ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES
SCHOOL OF INFORMATION SCIENCE

RAINFALL PREDICTION USING COMBINED SATELLITE AND STATION DATA: A DEEP LEARNING APPROACH

A Thesis Submitted to School of Graduate Studies of Addis Ababa University in Partial Fulfilment of the Requirements for the Degree of Master of Science in Information System

By:

Mubarek Jemal

Advisor:

Melkamu Beyene (Ph.D.)

JULY, 2023

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY
COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCES
SCHOOL OF INFORMATION SCIENCE

RAINFALL PREDICTION USING COMBINED SATELLITE AND STATION DATA: A DEEP LEARNING APPROACH

By: Mubarek Jemal

Name and signature of Members of the Examining Board

Melkamu Beyene (Ph.D.) ______________ ___________
Advisor Signature Date

Michael Melese (Ph.D.) ______________ ___________
Examiner Signature Date

Wondwossen Mulugeta (Ph.D.) ______________ ___________
Examiner Signature Date

July, 2023

DECLARATION

This thesis has not previously been accepted for any degree and is not being concurrently submitted in candidature for any degree in any university.

I declare that this thesis entitled “RAINFALL PREDICTION USING COMBINED SATELLITE AND STATION DATA: A DEEP LEARNING APPROACH” is a result of my investigation, except where otherwise stated. I have conducted this study independently while receiving guidance and support from my research advisor. Any external sources are acknowledged by citations giving explicit references. A list of references is appended.

Signature: ___________________________

Mubarek Jemal Ahmed

The thesis has been submitted for examination with my approval as a university advisor.

Advisor‘s Signature: __________________________

Melkamu Beyene (Ph.D.)

ACKNOWLEDGMENTS

Before all, I praise Almighty ALLAH for all the blessings I had. It would have been almost impossible for me to complete this thesis if many people next to Allah (S.W) weren‘t willing to give me their helping hands.

I would like to express my heartfelt gratitude to my advisor Melkamu Beyene (Ph.D.) for his valuable insights and guidance, constructive feedback, and continues support in shaping the direction and quality of this thesis. Without his mentorship, I wouldn‘t have gotten far doing this thesis work.

I extend my sincere gratitude to the National Meteorological Agency staff particularly Ato Leta Bekele and Aynalem Shegaw for providing me with the necessary data for the study and for their unreserved help throughout the study time.

Thank you, all my loved family and friends Especially, Kirubel wondimagegnehu, Mulate Kedicho, and Faris Kedicho for your moral support, encouragement, and Motivation during my academic year and in this thesis work.

ABSTRACT

Currently, Ethiopia has high rainfall variance, which is a result of global climate change that has an influence on the environment, property values, and human lives. Accurate rainfall prediction is highly important to smart agriculture practices for developing countries. For rainfall prediction, using station data alone often lacks the required accuracy and spatial coverage, and satellite data has spatial coverage but cannot predict rainfall as accurately as station data. The objective of this research is to develop a model for rainfall prediction using deep learning approaches by combining weather station and satellite data. A design science research methodology was used to develop a rainfall prediction model with 30 years (1990 - 2020) of daily weather station data from the National Meteorological Agency Ethiopian and satellite data from TAMSAT v3.1 and JRA-55 climate models. In data engineering, missing values were handled using mean imputation by dividing the dataset based on the three seasons of Ethiopia. Deep learning approach that includes multi-layer perceptron (MLP), Convolutional Neural Network (CNN), Long Short Term Memory (LSTM), and Bidirectional Long Short Term Memory (BiLSTM) were experimentally evaluated to predict rainfall for selected areas. Lastly, we proposed a model using Bidirectional Long Short Term Memory (BiLSTM) architecture that capable of forecasting daily rainfall for Ethiopia. The performance of the model is evaluated using the state of the art performance evaluation metrics such as; Root Mean Square Error (RMSE), Mean Square Error (MSE), and Mean Absolute Error (MAE), and the results were 0.0472, 0.0025, and 0.021 respectively. We also compared the proposed model with other deep learning approaches like MLP, CNN, and LSTM. The proposed BiLSTM model outperformed LSTM with an RMSE of 0.0015; CNN with RMSE of 0.0023, and MLP with RMSE of 0.0025. The experimental results show that the Bidirectional Long Short Term Memory (BiLSTM) model has a lower RMSE, MSE, and MAE.

Key Words: Rainfall Prediction, Deep Learning, Design Science, Climate Model, BiLSTM

Table of Contents

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

ANN	Artificial Neural Network
BiLSTM	Bidirectional Long Short Term Memory
BPNN	Back-Propagation Neural Network
BPTT	Back-Propagation Through Time
CDT	Climate Data Tools
CNN	Convolutional Neural Network
CPU	Central Processing Units
CRU	Climatic Research Unit
DLWP	Deep Learning Weather Prediction
DNN	Deep Neural Network
DSR	Design Science Research
DSRP	Design Science Research Process
FGR	Full Gate Recurrence
GCM	Global climate model
GEM	Global Environmental Multi-scale Model
GFS	Global Forecast System
GPU	Graphics Processing Units
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
MLP	Multi-Layer Perceptron
MLR	Multiple Linear Regression
MSE	Mean Square Error
NAM	North American Mesoscale Model
NMA	National Meteorology Agency
NWP	Numerical weather prediction

RCM	Regional climate models
ReLU	Rectified Linear Unit
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
WRF	Weather Research and Forecasting

CHAPTER ONE

INTRODUCTION

1.1. Background of the Study

About 85% of Ethiopians live in rural areas and depend on agriculture for their livelihood. The agricultural system in Ethiopia is highly dependent on rainfall. Accurate rainfall predictions have highly important to smart agriculture practices for developing countries like Ethiopia [1]. It can enhance productivity, reduce resource wastage, and improve resilience in the face of climate variability. It has the potential to empower farmers with valuable information for decision-making, ultimately contributing to sustainable agricultural development and food security. Information on rainfall is essential for managing water resources, planning for food production and other outdoor activities [2]. When a severe dry season or heavy rain occurs during crucial growth and development stages, crop production may be significantly reduced [3]. Especially, in developing countries like Ethiopia agriculture is dependent on the weather and climate condition. In order to maximize agricultural returns and enhance livelihoods, adaptation and mitigation methods must take into account both weather and climate forecasts [3].

Globally agricultural production is now more uncertain and at risk due to the ongoing rise in unpredictable weather patterns, including excessive rains, temperatures, and frequent droughts [4]. Weather forecast helps us to know what is going to happen in the next 24 hours and up to two weeks ahead, but climate forecast helps to know what will likely happen in the coming seasons and years. Forecasting weather provide critical information about future weather. Forecasting the weather is an essential and vital process in people's day to day lives, evaluates the variation taking place in the atmosphere's current state. The first weather predictions were made in the nineteenth century [5]. In meteorology, weather forecasting is important [6]. A meteorologist's job in weather prediction is to forecast how the environmental air conditions will change at a certain time and location. Historically, forecasting relied heavily on human forecasters, they tried to detect the weather situation to forecast the condition short-range of the atmosphere by trial and error, but in the information era, it is now accomplished by using technology and data.

Ethiopia is located in the eastern horn of Africa. Its elevations range from hundreds of meters below sea level in the northeast to more than four thousand meters above sea level in the northern highlands[7]. Missionaries started meteorological weather prediction in Addis Ababa at the end of the 19th century [8] in Ethiopia. Furthermore, the Meteorological station was laid out in 1890 in Adamtilu and in 1986[1896] in Gambela. Then, at that point, after the end of the Second World War, from 1946-1949, the government did meteorology technology for the agricultural area to control locusts. After that, due to the growing demands for meteorological information for safe operations of air transport and as the other economic and social sectors began to understand the significance of meteorological services Ethiopian government officially recognized the National Meteorological Services Agency under proclamation No 201 of 1980 on December 31, 1980. The weather stations are increased gradually to collect ground based station weather data.

Data from satellite will only read remotely identified rainfall, temperature, and other weather data on the vegetation, ground, and water body, so on. Additional feature of the differences lies in the area over which the data is recorded. The benefits of satellite data are consistency, accessibility, and high spatial and temporal resolutions over large areas. Data from satellite are available at time scales that are advantageous for trend analysis [9],[11]. Since the rain is not directly observed by the satellite, the precipitation data must be recovered using some sort of heuristic or machine learning approach, such as regression analysis or binary classification for precipitation detection [12]. There are essentially two methods for predicting rainfall, such as dynamic and empirical methods [13]. The analysis of past rainfall data is the basis of the empirical approach. For in-depth analysis linked to climate and rainfall prediction, this data together with its relationships with atmosphere, wind, air temperature, surface temperature, and oceanic factors is used. The physical models that are based on equation systems and used to forecast the global climate system are known as the "dynamical approach". In this dynamic technique, the numerical rainfall forecasting method is applied [14].

Rainfall is a type of precipitation in which the water is disseminated throughout the Earth's surface, oceans, and atmosphere. It is crucial for life when water from the Earth's surface evaporates and rises as water vapor into the atmosphere, carrying heat with it. Rain is responsible for storing the majority of the planets freshwater, which is vital for the survival of plants and animals. When water vapor rises into the atmosphere, it condenses to form cloud droplets, which then release raindrops into the atmosphere [15], [16]. The measurement of rainfall using a rain gauge and temperature using a thermometer is influenced by several factors such as wind speed, wind direction, landscape characteristics, and fetch parameters. In contrast, satellite data is not affected by any of these factors [17].

In Ethiopia rainfall is highly inconstant both in distribution and amount across areas and seasons [18]. The macro-scale pressure systems and monsoon flows, which are linked to changes in the pressure systems, cause seasonal and yearly fluctuations in rainfall [19]. Additionally, the rainfall patterns in Ethiopia are influenced by multiple weather systems, including the Inter-tropical Convergence Zone (ITCZ), Red Sea Convergence Zone (RSCZ), Subtropical Jet (STJ), Tropical Easterly Jet (TEJ), Somali Jet, and Tropical Easterly Jet (TEJ), as mentioned in the NMA (1996).

Accurate and reliable predictions of rainfall can help in making informed decisions and mitigating potential risks associated with extreme weather events. Over the years, statistical methods, machine learning, and deep learning approaches have emerged as powerful tools for rainfall prediction. These approaches utilize historical weather data and other relevant variables to build predictive models that can forecast rainfall patterns. Traditional statistical models, such as autoregressive integrated moving average (ARIMA) and autoregressive (AR) models have been widely used for rainfall prediction. These models capture the temporal dependencies and statistical characteristics of rainfall time series data. However, their performance may be limited when facing complex and non-linear relationships present in weather data. Various machine learning algorithms, including decision trees, random forests, support vector machines (SVM), and gradient boosting methods, have been applied to rainfall prediction. These algorithms can capture intricate relationships between weather variables and rainfall patterns, enabling accurate predictions.

