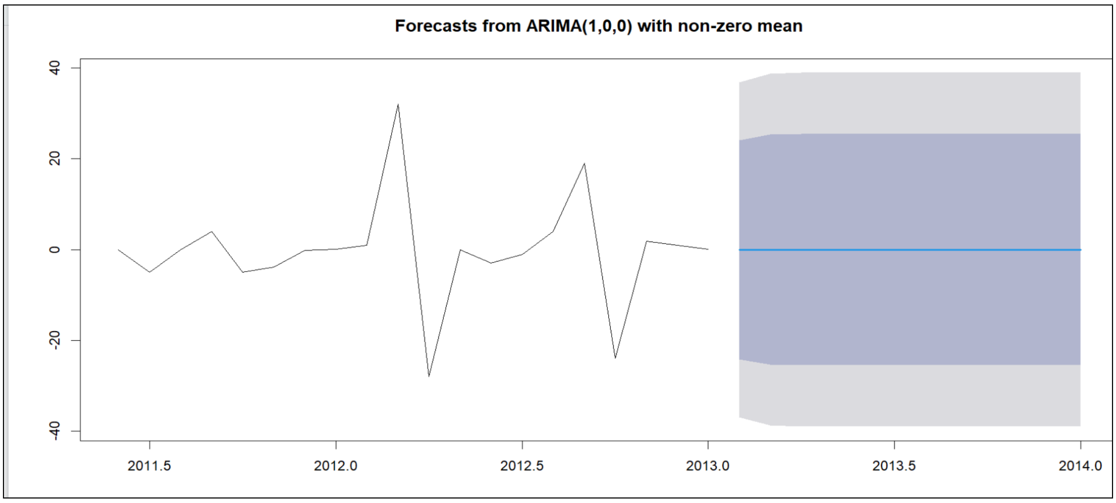

Figure 5.11: Prediction of Monthly Extreme Precipitation in Peshawar

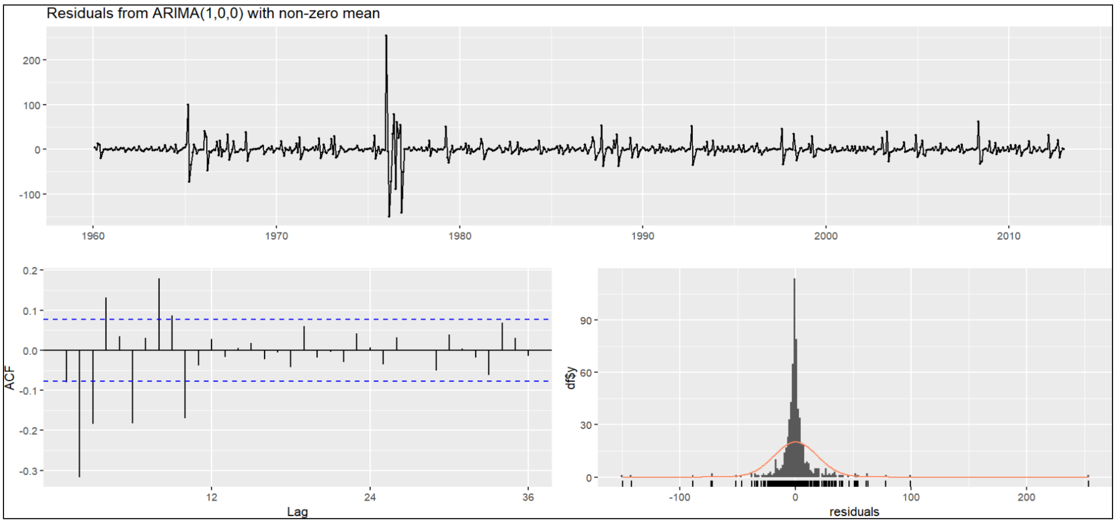

Figure 5.12: Residual plot for Peshawar Region

Table 5.1: The performance measures of the Arima are listed.