Furthermore, deep learning approaches, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory (LSTM) networks, excel at capturing complex temporal dependencies and patterns in sequential data. By providing historical weather data and other variables into these models, they can learn intricate relationships and make accurate rainfall predictions. Deep learning approaches have demonstrated promising results in rainfall prediction tasks, often outperforming traditional statistical and machine learning methods.

The inspiration for this research which on rainfall prediction is, its profound influence on various aspects of political, economic, and social life. In Ethiopia, a country heavily reliant on rain-fed agriculture and vulnerable to climate variability, rainfall prediction assumes even greater significance. Also, the availabilities of big data in meteorological sectors such as station data, satellite data, radar images, and etc., are needs proper methods and techniques that help to manage and predicts. Despite many global research efforts, the availability of models that can accurately predict rainfall is lacking. This is why many donors including the Giant ones are investing in AI based climate smart agriculture solutions as part of the effort to increase productivity of agriculture. There has been a lack of research conducted in this area within our country. Even after this study; rainfall prediction may require more researches because lack of research that predict the amount of rain in Ethiopia. Based on those reasons, this study motivated to design and develop the technology- based rainfall prediction model by utilizing deep learning approaches by combining station and satellite data's.

1.2. Statement of the problem

Rainfall is a nonlinear system that depends on a number of important variables, including hydrology, geography, and circulation [20]. For many purposes, including agricultural sectors, disaster preparedness, water resource management, an accurate rainfall forecast is important [4]. Agriculture depends mainly on rain especially in countries like Ethiopia, as an irrigation system is not much in practice. One of the key reasons is because of its natural topography. Even though it is incredibly useful, predicting rain or any other meteorological conditions is very difficult [21]. Because, rainfall is influenced by a complex relationships of various meteorological features, including temperature, relative humidity, wind speed, air pressure, and more. These variables interrelate in a nonlinear way, making it difficult to accurately develop and predict their characteristics. In turn, this consideration can be used to support many vital sectors that are affected by unpredictable rainfall like water resources, agriculture, and tourism [22]. Weather-related catastrophes like droughts and floods now often occur and result in significant losses. This needs additional enhancement in the accuracy of weather predictions [23]. More comprehensive and varied datasets are now available as a result of efforts to improve the collection of data. Radars, satellites, weather stations, and other remote sensing technologies offer valuable insights that help us understand weather patterns. Predictive models can capture a more thorough picture of atmospheric conditions, enhancing rainfall forecasts, by combining a bigger amount of high-quality data.

However, traditional methods of rainfall prediction using weather station data alone often lack the desired precision and spatial coverage and the ability to capture complex rainfall patterns [24]. Weather stations are often scarce and are frequently concentrated in cities or urban areas. This sparse distribution can result in limited spatial coverage, making it challenging to capture rainfall patterns accurately in remote or rural regions. The absence of nearby stations can lead to significant data gaps, hindering the ability to predict rainfall in those areas [25]. Weather station data alone might not give complete coverage of rainfall patterns. But just in the immediate vicinity of the station, station data is very accurate in capturing what is happening at ground level [26]. Weather stations are expensive, need regular maintenance, and are constantly capturing data. Poor coverage requires interpolation between stations in rural regions.

The development of satellite technology has transformed weather monitoring by enabling a more comprehensive understanding of atmospheric conditions across vast geographic regions [27]. A more comprehensive understanding of weather patterns is made possible by the useful information provided by satellite data on cloud cover, moisture content, and other atmospheric characteristics. However, satellites are unable to estimate precipitation on the ground as precisely as meteorological stations [28]. This is because of, the lower temporal resolution and potential limitations due to cloud cover or atmospheric interferences can impact the accuracy and timeliness of predictions, particularly for short-term rainfall prediction. It is feasible to get around the problems or Limitations of particular data sources and increase the accuracy of rainfall predictions by combining satellite data with station data. The combined dataset covers a wider geographic location and provides a more complete representation of rainfall patterns.

Weather stations provide point measurements of rainfall at specific locations, while satellite-based rainfall estimates cover larger spatial areas. By combining the two data sources, the resulting dataset covers a broader geographic extent, providing a more comprehensive representation of rainfall patterns. The satellite data fills in the spatial gaps between weather stations, especially in remote or data-sparse regions, improving coverage and capturing localized rainfall variations.

Finding reliable and accurate techniques or tools to evaluate and extract hidden knowledge from the vast amount of data is crucial due to the recent increase in the availability of climate data (radar and satellite maps, observational records, proxy data, observations from ships and aircraft, etc.). Understanding the unpredictability of rainfall and making accurate rainfall predictions can benefit from useful knowledge. In order to predict rainfall, National

Meteorology Agency of Ethiopia are using Numerical Weather Prediction (NWP) model. This NWP requires large computational resources due to the complex mathematical calculations involved. High-performance computing systems are often necessary to process the vast amount of data and perform the simulations [29]. The model is sensitive to the precision of the first conditions and input data. Small errors or uncertainties in the initial conditions can lead to significant divergences in the predicted outcomes.

As weather variables are contains spatial and temporal variations within combined dataset. Traditional statistical methods and some conventional machine learning algorithms may struggle to capture complex non-linear interactions between rainfall and the multitude of atmospheric variables obtained from combined station and satellite data [30],[31]. However, the patterns of rainfall often exhibit non-linear characteristics, that influenced by complex atmospheric conditions. Statistical and machine learning models may not capture these nonlinearities effectively, leading to limitations in accurately predicting rainfall patterns, as approaches assume linear relationships between independent variables and rainfall [32]. These methods also face difficulties in handling and processing high-dimensional data, leading to model complexity, overfitting, or increased computational requirements. This problem can impact the precision and the power of predictive models.

To overcome the limitations of NWP model, statistical and machine learning approaches, there is a necessity to develop an effective approach that combines satellite data with station data and employs advanced techniques like deep learning. In recent years, deep learning for weather prediction has emerged as an influential tool in several fields of research, including rainfall prediction [29]. Deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have confirmed remarkable capabilities in capturing complex patterns and making accurate predictions. These models can effectively learn from big datasets and extract meaningful values, making them appropriate for predicting rainfall tasks.

Rainfall pattern varies from location to location, based on the occurrence of geographic topographies such as mountains, valleys, and bodies of water can significantly impact rainfall patterns [33].There are geographical differences in rainfall as a result of local and regional variables influencing rainfall patterns. Also, rainfall model developed for one location may not work as well for another location, because rainfall depends on several dependent variables like wind speed, temperature, relative humidity, etc. of that location.

Several studies have explored the use of satellite data, weather station data and deep learning techniques independently for rainfall prediction. However, related research that combines both data sources and employs deep learning algorithms for rainfall prediction is relatively limited. As far as the researchers' knowledge is concerned, the use of deep learning in the rainfall prediction problem by combining the two sources is not well studied in the Ethiopian context.

1.3. Research Questions

The following research questions are investigated and answered in this study.

1. What are the most effective meteorological variables for predicting rainfall from weather station and satellite data?
2. To what extent do deep learning approaches predict rainfall based on combined weather station and satellite data in Ethiopia?
3. Which deep learning model architecture and parameters are more suitable to predict rainfall patterns using combined station and satellite data?

1.4. The objective of the Study

1.4.1. General Objective

The general objective of this study is to design a model for rainfall prediction using deep learning approaches by combining weather station and satellite data.

1.4.2. Specific Objectives

To achieve the general research objective, below are lists of specific activities executed in this study.

- To understand the existing rainfall prediction methods and frameworks.
- Collecting both station and satellite datasets.
- To prepare data for model building by cleaning, transforming and integrating from different sources.
- To develop a prediction model that can accurately predict a rainfall.
- To compare and test the models performance with other algorithms and existing approach.

1.5. Scope and Limitation of Study

The goal of this study is to come up with a best model for rainfall prediction using deep learning approach and compare the performance with other deep learning algorithms, but either statistical or machine learning techniques were not used. Includes collections of weather station and satellite data, literature review, data engineering techniques, training of deep learning models, evaluations of the models, and conclusion are incorporated in this study.

The scope of the research is limited to predicting rainfall in three locations in Ethiopia (i.e. Addis Ababa, Adam and Bishoftu). Those locations contain different geographic topographies and different weather conditions. Another scope of the research is limited to daily rainfall prediction, but has not provided the hourly, monthly, and seasonal rainfall prediction.

1.6. Significance of the Study

Predicting the rainfall has always been important to people's lives and their daily activities. With prediction information, people can better anticipate what will happen or what to expect and plan accordingly. Furthermore, accurate forecasting of extremely rainfall or the amount of the rain as well as identifying highly correlated variables with the rainfall or precipitation can be very helpful in formulating policies and creating a system for managing risks. The findings from this research will help different sectors, including governmental and non-governmental organization to avoid or reduce significant losses that happing because of unpredictable rainfalls. Specifically, this study has a lot of contribution;

- For Smart agriculture, farmers can make informed choices, improve productivity, reduce risks, and contribute to sustainable and resilient agricultural practices.
- For Climate Change Adaptation; plays a vital role in adapting agricultural practices to changing climate conditions.
- For transportation (such as shipping, aviation, and roads)
- For flood warning, industrial sectors, and
- For other socio-economic sectors to plan for near future.

1.7. Organization of Research

The remaining parts of this study are structured as follows:

Chapter two is the Literature Review and Related works part. The chapter introduces the evolutions of weather, rainfall prediction, Numerical weather prediction and Deep learning weather prediction methods, different deep neural networks, and literatures related to deep learning summary, and research gaps. While **Chapter three** is the Methodology section. This section describes the research methodology, research approaches, and design science research process and summarizes the chapter. **Chapter Four** is solution design. The artefact design process and the designed architecture are described. Furthermore, data acquisition and understanding, data engineering, and model building processes are discussed. C**hapter five** is experiment and evaluation of the artefacts. The chapter describes experimental setups, tools, and protocols. Lastly, conclusions and recommendations of the study are made in **Chapter six.**

CHAPTER TWO

LITERATURE REVIEW

2.1. Introduction

The main goal of this chapter is to present recent literature written to show weather prediction, especially rainfall prediction, and the role of deep learning algorithms in forecasting future weather conditions and shows the comparison of deep learning approach performances with other statistical and machine learning algorithms. This chapter starts by showing the evolution of weather prediction including rainfall predictions; clarifying basic concepts related to rainfall prediction. Due to the aforementioned factors, many academics and researchers have recently focused on rainfall prediction using a variety of methodologies, including deep learning, numerical-based, and statistical methods, are among others. But, it is challenging to cover all the studies which are done by different researchers. Therefore, we reviewed the most essential research that related to the deep learning approach, machine learning, and Statistical technique based which are related to this study.

2.2. Evolution of weather

The science of weather prediction originated from the practice of early civilizations utilizing regular astronomical and meteorological events to monitor and predict seasonal weather variations. All the way through history, efforts have been made to create predictions based on historical weather parameters and individual observations. But by the end of the Renaissance, it had become gradually clear that the natural philosophers' hypotheses were insufficient and that more information was required to improve the understanding of the atmosphere. Instruments were required to measure the characteristics of the atmosphere, such as moisture content, temperature, pressure, and so forth [34].

As these meteorological instruments were being improved during the seventeenth through nineteenth centuries, other related observational, theoretical, and technological advancements added to our understanding of the atmosphere. Additionally, people in various locations began taking and recording atmospheric measurements. This knowledge enabled the development of fundamental weather maps and facilitated the recognition and examination of surface wind patterns and storm systems. More data became accessible for observational weather prediction in the nineteenth and twentieth centuries with the establishment of regional and global meteorological observation networks. In the early 20th century, Ethiopia began to use more modern methods such as radiosondes and weather balloons to measure atmospheric conditions. In the late 20th century, Ethiopia began to use satellite imagery and computer models to improve weather prediction accuracy. Today, Ethiopia has access to a variety of sophisticated weather prediction tools, including Doppler radar, numerical weather prediction models, and satellite imagery. These tools are used to provide more accurate and timely forecasts, which are essential for the country’s agricultural and economic development.

2.3. Weather prediction

Weather is the condition of the atmosphere at a specific moment and location, which includes a number of meteorological factors including temperature, humidity, wind direction and speed, cloud cover, precipitation, and air pressure. Solar radiation, the Earth's rotation, air pressure systems, moisture content, and interactions between the atmosphere, seas, land, and other geographical characteristics are only a few of the many variables that affect weather [34]. These elements help to create weather patterns and phenomena, which can include calm skies and clear skies as well as storms, rain, snow, fog, and severe occurrences like hurricanes, tornadoes, and heat waves. In many aspects of human existence, including agriculture, transportation, building, energy generation, outdoor activities, and crisis management, weather forecasts are essential [26]. People, organizations, and governments may plan activities, make informed decisions, and reduce risks related to changing weather conditions with the help of accurate weather predictions.

Weather prediction has long been considered as an issue of physical theory, and meteorological experts have dedicated themselves to increasing prediction accuracy by considering of physical principles, which is a theory-driven approach. It has evolved into a typical huge spatio-temporal data due to the rapid expansion of multi-scale meteorological, multi-dimensional, and multi-source data [23]. The technique of evaluating and projecting future atmospheric conditions based on current observations, historical data, and scientific models is known as weather prediction. The objective of weather forecasting is to deliver precise and timely information about the anticipated weather, including temperature, precipitation, wind patterns, and other atmospheric factors, for particular places and periods. Predicting the time, intensity, duration, and spatial distribution of precipitation occurrences falls under the category of weather prediction, more especially rainfall prediction [35]. For many purposes, including agriculture, water resource management, flood forecasting, urban planning, and emergency response, rainfall prediction is essential.

2.4. Rainfall prediction

Rainfall prediction is a complex and challenging task that involves understanding and forecasting the timing, amount, intensity, and spatial distribution of precipitation events. In different areas, rainfall amounts are measured by rain gauges and weather stations[29]. These on-the-ground measurements offer vital data for understanding past rainfall patterns and identifying trends. The ability to anticipate rainfall is aided by the data that satellite imagery, weather radar systems, and other remote sensing technologies provide on cloud formations, precipitation systems, and rainfall intensity across broad areas [28]. The agricultural industry and the production of seasonal fruits and vegetables are frequently protected by rainfall prediction, which helps to maintain their quantity and quality in accordance with the amount of rain they need [35]. Data scientists have been attempting to use data-driven computing paradigms to mine the relationships between complex temporal and spatial meteorological data elements in recent years [29]. Deep Learning Weather Prediction (DLWP) has emerged as a popular research area and is anticipated to be able to address the data challenges associated with the traditional theory-driven methods. In this section, we briefly discuss the two computing paradigms of weather prediction;

2.4.1. Numerical Weather Prediction (NWP)

By addressing the physics and dynamics equations that describe how the atmosphere moves and changes, numerical weather prediction (NWP) is a set of procedures that forecasts future atmospheric conditions. The most general method for predicting the weather is to use global or regional models. These models now have higher resolution, more precise handling of nonlinear dynamics, and a better description of physical processes. The memory and speed of computers have increased during the past three decades, and this area of numerical prediction activity has practically kept up with them. Traditionally, incorporating the governing partial differential equations (PDEs) based on the existing weather situations yields the weather conditions in the future [36]. The thermodynamic, kinetic, radiative, and chemical processes of the atmosphere are described by nonlinear PDEs. The PDEs of NWP must be resolved numerically using temporal and spatial discretization since the physical process is continuous.

The meteorological components that are typically predicted by numerical weather prediction models include temperature, wind speed, rainfall, average sea level pressure, etc. Generally speaking, the essential steps of the NWP may be summed up as follows, according to [29]: acquiring the initial datasets of observations, such as those from remote sensing, in-situ observation, and simulations of the NWP models at the most recent time point; pre-processing the raw datasets through the data assimilation analysis system, including data quality control and quality assurance; inputting the pre-processed datasets into the atmospheric dynamic model equations for prediction; and post-processing and visualizing the outcomes of predictions. As a result, it is possible to find approximations of the PDE solutions using finite difference equations (FDEs). As an illustration, the advection equation [37]

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \qquad \dots\dots\dots\dots\dots \quad (1)$$

We take discrete values for $x\ and\ t\colon x_j = j\Delta x\ \ and\ t_n = n\Delta t$ where Δx is the grid space and Δt is the time step of combination. The solution of the FDE is defined at the distinct points $\left(x_j, t_n\right) = (j\Delta x, n\Delta t)$:

$$u_j = u(j\Delta x, n\Delta t) = u\left(x_j, t_n\right) \ \dots\dots\dots\dots\dots\dots.(2)$$

The NWP model assumes that the atmosphere is made up of several lumps, the corner points of which are referred to as the grid points. The difficulty of the simulation rises with the number of lumps. Through simulation, the model creates the model atmosphere's future state from its original condition at all grid points [38]. Since the start of the twentieth century, numerical weather prediction has been evolving. Since then, ongoing efforts to produce robust computers, improve modelling methods, and provide more precise observations from a variety of sources have all helped to provide reliable weather forecasts for use by individuals and organizations all around the world [39]. Before being applied to a domain (geographic region), the equations are converted into computer code and employ governing equations, numerical techniques, parameterizations of various physical processes, together with initial and boundary conditions. The NWP process almost always involves omissions, estimates, approximations, and compromises.

$$f(x_i + \Delta x) = f(x_i) + \Delta x\frac{\partial f}{\partial x}\bigg|x_i + \frac{\Delta x^2}{2!}\frac{\partial^2 f}{\partial x^2}\bigg|x_i + \cdots + \frac{\Delta x^n}{n!}\frac{\partial^n f}{\partial x^n}|x_i \quad \dots\dots\dots.(3)$$

Numerous research using Mesoscale modeling have been conducted to examine the impact of urban areas on circulation [40], [41]. The studies commonly have a high horizontal resolution with a clearly discernible city. However, limitations on the resolution of the model for operational numerical weather prediction (NWP) models result in frequently poorly resolved cities. These limitations are brought on by the need to provide customers with forecasts as quickly as possible and the availability of limited computing resources. The complexity of the model is also affected by the timing and resource concerns, as more complicated models require more computation and take longer to integrate [42].

To precisely and timely anticipate the weather has long been the goal of meteorologists. However, there are a number of problems with the traditional theory-driven numerical weather prediction (NWP) methods, such as the need for expensive computational resources, a lack of knowledge of physical principles, and difficulties in extracting information from the enormous amount of observation data [29]. Deep learning methods have demonstrated their effectiveness in extracting temporal and spatial features from spatio-temporal data, as evidenced by their successful implementation in diverse domains such as computer vision, speech recognition, and time series prediction. These data-driven techniques have proven to be valuable tools for leveraging the power of deep learning in uncovering meaningful patterns and insights from spatio-temporal data. Global and regional climate models are used in numerical weather prediction to represent and forecast the behavior of the atmosphere, seas, land surface, and other natural phenomena.

2.4.2. Global and Regional Climate Models

Climate models are computer-based simulations that describe and forecast the behavior of the atmosphere, oceans, land surface, and ice using intricate mathematical procedures. To forecast future climate trends, these models use a variety of data sources, such as historical weather information, measurements of atmospheric and oceanic conditions, and physical equations. Over the past 20 years, significant evolution has been made in the creation and application of models, and the available models now provide us with a solid roadmap for the course of future climate change. Due to the numerous variables involved, computer models are unable to accurately predict the future. The models are primarily based on physical rules, but they also incorporate empirical methods that, for instance, study the intricate processes involved in cloud formation.

The most complex computer simulations represent the entire climate system. Climatic Research Unit (CRU) [43] captures the interactions between the various elements, such as ice and land, in addition to connecting the atmosphere and ocean. As more information becomes available and as the understanding of the climate system Earth's in advances, climate models are regularly improved and updated. Despite their complexity, these models are useful for forecasting future weather patterns and can be used to guide conclusions in water management, the agricultural sector, and other sectors where rainfall is significant.

Global climate models are used to simulate the overall behavior of the Earth's climate system, including large-scale atmospheric and oceanic circulation patterns. Future weather patterns and climate change are predicted using global climate models. The models at roughly 200 – 500-km resolution was considered insufficient for generating climate information required for evaluating the impacts of climate change and variability. These models are complex computer simulations that use mathematical equations to represent the physical processes of the atmosphere, oceans, land surface, and ice [44]. The following are a few of the most well-known global numerical models:

Global Forecast System (GFS): The National Organization for the Atmosphere in America developed it. The model is used to provide forecasts of weather conditions across the globe, including rainfall predictions. The model takes into account a variety of atmospheric factors, including pressure, speed and direction of the wind, temperature, and humidity as well as oceanic variables such as temperature of sea surface. In addition to simulating atmospheric parameters, the GFS model also takes into account external factors that can influence weather patterns, such as solar radiation, the earth's rotation and the influence of other planets on the earth's atmosphere. The model also assimilates observational data from a range of sources, such as weather balloons, aircraft, satellites, and surface observations, to help improve its accuracy.

Global Circulation Model (GCM) is a type of numerical weather prediction model used to simulate and predict the earth's atmosphere on a global scale. GCMs are used to simulate the atmosphere's response to changes in climate, such as those caused by human activities. They are also used to forecast weather and climate on a diversity of timescales, from days to centuries. The models are run on powerful computers and use data from satellites, weather stations, and other sources to simulate the atmosphere's behavior. The results of the simulations are used to make predictions about future weather and climate conditions.