Regions	RMSE	MAE
Gilgit	0.009701972	7.725401
Balakot	0.025186871	7.144279
Chilas	0.4578095	7.344279
Peshawar	0.003274057	7.993098
Murree	0.002360108	8.103732

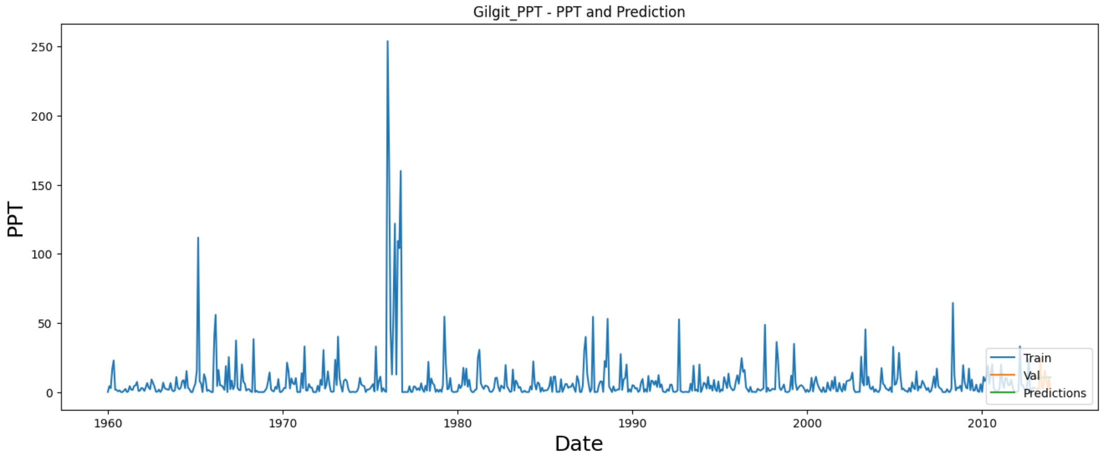

Figure 5.13: Prediction for Gilgit region using LSTM

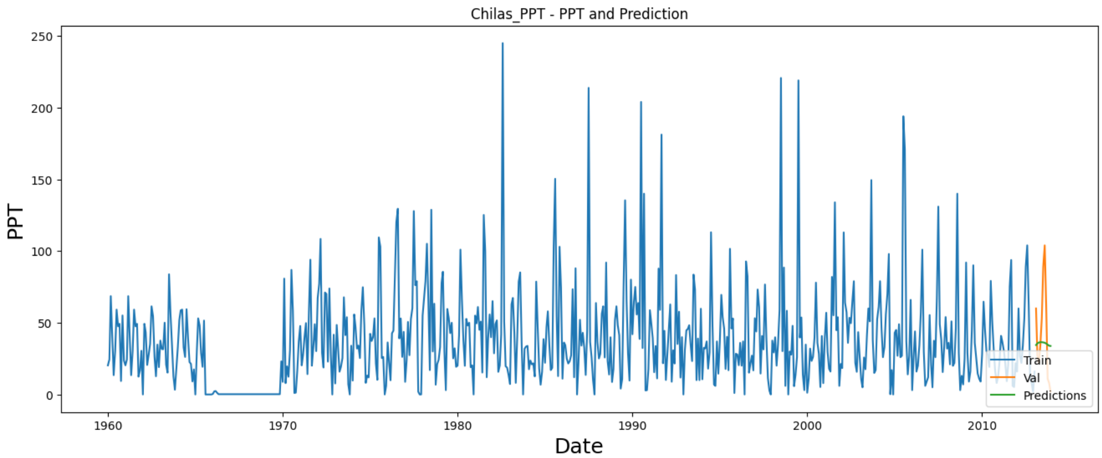

Figure 5.14: Prediction for Chilas region using LSTM