Regional climate models: are employed to model the climate in a specific region, using data from the global models as input. Political boundaries or geographical characteristics with a uniform climate serve as a proxy for regional climate. Regional Climate Models (RCMs) produce 10 to 50 km surfaces by re-modeling GCM outputs; as a result, they are only applicable to a small number of GCMs (for which boundary conditions are available), and they require a lot of processing power, time, and storage to produce a single scenario-by-period output. As a result, most assessment offices and agricultural researchers find it difficult to obtain RCM outputs. Some of the most common regional numerical models include:

The **Weather Research and Forecasting (WRF)** Model was developed supportively by NCEP and by the community of meteorological research. The simulations are based on physical laws and equations that define the behavior of the atmosphere, such as the conservation of energy, the conservation of momentum, and the conservation of mass [50]. The models are used to build the atmosphere's answer to various inputs, such as changes in temperature, pressure, and humidity. The models are then used to forecast the weather, including temperature, precipitation, and wind speed.

The **North American Mesoscale Model:** (NAM) produces many grids (or domains) of weather predictions at varying horizontal resolutions throughout the North American continent. Numerous meteorological variables, such as temperature, precipitation, lightning, and turbulent kinetic energy, are represented in each grid. NAM occasionally utilizes other numerical weather models to track major weather phenomena like hurricanes and to produce high-resolution predictions over specified areas [51].

The **JRA-55** (Japanese 55-year Reanalysis) weather data model is a global atmospheric reanalysis system developed by the japan meteorological agency (JMA).The process of merging past observations with a NWP model to create a consistent record of past weather conditions is referred to as reanalysis. Over the past 60 years, meteorologists, climatologists, and researchers have utilized JRA-55 extensively to investigate weather patterns and climate variability [45]. JRA-55 takes in a lot of observational data from various global observing networks, such as surface weather observations, upper-air soundings, satellite data, and other sources. Weather stations, buoys, ships, aircraft, and satellites collect these observations [46].

MMS – The fifth Generation Mesoscale model: is the most recent of a series of terrain-following, hydrostatic or non-hydrostatic, limited-area models using sigma coordinates that aim to mimic or forecast regional Mesoscale atmospheric circulation [52].

To predict rainfall for specific areas, scientists typically use a process called downscaling, which involves using the output from global climate models as input to regional climate models. This allows scientists to create more detailed predictions of local climate conditions, including rainfall.

2.4.3. Downscaling Global model

One of the approaches developed to enhance model performance is called downscaling. This is the process of using regional climate models for limited areas at high resolution, which allows for a more faithful representation of small-scale atmospheric processes and gives a more detailed structure to regional climate scenarios. Downscaling is the process of taking the output of global climate models (GCMs), which typically have a coarse spatial resolution, and generating high-resolution climate data for a specific region. This is important because regional climate models (RCMs) require high-resolution data to accurately simulate the local climate [53]. Different downscaling techniques differ in terms of accuracy, output resolution, computational and time requirements, and climate science robustness (that is, theoretical background) [50]. There are several methods for downscaling GCMs, including statistical downscaling and dynamical downscaling.

Statistical downscaling: Generate high-resolution climate data using statistical relationships between large-scale atmospheric parameters (such as pressure, sea surface temperature, and precipitation) and local climatic factors (such as precipitation and temperature) [44]. This method is often used when there is a lack of high-resolution atmospheric data, or when the focus is on generating long-term projections. Statistical downscaling requires identifying empirical relationships between huge patterns of climatic factors and local climate. Once such relationships are established, statistical downscaling can be used to ascertain local climate variation from global or regional model results. This approach requires less computing power than dynamic downscaling [54].

Statistical downscaling builds statistical relationships between historically recognized climate data and climate model outputs for the same historical period. Linear regression is a simple and universally used method for statistical downscaling. This method establishes a linear relationship between large-scale climate indicators, such as humidity simulated by GCMs or RCMs, and locally observed humidity. This relationship is developed by evaluating the locally observed data and correlating it with the GCM or RCM output. Values α and β are estimated based on historical time periods and used to derive reduced future climate projections.

$$y = \alpha + \beta x$$

Where, α represents the intercept and β represents the slope

Dynamical downscaling involves using a RCM to simulate the local climate, driven by boundary settings proposed by the GCMs. This method is more computationally intensive and requires high-performance computing resources, but can provide more detailed and physically consistent results. Dynamic downscaling is established based on regional climate models (usually only the meteorological part) with an improved horizontal grid resolution of ground features such as terrain [49]. Here are the general steps for downscaling GCMs as input to RCMs [49]:

Choose the GCMs to use as input, based on their performance in simulating the large-scale atmospheric conditions that affect the target region. Then select the downscaling method that is most appropriate for the specific research question and available data. Obtain the boundary conditions needed to run the RCM. Then configure the RCM to simulate the local climate conditions, using the GCM boundary conditions and any additional local data available. Validate the downscaling results against observed data or higher-resolution data, and adjust the method. Lastly, use the downscaled data to study the influences of climate variation on the target region and to inform adaptation strategies.

2.4.4. Calibrating global data

Calibration in statistics adjusts the values of the parameters of a parametric model to assure that the model outputs data that matches as closely as possible empirically found data for a given set of input process [55]. The purpose of calibration is to reduce measurement ambiguity by ensuring the accuracy of test equipment. Calibration evaluates and controls faults and uncertainties in the measurement process to acceptable levels. This is done by comparing data from different sources and making adjustments to ensure that the data is comparable. This process is important for ensuring that calibrated data is consistent and used to make knowledgeable decisions. Calibrating global data for local weather prediction typically involves two main steps [56]: (1). downscaling the global data to the local level and (2) calibrating the downscaled data using local observations. The work of [55], stated some general steps that can be followed:

Identify the global dataset to be used for the analysis. This could be a satellite-based dataset or a global climate model output. Then downscale the global data to the local level. This involves using statistical or dynamical methods to transform the coarse-resolution global data into high-resolution local data. Compare the downscaled data with local observations. This involves analysing the statistical variables of the downscaled data, such as the variance, mean, and correlation, and comparing them with the corresponding properties of local observations. Calibrate the downscaled data using local observations. Then validate the calibrated downscaled data.

It's important to note that climate models are not perfect and there is always some degree of uncertainty in their predictions. However, they are an important tool for understanding and predicting future climate and weather conditions, and can be used to inform planning and decision-making in a variety of contexts, from agriculture and water management to disaster preparedness and infrastructure development. Rainfall forecasting is a difficult task because of the complex and nonlinear patterns of the atmospheric system. Though, most existing methods for rainfall forecasting rely on physical models that require a lot of data and computational resources, or statistical models that have limited generalization ability and accuracy.

2.4.5. Deep Learning Weather Prediction (DLWP)

Deep Learning Weather Prediction (DLWP) is a data-driven technique. The initial dataset is input to the DNN model. DNN models are designed to extract basic rules and relationships from initial data and capture the nature of climate variation from vast amounts of data. Conventional method is anticipated to benefit greatly from the addition of DLWP. Many researchers are currently trying to integrate data-driven deep learning into weather prediction, with some preliminary results [29]. Researchers have also used convolutional neural networks to develop a machine-learning weather forecasting system called deep learning weather forecasting (DLWP). This model is trained using historical weather data. This differs from standard numerical weather models, which build mathematical representations of the laws of physics.

Furthermore, artificial neural networks (ANNs) have been created with the objective of implementing artificial systems that can perform intelligent tasks similar to the human brain. It is a dominant data modeling techniques that can catch and represent complex interactions among inputs and outputs. Overall, ANNs can estimate any nonlinear task [10]. A deep neural network (DNN) is a type of artificial neural networks that consists of a multi-layered architecture and can reconstruct the raw data set from the initial feature space into the learned feature space. In other words, instead of manually selecting features, a neural network (NN) [57] can be used to "learn" the features and achieve greater precision and better overview with the learned features. Deep learning (DL) has accomplished promising results in several areas, including: in computer vision [58], natural language programming [61], speech recognition [59],[60], and the scientific field of physics [62],[63], bio-information [65], and chemistry [64].

Recently, deep learning (DL) has been practiced to the study of time series problems [66]. In this problem, the interaction between features is clear but difficult to identify. Especially when system behavior is governed by spatial or temporal situations (such as weather systems), conventional machine learning (ML) approaches may not be ideal, but spatio-temporal features can be automatically extractable by DL approach is good to understand the system [67]. The predicted accuracy will increase if the association is evidently examined and the attributes are characterized. As a result, DL has gained acceptance as an acceptable and useful method for examining the properties of time series.

Deep neural networks with many hidden neural layers and good feature learning can overcome the complication of trainings due to layer-by-layer initialization and achieve universal optimization of the network [68]. From the perspective of DLWP's DNN architecture, models can be classified into the following three groups: [68].

Architectures that constructed on the simple DNN models, such as those based on Auto-encoders [69], [70], CNNs[71], and LSTM networks [67]. The second category is hybrid models composed of the basic DNN models to capture more multifaceted spatial and temporal features, which are totally data-driven. The third category is the combined architecture of DNN and NWP models, which is not only data-driven but also theory-driven. This is a new research topic aimed at developing forecasting performance.

2.5. Deep Neural Networks

2.5.1. The Concept of Deep Neural Networks

Deep neural network models were basically started by neurobiology. At a higher level, a biological neuron accepts numerous signals through synapses connects its dendrites, and releases a single stream of activity potentials into its axons. Multiple input complexities are decreased by classifying input patterns. Inspired by this intuition, an artificial neural network model consists of units that fuse multiple inputs to produce a single output. Neural networks destination brain-like functions and are based on basic artificial neurons. A non-linear function of the weighted sum of the inputs (e.g. max (0, value)). These pseudo-neurons are grouped into layers, with the output of one layer becoming the input of the next layer in the sequence.

A neural network is a system of interrelated nodes that functions relatively similarly to human neurons in terms of fundamental processing units. The network processing power lies in the inter-unit connection strengths, commonly referred to as weights, which are acquired through a process of adaptation or learning from a set of training patterns. Neural networks are often utilized in statistical analysis and data modeling as an alternative to conventional nonlinear regression or cluster analysis techniques [72]. A neuron receives input signals from numerous additional (afferent) neurons. Each node in the input layer of figure 2.1 stands for a different characteristic. To output the values of the neurons into the following layer, they are linearly mixed using trainable weights (represented by the connections between nodes). An activation function is used to feed these values into the model, adding some nonlinearity. Next, a word with a trainable bias is added. As a result, each layer's values (apart from the input layer) are specified as follows:

$$a_i^j = f(\sum\nolimits_k w_{k,i}^j a_k^{j-1} + b^{j-1})$$

Where; a_i^j - Activation value of the ith neuron in the jth layer,

f - Corresponds to the activation function of the current layer.

$w_{k,i}^j$ – Trainable weight joining the kth neuron in the prior layer with the ith neuron in the existing layer,

a_k^{j-1} – Activation of the kth neuron in the preceding layer,

b^{j-1} – Bias of the preceding layer

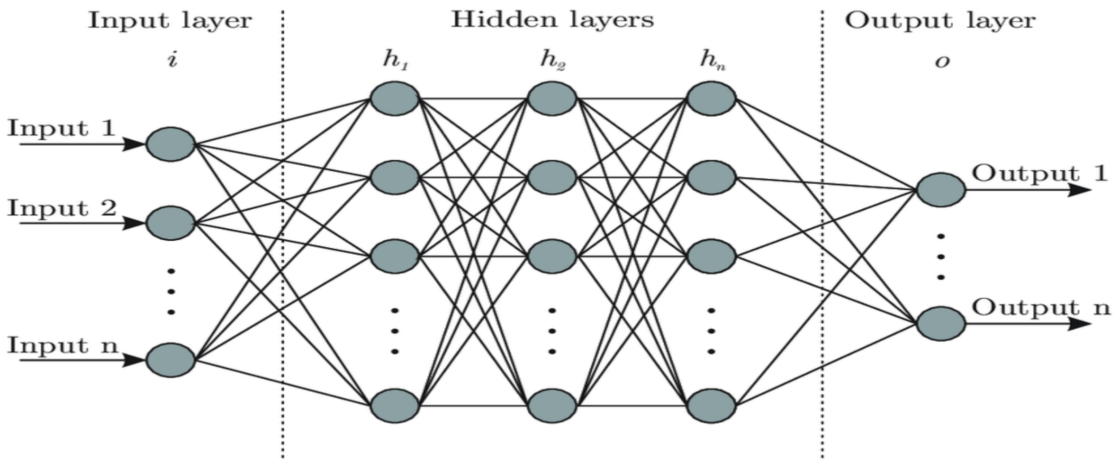


Figure 2.1| Basic Neural Network Structure, from [73]

The best way to describe artificial neural networks is as "computational models" with specific capabilities, such as the capacity to summarize, cluster, or reorganize data. Their function is also reliant on parallel processing. An artificial neural network consists of a group of fundamental processing units that communicate with each other by transmitting signals through numerous weighted connections. As illustrated in Figure 2.1, it is helpful to differentiate between three different types of units within neural systems: input units (indicated by an index i), hidden units (indicated by an index h), and output units (indicated by an index o), which send data outside the neural network. For 20 years, neural networks did not advance significantly; nevertheless, in 2006, interest in the idea of deep learning started to grow. Deep learning originally gained attention in the field of speech recognition. Since then, a variety of applications have been impacted by deep learning research. [74], and different techniques and models were used in this study and developed for time series predictions are discussed below.

2.5.2. Multilayer Perceptron’s (MLPs)

The perceptron, which lacks hidden layers, serves as the fundamental form of neural network architecture primarily employed for linearly separable pattern classification. As a supervised learning model, the perceptron operates as a binary classifier. This simplistic configuration of an Artificial Neural Network (ANN) was developed in 1957 by Frank Rosenblatt at the Cornell Aeronautical Laboratory, with funding from the United States Office of Naval Research [75]. It is important to note that a single-layer perceptron is incapable of being trained to recognize multiple classes of patterns; it can only learn linearly separable patterns. Subsequently, it was realized that a feed-forward neural network with two or more hidden layers possesses superior processing capabilities compared to a single-layer perceptron [76].

The classification of linearly distinct sequences is one of the main applications of perceptron, which are the most fundamental type of neural network design with no hidden layers. The perceptron model, often known as a binary classifier, belongs to the field of supervised learning. The perceptron method is the most basic ANN configuration ever developed. Frank Rosenblatt created it at the Cornell Aeronautical Laboratory in 1957 with funding from the US Office of Naval Research [75]. Single Layered Perceptron is capable of learning linearly separable associations; it cannot be trained to distinguish several classes of pattern. Later, it was shown that any feed-forward neural network with two or more hidden neural layers had the fastest processing speed [76].

A type of feed-forward artificial neural network is the multilayer perceptron (MLP). An MLP model, which consists of a number of fully linked layers, is the most basic type of deep neural network. MLP machine learning algorithms can outperform the high resource requirements of contemporary deep learning frameworks [77]. The three node layers that make up an MLP are the input layer, the hidden layer, and the output layer. Every node is a neuron with a nonlinear activation function, with the exception of the input nodes [78]. Back-propagation is a method of continually adjusting the weighted and threshold values in a multilayer perceptron network in order to reduce the discrepancy among the targeted/desired output and the achieved output [79]. It is also linked to neural nodes or hidden layers that are layered between inputs and outputs. More difficult features (like XOR logic) are frequently learned with the support of the hidden layers [80].

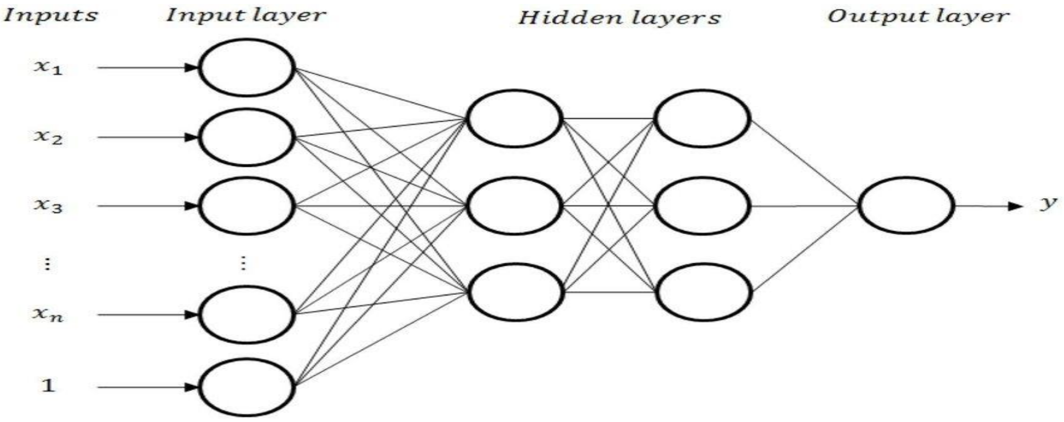


Figure 2.2| Architecture of Multi-layered Perceptron [81]

Figure 2.2 depicts one input layer, levels of two hidden, and also the final output layer. Each layer has completely connected. This indicates that the nodes from the layer below are related to the current node. Each layer has a weight matrix that contains all of the weight for that layer. The non-linear nature of the function we employ in the nodes implies that the output will not rely linearly on the input. In other words, just because the input goes up by 10% doesn't guarantee that the output will follow the trend; it may go up by more than 10% or less than 10% [81].

The feature is the total amount of input data that is fed into the network. Therefore, if we consider $X_1, X_2, \dots, X_n$ that means we are having 'n' characteristics. A feature is a characteristic whose impact on output can be seen. The total of all the weighted characteristics is then computed by multiplying these features by their respective weights, giving us the dot product.

$$\sum_{l=1}^{m} w_l x_l = w_1 x_1 + w_2 x_2 \dots \dots w_m x_m$$

The acquired dot product is next combined with the bias to produce Error. The reference source not found is then used as input to the activation function, which produces the output for the first neuron in the first hidden layer h_1.

$$Z = \sum_{l=1}^{m} w_l x_l + bias$$

Similar steps are taken for the second and third neurons, h_2 and h_3 of hidden layerh_1, respectively. If a second hidden layer is present, its inputs are obtained from the first hidden layer's neurons' outputs, and vice versa.

2.5.3. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNN or ConvNet) are another type of deep neural network. The greatest widespread use of CNN is in computer vision. The Artificial Intelligence system learns to automatically extract the properties of these inputs to finish a given goal, such as picture classification, face authentication, and image semantic segmentation, with the use of a set of real-world photos or videos and CNN [82].. One or more convolution layers in CNN models employ convolution operations to extract the basic properties from input, in contrast to completely connected layers in MLPs. Each layer is made up of a group of nonlinear functions that compute weighted sums of the outputs from the layer before that are located at different points and are geographically close by.

A typical convolutional neural network consists of blocks of single or multiple convolutional and pooling layers followed by one or more fully connected (FC) and output layers. The fundamental building blocks of CNNs are convolutional layers. The goal of this level is to learn a feature representation of the input. The learnable convolutional kernels or filters used in convolutional layers are used to create various feature maps. A reaction field from the previous level is connected to each entity in the feature map. A convolutional neural network (CNN) generates a new feature map by convolving the input with a kernel and subsequently applying a nonlinear activation function element-wise to the convolution result. The convolutional output is then sampled or pooled, where a reduced representation of the input is produced as a single output.

There are numerous sub-sampling methods, including minimum pooling, maximum pooling, average pooling, and others. Pooling makes the network transformation invariant and lowers the number of parameters that must be calculated. Essentially, the last component of a CNN is built up of one or multiple FC layers, which are commonly present in a feed-forward neural network. The last pooling or convolutional layer provides input to the FC layer, which creates the final output of CNN. The advancement of CNN is dependent on a number of other characteristics in addition to layer design, comprising the activation function, regularization, optimization, loss function, and processing speed [83]. Figure 2.3 depicts the fundamental conceptual model of CNN, and the following sections cover several layer types.

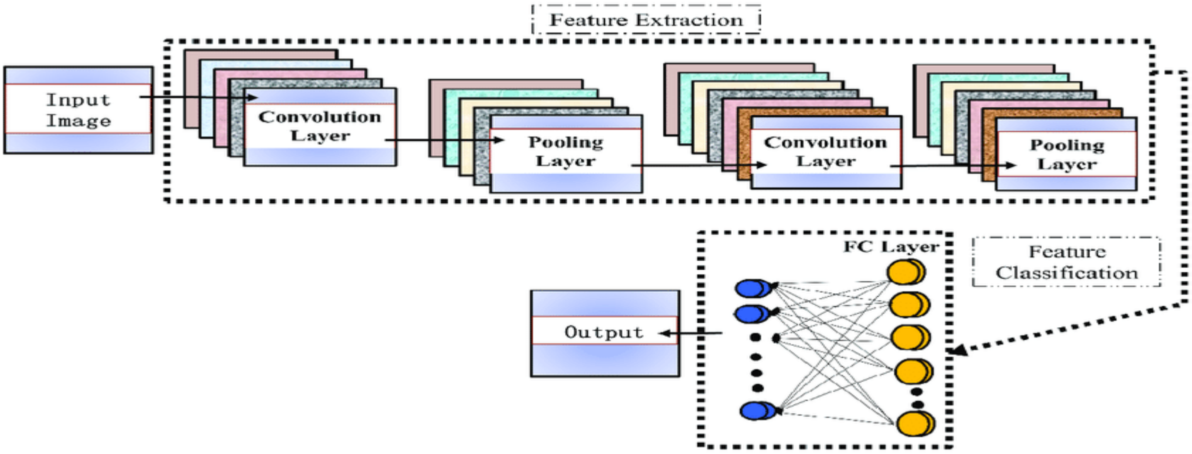


Figure 2.3| Conceptual model of CNN [84]

The pooling layer in a convolutional neural network (CNN) performs sub-sampling on the generated feature maps obtained from convolution operations. Its purpose is to reduce larger-size feature maps into smaller ones while preserving the most significant characteristics or information at each pooling stage [84]. The pooling operation, similar to convolution, involves defining the size of the pooled region and the stride of the operation. Different pooling techniques such as max pooling, min pooling, average pooling, gated pooling, tree pooling, among others, are employed by various pooling layers.