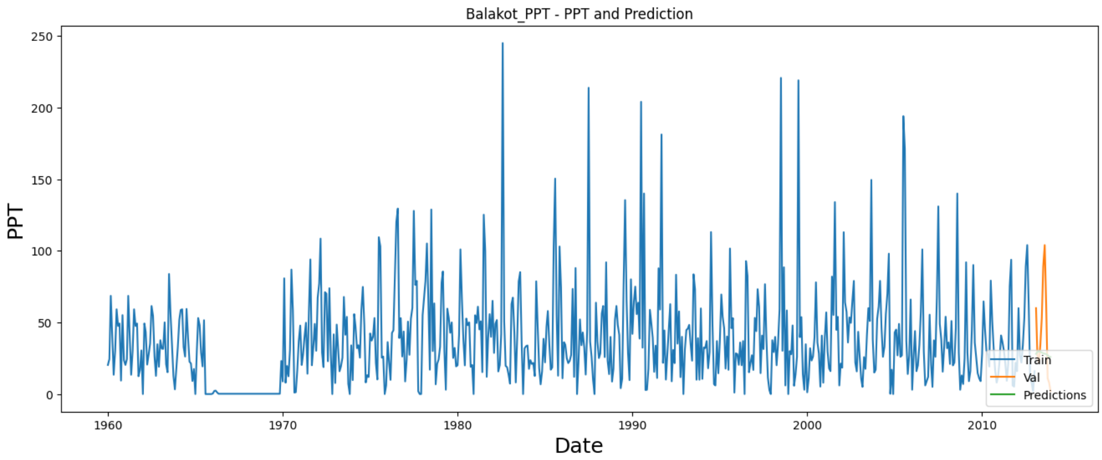

Figure 5.15: Prediction for Balakot region using LSTM

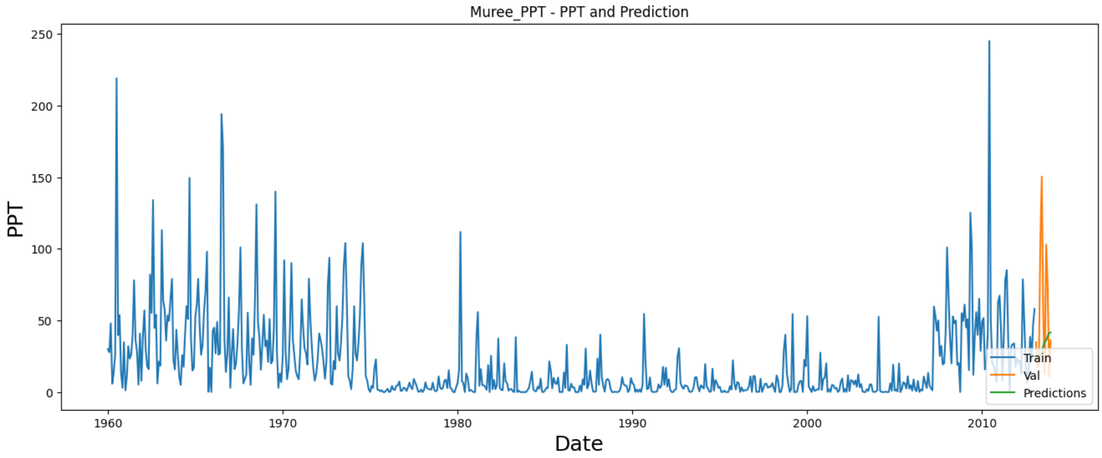

Figure 5.16: Prediction for Murree region using LSTM

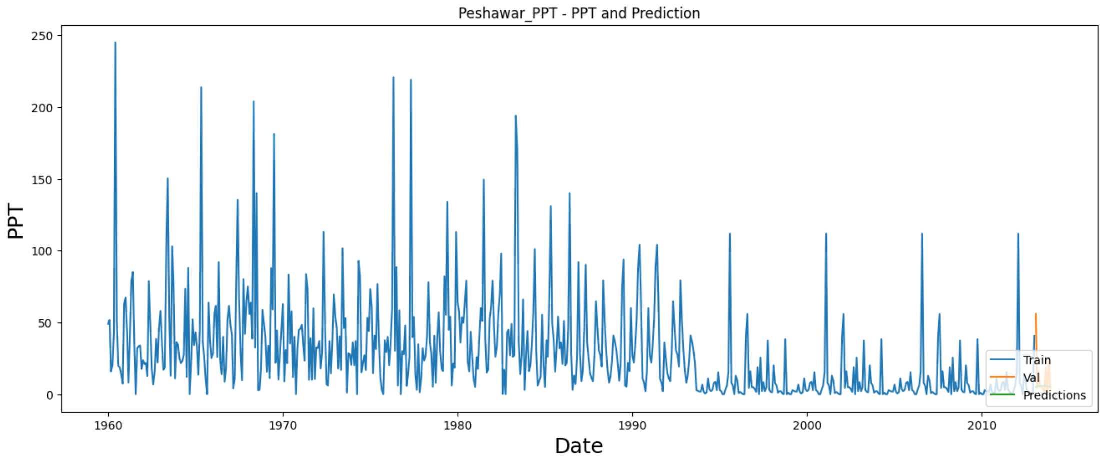

Figure 5.17: Prediction for Peshawar region using LSTM

Table 5.2: The performance measures of the LSTM are listed.