Assume that our convolutional layer comes after a layer of $N \, x \, N$ square neurons. The output of the convolutional layer will be of size$(N-m+1)x\,(N-m+1)$ if we employ a $m \, x \, m$ filterω. In order to determine the pre-nonlinearity input to a specific unit x_{ij}^{l} in the layer, we need to add the contributions from the preceding layer cells, weighted by the components of the filter.

$$x_{ij}^{l} = \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} \omega ab y_{(i+a)(j+b)}^{l-1} \qquad -------(1)$$

Then, the convolutional layer applies its nonlinearity:

$$y_{ij}^{l} = \sigma\left(x_{ij}^{l}\right) \quad -----------(2)$$

The simple max-pooling layers are unable to learn for themselves. They just produce a single result, which is the maximum in the given $k \, x \, k$ rectangle. As an illustration, if their input layer is a $N \, x \, N$ layer, they will produce a $\frac{N}{k} \times \frac{N}{k}$ layer as each $k \, x \, k$ block is condensed by the max function to a single value.

Assuming we have an error function, E, in the Backpropagation convolutional layer, the error we know and need to compute for the preceding layer is the partial of E with respect to each neuron output $(\frac{\partial E}{\partial y_{ij}^{l}})$. Noting that we must total the contributions of all expressions in the chain rule where the

$$\frac{\partial E}{\partial \omega_{ab}} = \sum_{i=o}^{N-m} \sum_{j=0}^{N-m} \frac{\partial E}{\partial x_{ij}^{l}} \frac{\partial x_{ij}^{l}}{\partial \omega_{ab}} = \sum_{i=0}^{N-m} \sum_{j=o}^{N-m} \frac{\partial E}{\partial x_{ij}^{l}} y_{(i+a)(j+b)}^{l-1} ----- (3)$$

$$\frac{\partial x_{ij}^{l}}{\partial \omega_{ab}} = y_{(i+a)(j+b)}^{l-1} -------------- (4)$$

$$\frac{\partial x_{(i-a)(j-b)}^{l}}{\partial y_{ij}^{l-1}} = \omega_{ab} ------------- (5)$$

$$\frac{\partial E}{\partial y_{ij}^{l-1}} = \sum_{a=0}^{m-1}\sum_{b=0}^{m-1}\frac{\partial E}{\partial x_{(i-a)(j-b)}^{l}}\frac{\partial x_{(i-a)(j-b)}^{l}}{\partial y_{ij}^{l-1}} = \sum_{a=0}^{m-1}\sum_{b=0}^{m-1}\frac{\partial E}{\partial x_{(i-a)(j-b)}^{l}}\omega_{ab} --(6)$$

The pooling layer's primary drawback is that it occasionally makes CNN perform worse as a whole. This is because, without worrying about the feature's exact location, the pooling layer enables CNN to determine if a certain feature is present in the input picture or not. After each learnable layer (a layer with weights, such as a convolutional or FC layer) in the CNN architecture, non-linear activation layers are used [84]. The CNN model can learn more complex ideas and successfully transform inputs to outputs in a non-linear way because of the non-linear behavior of those layers. It is crucial for an activation function to be differentiable in order to allow the model to be trained via error back-propagation.

2.5.4. Recurrent Neural Networks

Recurrent neural networks, are a type of artificial neural networks, the output from specific nodes can impact the subsequent input to those same nodes. This occurs due to the presence of connections between nodes that can form cycles. Recurrent neural networks (RNN) feed the outputs from earlier states into the present state. The buried layers of an RNN can retain data. The concealed state is updated based on the output generated in the preceding stage. Unlike conventional neural networks, RNNs make advantage of the network's sequential data. In many situations where the inherent structure in the data sequence gives crucial knowledge, this trait is crucial. For example, comprehending a word within a sentence requires knowing its context. This means that an RNN, which consists of the input layer x, the hidden (state) layer h, and the output layer o, may be assumed of as a short-term memory unit [85].. An RNN's fundamental structure is depicted in Figure 2.

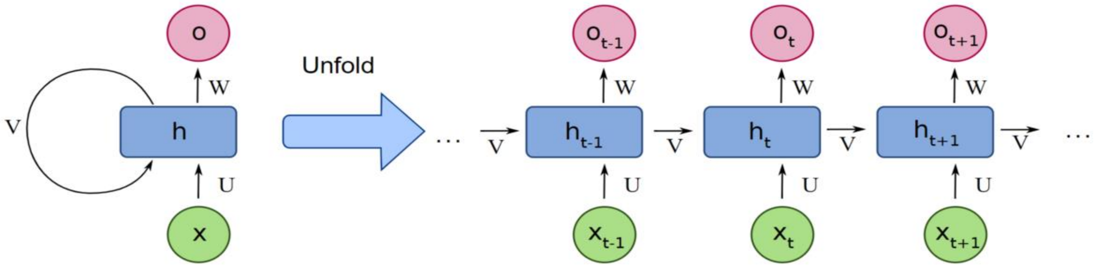


Figure 2.4| Unfolded RNN Structure

Similar to feed-forward and convolutional neural networks (CNNs), recurrent neural networks (RNNs) utilize training data to acquire knowledge. Their "memory," which enables them to use information from previous inputs to alter both the current input and output, sets them apart from other systems. Unlike typical deep neural networks, which presume that inputs and outputs are independent of one another, recurrent neural networks' outputs are reliant on the preceding parts in the sequence. The back-propagation through time (BPTT) technique, which is slightly different from conventional back-propagation since it is designed for sequence data, is used by recurrent neural networks to compute the gradients.

It takes modeling linkages and patterns in time series data to predict future development. In the past, time series such as market projections [86], network traffic forecasting [87], and weather forecasting [69] have all been predicted using recurrent neural networks (RNNs). RNNs frequently employ a sequence of prior time steps to forecast the subsequent step. Very huge datasets that cannot be presented all at once are created when predictions are made for each time step of the testing data. A distinct collection of hyper-parameters, such as the number of neurons, layers, or prior time steps, can be used to train each RNN. Over various parts of the training and test data, the trained RNNs' prediction error fluctuates both geographically and temporally [88]. Neural networks have recently made significant advancements across a wide range of scientific fields. Their ability to acquire non-linear relationships, which frequently outperforms manually crafted features, is their main strength. However, neural networks have some restrictions. It’s challenging to modify their hyper-parameters and to deduce how and why a network functions, making theoretical predictions challenging [88].

Due to their memory function, recurrent neural networks (RNNs) are frequently utilized for time series prediction issues [89]. Using RNNs, large datasets that can't be displayed all at once can be analysed incrementally [88]. Standard RNNs are challenging to train, though, because of the well-known gradient vanishing and exploding phenomenon [90], particularly when applied to problems with long-term dependencies. These issues are categorized according to the magnitude of the gradient or the slope of the loss function along the error curve. The weight parameters are updated until they are statistically insignificant, or 0, when the gradient is too tiny. At that point, the algorithm stops learning. Exploding gradients result from an excessive gradient, which renders the model unstable. In this case, the model weights eventually grow to be excessively big and take the form of Nan. In contrast to deep CNNs, which can have over 100 layers, the majority of RNNs only have 2 or 3 layers [91]. Long-short-term memory (LSTM) network has been proposed as an RNN version to overcome the gradient issues with ordinary RNNs [92].

2.5.5. Long-Short Term Memory

Sepp Hoch Reiter and Juergen Schmidhuber first introduced the Long Short-Term Memory (LSTM) architecture as a solution to the vanishing gradient problem [93]. It is currently extensively utilized because of its better performance in properly modeling both short and long term dependencies in data. Long short-term memory (LSTM) in RNN has become a powerful and scalable model for a variety of learning issues involving sequential data. However, LSTMs are currently employed to address a variety of learning issues that, in terms of scale and type, are very unlike from those on which these advances were first evaluated. The heart of the LSTM design consists of a memory cell that can reserve its state over time and nonlinear gating devices that manage information flow into and out of the cell [92]. While attempting to tackle the vanishing gradient problem, LSTM maintains constant error owing back over time without imposing any bias towards recent data.

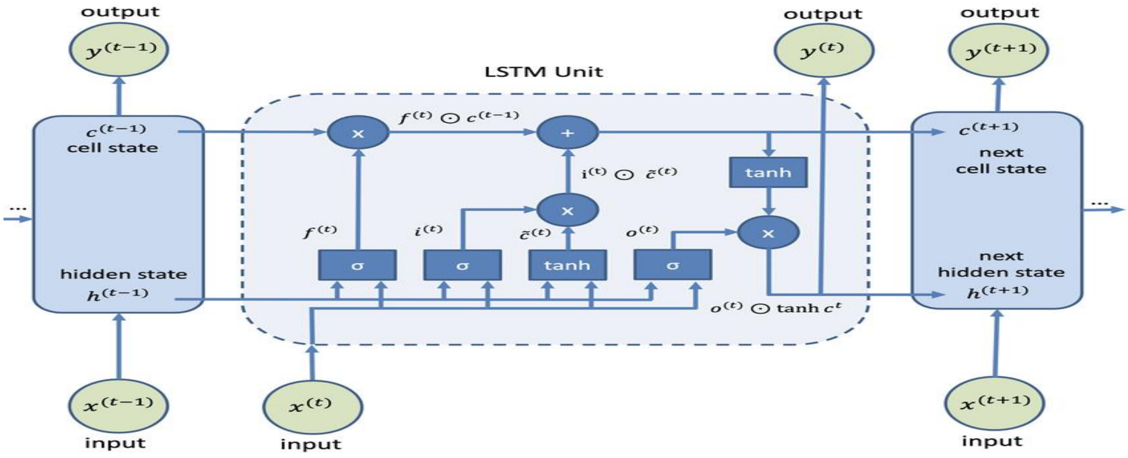


Figure 2.5| The architecture of LSTM

Input and output gates, (potentially multiple) cells, not forget gates (NFG) or peepholes (NP) connections, and (maybe) numerous cells were all included in the original LSTM block [94],[95]. The input activation functions, unit biases, and output gates were not included in certain tests. Back-propagation through time (BPTT) [96] and real-time recurrent learning [95] were both applied during the training phase. In the context of recurrent neural networks, only the gradient of the cell is sent backward through time, while the gradients for the other recurrent connections are truncated. Another characteristic of that version was full gate recurrence (FGR), which meant that all the gates got recurrent inputs from all the gates at the preceding time step in addition to the recurrent inputs from the block outputs. LSTM has problems with long term dependency. To solve this problem some internal state is added. The state is an actual cell, which consists of three parts. Each part is a gate, which is discussed below;

Input Gate: The input gate has the ability to update the cell's state with new information. The fundamental three-step process for this information addition is shown in the diagram above. Sigmoid function controls over the need of which values should be added to the cell states. This procedure acts as a filter for the combined data from h $^{(t-1)}$ and x $^{(t)}$, and it shares major similarities with the forget gate.

$$i_t = \sigma(\omega_i[h^{t-1}, x^t] + b_i$$

Forget Gate: The forget gate is responsible for deciding how much of the previous cell state should be forgotten or retained. A forget gate deletes information related to the cell state. When a filter is multiplied, it serves to discard information that is no longer necessary for the LSTM to comprehend or information that holds lesser significance [97]. This is essential for enhancing the functionality of the LSTM network.

$$f_t = \sigma(\omega_f[h^{t-1}, x^t] + b_f)$$

The gate accepts h^{t-1} and x^t. h^{t-1} as its two inputs, as shown in figure 2.5 above. The inputs at that specific time step arex^t, whereas the hidden state from the previous cell, or its output, ish^{t-1}. Before a bias is imposed, the weight matrices are multiplied by the inputs. Subsequently, this value undergoes the sigmoid function transformation. The sigmoid function is employed to express each value in the cell state as a vector with values ranging between 0 and 1. For instance, if the forget gate yields a '0' for a specific value in the cell state, it signifies that the forget gate intends to discard or eliminate the corresponding information.

Output Gate: Not all data that travels with the cell state is appropriate for output at a specific moment. The values are scaled to the range of -1 to +1 when the Tanh function is applied on the cell state. h(t-1) and x(t) values are then used to form a filter that controls the values that must be produced from the vector established previously. Once more using a sigmoid function, this filter multiplies the result of the regulatory filter by the vector that is produced before delivering it as an output and also to the following cell's hidden state.

$$o_t = \sigma(\omega_o[h^{t-1}, x^t] + b_o)$$

In general, each LSTM block (as seen in Figure 2.4) has three multiplicative units and one or more recurrently linked memory cells [98],[99]: The LSTM has three gates: the input gate, which receives the current x^t and controls whether it considers its current input; the forget gate, which enables the LSTM to forget its prior memory h^{t-1}, which is essential for solving gradient problems; and the output gate, which determines how much memory to transfer to the hidden state $h^{(t)}$, or what to output. The long-term sequences required for long-term weather forecasting and climate modeling can be captured by LSTM thanks to this architecture.

2.5.6. Bidirectional Long-Short Term Memory (Bi-LSTM)

A BiLSTM, which stands for bidirectional Long Short-Term Memory, is a model designed for sequence processing. It comprises two LSTMs, with one LSTM processing input in the forward direction, while the other LSTM processes it in the backward direction. By successfully increasing the network's informational pool and the algorithm's context (by, for example, knowing what words immediately follow and come before a given word in a phrase), BiLSTM improves network performance. The BiLSTM utilizes information from both directions, enabling the input to flow in both forward and backward directions. This unique characteristic distinguishes it from the regular LSTM. By leveraging the bidirectional nature of the sequence, the BiLSTM becomes a powerful tool for capturing the sequential relationships among words and sentences [100].

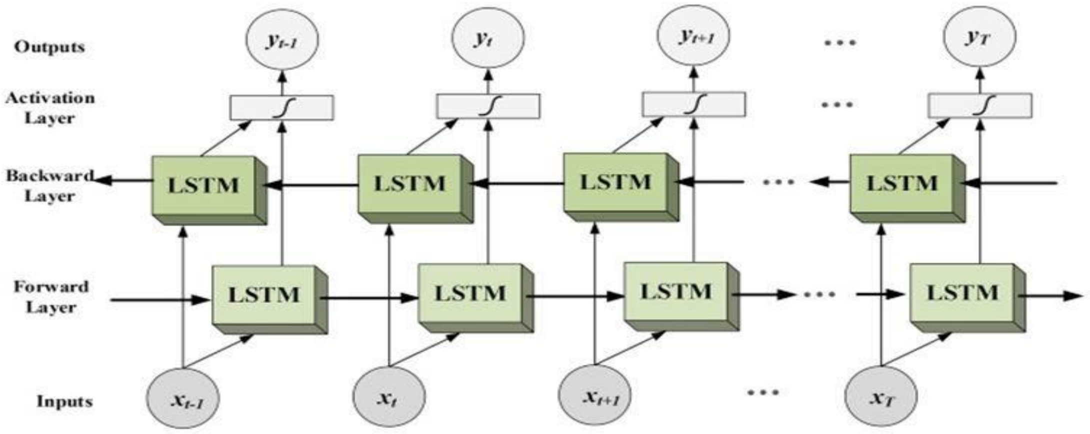


Figure 2.6| The architecture of Bidirectional LSTM [100]

In conclusion, BiLSTM introduces an additional LSTM layer that reverses the direction of information flow. This extra layer processes the input sequence in a backward manner. The outputs generated by both LSTM layers are subsequently combined using various methods such as sum, average, multiplication, or concatenation, offering flexibility in information integration.

2.6. Related Literature

The development of prediction techniques comes a long way, from depending on an individual's experience to simple numeric methods to complex atmospheric models. While artificial neural networks (ANN) and other machine learning algorithms have been used by researchers to predict rainfall, researches on the usefulness of existing deep learning approaches are limited, especially on data collected from both grid satellite data and ground weather stations. Rainfall forecasts can be made for a small amount of time, such as an hour, a day, or a week in the future, or for a long amount of time, such as a month, a season, or a year in advance. A Neural Network (NN) is a group of neurons with several hidden layers that functions much like the human brain.

The paper[22] conducted a research on; "Prediction of Rainfall Using Data mining technique over Assam" in order to develop a model for summer monsoon season rainfall prediction using empirical statistical technique. They collected 6 years datasets from regional meteorological Centre, guwahati, Assam, India, between2007-2012. Parameters such as maximum and minimum temperature, wind direction, pressure, relative humidity etc. were included and they conducted rainfall prediction using Multiple Linear Regression (MLR). In the study the researchers were used different methods to combine rain gauge observations and satellite based rainfall estimates data. The accuracy of the model was evaluated by using adjusted R-squared and their result shows that the developed model based on MLR indicates acceptable accuracy. Their findings show that, accuracy of 63% in the variation of rainfall for their proposed model to predict monthly rainfall.

In order to train the Global Average Monthly Rainfall data using Deep Learning architectures (LSTM and ConvNet) and to produce precise prediction results in the test data with a minimum amount of error, [101] proposed "Deep Learning Models for the Prediction of Rainfall" in April 2018. Researchers confirmed that deep learning technologies are distinct from neural networks showing that the models have been altered to account for the addition of more hidden layers and improved performance. By combining precipitation estimation from microwave data, infrared data, and rain gauge observations, a monthly precipitation dataset that spans the globe from July 1979 to January 2018 was compiled. The performance of generated models was assessed using the Mean Absolute Percentage Error (MAPE) and the Root Mean Squared Error (RMSE). With an RMSE of LSTM at 2.55 and an RMSE of ConvNet at 2.44, the models have an outstanding design.

The research work done by [102], proposed rainfall forecasting models for annual and non-moon sessions in Odisha (India) using regression techniques. Two distinct regressions processes; one for yearly data and the other for eight months of non-monsoon sessions have been used in the article. Two machine learning methods are SVM and MLP have been used to do the regression analysis. The maximum average monthly temperature, minimum average monthly temperature, relative humidity, and annual rainfall (mm) were collected between 1991 and 2005 in the state of Odisha. By comparing the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Correlation coefficient (r) data, the models are compared. The study demonstrated that SVM model performed better than MLP for predicting yearly rainfall in Odisha, but that MLP performed better for predicting rainfall during the five non-monsoon months of January through November. For the three non-monsoon months, prediction rainfall will be more accurate if Multi-Layer Perceptron was used.

The work of [103]; developed deep learning model to predict rainfall based on weather variables recorded by the weather stations in Jimma. They suggested a prediction model based on Long Short-Term Memory (LSTM) that could predict daily rainfall for Jimma. The researchers used the mean of maximum temperature, minimum temperature, relative humidity, solar radiation, wind speed, and precipitation to derive characteristics from meteorological data. RMSE, MAPE, Nash-Sutcliffe model efficiency (NSE), and R2 were used in experiments to evaluate the proposed models, and the results were 0.01, 0.4786, 0.81, and 0.9972, respectively. The proposed model was also contrasted with other machine-learning models, including the Multilayer Perceptron (MLP), k-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Decision Tree (DT). The results of the experiments demonstrate that the suggested model has a higher R2 and a lower RMSE.

The study of [104], Proposed "Bidirectional Long Short Term Memory Gated Recurrent Unit (BLSTM-GRU) model for monthly rainfall prediction over Simtokha a region in the capital of Bhutan, Thimphu". The developed model has composed of 7 layers together with the input and output layers. The datasets contains daily records of weather parameters are collected from the national center of hydrology and meteorology department (NCHM) from 1997 to 2017. Machine learning and deep learning models are trained using data from 1997 to 2015, and models are tested using data from 2016 to 2017. They compared their own results with Linear Regression, MLP, LSTM, CNN, and Gated Recurrent Unit (GRU) based on the metrics recorded by the local automated weather station. With MSE, RMSE, R2, and correlation coefficient values of 0.0075, 0.087, 0.870, and 0.938, respectively, the proposed BLSTM-GRU model outperformed LSTM on all four performance matrices.

The study of [20], Proposed a "Rainfall forecast model based on the Attentive Interpretable Tabular Learning neural network (TabNet)". The work employed self-supervised learning to boost convergence and stability of the TabNet model. The suggested rainfall forecast model was tested using a five-year period of meteorological data from 26 stations in the Beijing-Tianjin-Hebei area of China. The data was gathered daily from January 2012 to December 2016. They chose features via TabNet's instance-wise feature selection technique. In order to increase the model's accuracy, they merged conventional techniques with machine learning techniques. They also utilized feature engineering techniques to teach the model how rainfall varies seasonally. The constructed models were evaluated using the modified Kling-Gupta efficiency (KGE), mean absolute error (MAE), random error (RMSE), and mean absolute percentage error (MAPE). Comparative tests revealed that the suggested model performed better.