Regions	RMSE	MAE
Gilgit	8.1828431150925	7.22638620896
Balakot	34.922131977467465	27.204247908158735
Chilas	32.40819699041126	26.913923783735797
Murree	51.486152270232665	37.22803590947932
Peshawar	17.479056436637226	10.296943421450528

5.4 Conclusion

Conclusion: Weather forecasts are so complicated, that supercomputers are utilized to execute daily computations in past meteorological work. This paper employed the ARIMA model and the LSTM model to match the monthly exceptional precipitation in the Upper Indus Basin using a personal computer. Comparatively, it is clear from the data that both the ARIMA and LSTM approaches were capable of living up to our expectations. The traditional algorithm (ARIMA) model and the deep learning-based algorithm (LSTM) model were compared to the real precipitation data in this study. As a result of the lower error rate in comparison to other models, the outcomes demonstrate the overall efficiency of ARIMA as a technique for forecasting. It should be noted that the ARIMA model performed better with less data. The results of this study should be accurate because real-time data were used to conduct it. Since we only have a small amount of data, Arima fared better than expected in our study. However, the outcome might have been different if we had access to large data. In order to corroborate this conclusion, it is strongly encouraged to do this study once more with more substantial data and compare findings with those from other studies.

Bibliography

[1] Warren S McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[2] Judith E Dayhoff. *Neural network architectures: an introduction*. Van Nostrand Reinhold Co., 1990.

[3] Dan Hammerstrom. “Neural networks at work”. In: *IEEE spectrum* 30.6 (1993), pp. 26–32.

[4] Jure Zupan. “Introduction to artificial neural network (ANN) methods: what they are and how to use them”. In: *Acta Chimica Slovenica* 41 (1994), pp. 327–327.

[5] H Luo and S Hanagud. “Dynamic learning rate neural network training and composite structural damage detection”. In: *AIAA journal* 35.9 (1997), pp. 1522–1527.

[6] Rbc Allard. “Use of time-series analysis in infectious disease surveillance.” In: *Bulletin of the World Health Organization* 76.4 (1998), p. 327.

[7] Imad A Basheer and Maha Hajmeer. “Artificial neural networks: fundamentals, computing, design, and application”. In: *Journal of microbiological methods* 43.1 (2000), pp. 3–31.

[8] Intergovernmental Panel On Climate Change. “Climate change 2007: Impacts, adaptation and vulnerability”. In: *Genebra, Suıça* (2001).

[9] Carlos Gershenson. “Artificial neural networks for beginners”. In: *arXiv preprint cs/0308031* (2003).

[10] Sun-Chong Wang. *Interdisciplinary computing in Java programming*. Vol. 743. Springer Science & Business Media, 2003.

[11] Kenneth Hewitt. “The Karakoram anomaly? Glacier expansion and the’elevation effect,’Karakoram Himalaya”. In: *Mountain Research and Development* (2005), pp. 332–340.

[12] Anthony Arguez and Russell S Vose. "The definition of the standard WMO climate normal: The key to deriving alternative climate normals". In: *Bulletin of the American Meteorological Society* 92.6 (2011), pp. 699–704.

[13] Andrej Krenker, Janez Bešter, and Andrej Kos. "Introduction to the artificial neural networks". In: *Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech* (2011), pp. 1–18.

[14] Karen Frenken et al. "Irrigation in Southern and Eastern Asia in figures: AQUASTAT Survey-2011." In: *Water Reports* 37 (2012).

[15] G Hinton, N Srivastava, and K Swersky. "Neural networks for machine learning—Lecture 6e—rmsprop: Divide the gradient by a running average of its recent magnitude". In: *URL: https://www. cs. toronto. edu/tijmen/csc321/slides/lectureslideslec6. pdf* (2012).

[16] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[17] Michael I Jordan and Tom M Mitchell. "Machine learning: Trends, perspectives, and prospects". In: *Science* 349.6245 (2015), pp. 255–260.

[18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[19] R Rajbhandari et al. "Projected changes in climate over the Indus river basin using a high resolution regional climate model (PRECIS)". In: *Climate Dynamics* 44 (2015), pp. 339–357.

[20] Jürgen Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural networks* 61 (2015), pp. 85–117.

[21] Mohammad Y Anwar et al. "Time series analysis of malaria in Afghanistan: using ARIMA models to predict future trends in incidence". In: *Malaria journal* 15.1 (2016), pp. 1–10.

[22] Banhi Guha and Gautam Bandyopadhyay. "Gold price forecasting using ARIMA model". In: *Journal of Advanced Management Science* 4.2 (2016).

[23] Abien Fred Agarap. "Deep learning using rectified linear units (relu)". In: *arXiv preprint arXiv:1803.08375* (2018).

[24] Krystal M Perkins et al. "International perspectives on the pedagogy of climate change". In: *Journal of Cleaner Production* 200 (2018), pp. 1043–1052.

[25] I Roy. “Role of “natural factors” on recent climate change underestimated, research shows”. In: *ScienceDaily, University of Exeter* (2018).

[26] Bahareh Karimi et al. “Monthly rainfall prediction using ARIMA and gene expression programming: A case study in Urmia, Iran”. In: *Online J Eng Sci Technol* 2.3 (2019), pp. 8–14.

[27] Christopher E Ndehedehe et al. “Modelling the impacts of global multi-scale climatic drivers on hydro-climatic extremes (1901–2014) over the Congo basin”. In: *Science of The Total Environment* 651 (2019), pp. 1569–1587.

[28] Negar Shahvari et al. “Assessing climate change impacts on water resources and crop yield: a case study of Varamin plain basin, Iran”. In: *Environmental monitoring and assessment* 191 (2019), pp. 1–12.

[29] Muhammad Usama et al. “Unsupervised machine learning for networking: Techniques, applications and research challenges”. In: *IEEE access* 7 (2019), pp. 65579–65615.

[30] Bailin Yang et al. “Traffic flow prediction using LSTM with feature enhancement”. In: *Neurocomputing* 332 (2019), pp. 320–327.

[31] Mohamed Alloghani et al. “A systematic review on supervised and unsupervised machine learning algorithms for data science”. In: *Supervised and unsupervised learning for data science* (2020), pp. 3–21.

[32] Rene Y Choi et al. “Introduction to machine learning, neural networks, and deep learning”. In: *Translational vision science & technology* 9.2 (2020), pp. 14–14.

[33] Yiqing Hua. “Bitcoin price prediction using ARIMA and LSTM”. In: *E3S Web of Conferences*. Vol. 218. EDP Sciences. 2020, p. 01050.

[34] Mudassar Hussain et al. “A comprehensive review of climate change impacts, adaptation, and mitigation on environmental and natural calamities in Pakistan”. In: *Environmental monitoring and assessment* 192 (2020), pp. 1–20.

[35] Tammy Jiang, Jaimie L Gradus, and Anthony J Rosellini. “Supervised machine learning: a brief primer”. In: *Behavior Therapy* 51.5 (2020), pp. 675–687.

[36] Adil Moghar and Mhamed Hamiche. “Stock market prediction using LSTM recurrent neural network”. In: *Procedia Computer Science* 170 (2020), pp. 1168–1173.

[37] Graeme L Stephens et al. “Earth’s water reservoirs in a changing climate”. In: *Proceedings of the Royal Society A* 476.2236 (2020), p. 20190458.

[38] Rizgar Zebari et al. “A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction”. In: *Journal of Applied Science and Technology Trends* 1.2 (2020), pp. 56–70.

[39] Meftah Elsaraiti and Adel Merabet. “A comparative analysis of the arima and lstm predictive models and their effectiveness for predicting wind speed”. In: *Energies* 14.20 (2021), p. 6782.

[40] Jason Bell. “What is machine learning?” In: *Machine Learning and the City: Applications in Architecture and Urban Design* (2022), pp. 207–216.

[41] Zhiqiang Li, Yaming Wang, and Jiazi Xie. “Application of the ARIMA-LSTM model in monthly precipitation prediction”. In: *2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. Vol. 5. IEEE. 2022, pp. 1434–1441.

[42] Uppala Meena Sirisha, Manjula C Belavagi, and Girija Attigeri. “Profit prediction using Arima, Sarima and LSTM models in time series forecasting: A Comparison”. In: *IEEE Access* 10 (2022), pp. 124715–124727.

[43] Jiwei Zhao, Guangzheng Nie, and Yihao Wen. “Monthly precipitation prediction in Luoyang city based on EEMD-LSTM-ARIMA model”. In: *Water Science & Technology* (2022), wst2022425.

[44] Multi-Objective Decision Making. “Synthesis Lectures on Artificial Intelligence and Machine Learning”. In: ().