The study conducted by [30], Proposed an Artificial Neural Network model that employed on Map-reduce framework for short-range rainfall prediction. Implementing an Artificial Neural

Network using a Hadoop-based solution makes it quicker and more scalable. The study used original gridded format climate daily statistics based IMD data over the previous 63 years (from 1951 to 2013) of temperature and precipitation values were utilized as input for developing, verifying, and testing the system. For training, the first 45 years of data were used, the following 6 years were used to validate several models, and the final 12 years were used for testing. Rainfall data was in 0.25 × 0.25 degree grids, while temperature data was in 1 × 1 degree grids. Both temperature and rainfall data were at distinct resolutions. The temperature and rainfall files were compared, and the grid resolution was changed accordingly. Additionally, missing values were removed from the newly organized files. So, map-reduced was used at the data pre-processing stage itself. The location-specific temperature and rainfall data were extracted from the source files using mappers. Reducers were employed to write the data to new files in the required format. In the study, ANN was used to forecast the weather for the coming day.

The work of [71], a multitask convolutional neural network model was proposed to automatically extract features from the time series data obtained at observation sites and utilize the correlation between the various sites for multitasking weather prediction. A feature transformation network (CNN model) to represent the input features, an output model to forecast rainfall amounts, and a multi-task module to take site correlations into account are the three key parts of the proposed approach. The input characteristics, also known as multi-site features, were obtained by the researchers from neighbouring sites or a network of rain gauges. They used two real-world datasets to assess the suggested technique. The first dataset was the total amount of rain that fell each day at a meteorological station in Manizales, while the second was a large-scale rainfall dataset gathered by the observation sites throughout China's Guangdong region. The study demonstrates how to estimate rainfall using a collection of meteorological variables gathered from various nearby observation locations. The experiment demonstrated that the suggested strategy outperforms numerous cutting-edge deep learning-based rainfall prediction models, which served as the baseline approaches.

The study of [105], proposed an AI and LSTM models that can easily predict the rainfall amount. Data were collected from six regions with six variables (temperature, wind speed, humidity, dew point, wind pressure, and wind direction). The study data obtained from Bangladesh Meteorological Department (BMD). For training purposes, the most recent months' worth of monsoon season meteorological data (from August 1 to August 31, 2020) was gathered. They discovered 76% accuracy in their work after applying LSTM and RNN to analyze all collected data.

The work done by [106], proposed an ensemble learning methods to improve the efficiency of rainfall prediction. The Malaysian Meteorological Data, which consists of 1,581 instances was organized into two classes, namely "active rainfall" and "no rainfall," and was used as the basis for the ensemble learning approach, which combined multiple ML and multiple rainfall prediction classifiers, including Naive Bayes, Decision Tree, Support Vector Machine, Random Forest, and Neural Network. According to the test dataset, the study compares the performance of the five classifiers using the three criteria. The neural network surpasses the other methods among the classifiers, with accuracy of 72.7%, recall of 74.5%, and F-Measure of 73.2%. The ensemble's ability to further raise the upper bound on the rainfall prediction model was further investigated in the study. The study of the ensemble experiment showed that the ensemble approach does, in fact, improve rainfall prediction.

The research work done by [107], was identify the relevant atmospheric features that cause rainfall, and predicted the intensity of daily rainfall using machine learning techniques. The study's machine learning model received relevant environmental variables that were chosen using the Pearson correlation approach. The performance of three machine learning techniques (Random Forest, Multivariate Linear Regression, and Extreme Gradient Boost) was assessed by Mean Absolute Error and Root Mean Squared Error methods after the authors collected data from the local meteorological office in Bahir Dar, Ethiopia. According to the performance data, XGBoost Gradient descent performed better than MLR and RF. The XGBoost method predicted the rainfall utilizing important, chosen environmental factors better than the RF and the MLR. The MAE and RMSE values of the XGBoost gradient descent techniques were 3.58 and 7.85, respectively.

In the work of [78], a novel model architecture was proposed, integrating the Convolutional Neural Network (CNN) with a ResNet-152 pre-training model and the Recurrent Neural Network (RNN) with a Long Short-term Memory Network (LSTM) layer for training purposes. The researchers encoded satellite images using CNN and extracted image feature vectors during the training process. These vectors, along with meteorological data, were used as inputs for the RNN. While satellite weather imagery is captured from space, the researchers obtained ground-based cloud images at one minute intervals for weather forecasting. To assess performance, they compared three methods: CNN alone, LSTM alone, and their proposed CNN-RNN approach.

In the study conducted by [108], a comparative analysis was performed between the LSTM neural network and the wavelet neural network (WNN) to predict spatio-temporal trends of rainfall and runoff time-series in hydrologic basins with limited gauge data. Satellite meteorological data were acquired from the NCEP-CFSR website. Both models employed an identical deep learning architecture, consisting of four hidden layers, each comprising 30 neurons. Utilizing the satellite-based meteorological data, the LSTM model achieved a higher accuracy in predicting mean monthly rainfall within the basin, yielding an R2 value of 0.8610, as opposed to 0.7825 obtained by the WNN. This outcome suggests that optimized LSTM and wavelet neural network models, incorporating satellite data, can be relied upon for rainfall and runoff trend predictions in basins where hydrological and meteorological data are scarce or of low quality. The researchers further recommended the inclusion of basin physical characteristics such as slope, elevation, and flow accumulation as additional training inputs in future studies.

2.6.1. Related work review Summary

Table 2.1| Summary of related work

Author, Title & Year	**Objective**	**Methodologies**	**Key Findings**	**Recommendation & Future Work**
S. Aswin et al. [101], “Deep Learning Models for the Prediction of Rainfall” (2018)	Developing rainfall prediction model using deep learning approaches	LSTM and ConvNet were used	The error rates observed in both the LSTM model and ConvNet model were nearly identical without the use of GPU. LSTM shows 2.55 RMSE and ConvNet shows 2.44.	The research could integrate precipitation estimation methods that utilize microwave data, infrared data, and rain gauge observations.
P. S. Dutta and H. Tahbilder[22], “Prediction of Rainfall Using Data mining Technique over	To develop data mining model that predict rainfall over Assam	Multiple Linear Regression was used	Their proposed model achieved an accuracy of 63% in predicting	The work can be extended for multiple stations in future

Assam" (2014)			variations in rainfall.	
Zhang et al. [102], "Annual and Non-Monsoon Rainfall Prediction Modelling Using SVR-MLP: An Empirical Study From Odisha" (2020)	Developing rainfall prediction model for annual and non-monsoon session	Multilayer perception and Support Vector Regression	The study concluded that, when it came to predicting annual rainfall, SVM model showed higher performance compared to MLP.	They recommended that, SVR yields better accuracy in non-monsoon rainfall prediction for Jan, Feb, April, Oct, and Nov, while MLP outperforms in the remaining three months.
Endalie et al. [103], "Deep learning model for daily rainfall prediction: case study of Jimma, Ethiopia" (2022)	To develop a predictive model using LSTM for daily rainfall forecast for Jimma.	Deep learning algorithm-LSTM	The LSTM-based rainfall predictive model was considered suitable for various applications, including smart agriculture, demanding accurate rainfall prediction.	For future they aim to develop a rainfall prediction model that includes sea-surface temperature, Climate indices, and global wind circulation.
Manoj et al. [104], "Deep BLSTM-GRU Model for Monthly Rainfall Prediction: A Case Study of Simtokha, Bhutan" (2020)	To develop and propose deep BLSTM-GRU model for monthly rainfall prediction.	Bidirectional Long Short Term Memory Gated Recurrent Unit (BLSTM-GRU) was proposed	The combination of BLSTM and GRU layers demonstrated greater performance on the collected dataset compared to all other models.	For future they recommended to enhance the predictive model by integrating global and regional weather patterns, such as sea surface temperature and global wind circulation.
Jianzhuo et al. [20], "Rainfall Forecast Model Based on the TabNet Model" (2021)	To minimize rainfall prediction errors, by combining satellite-based rainfall observation with machine learning prediction	(TabNet)-Attentive Interpretable Tabular Learning neural network	The study improved model accuracy by combining conventional methods with machine learning and utilized feature selection to capture seasonal rainfall patterns, resulting	In future research, the researchers suggested employing larger datasets, optimizing parameters, and adopting more effective feature engineering methods to enhance the model's robustness.

	techniques.		in the higher performance of the proposed model.	
Namitha et al. [30], “Rainfall Prediction using Artificial Neural Network on Map-Reduce Framework” (2015)	To Implement an Artificial Neural Network model on a map-reduce framework for short-term rainfall prediction.	ANN that implemented on Map-reduce framework	The finding indicate that the map-reduce framework effectively reduces the runtime of the proposed model. Moreover, this framework remains reliable for rainfall forecasting, even with terabytes or petabytes of data.	To achieve a more accurate rainfall prediction model, future work should involve exploring different map-reducible machine learning algorithms.
M. Qiu *et al.* [71], “A Short-Term Rainfall Prediction Model using Multi-Task Convolutional Neural Networks” (2017)	The objective of multi-task learning and deep learning methods is proposed to predict short-range rainfall amounts using multisite features.	Convolutional Neural Network (CNN)	The study concluded that employing a set of weather variables obtained from multiple nearby observation sites is beneficial for rainfall prediction.	The study recommended that employing a set of weather variables collected from multiple nearby observation sites for rainfall prediction.
C. M. Liyew and H. A. Melese. [107], “Machine learning techniques to predict daily rainfall amount.” (2021)	The objective is to identify the important atmospheric variables contributing to rainfall and employ machine learning techniques to predict the intensity of daily rainfall.	Multivariate Linear Regression, Extreme Gradient Boost, and Random Forest.	The results revealed that the XGBoost Gradient Descent algorithm outperformed MLR and RF. The RMSE and MAE values for XGBoost Gradient Descent were 7.85 and 3.58.	To improve the accuracy of rainfall prediction for future the author’s recommended using sensor with meteorological datasets with big data analysis.

2.7. Research gaps:

The methodology employed to predict rainfall in existing works was well established from literature on the machine learning and deep learning techniques like the Aswin et al.(2018)

[101] study. Researchers from diverse backgrounds have conducted numerous studies aiming to enhance the prediction accuracy of daily, monthly, and annual rainfall amounts by utilizing meteorological data from various countries. Different researchers conducted using data mining approaches [109],[102], [22] Big Data analysis [30], machine learning algorithms [78], [106], and different deep learning approaches [71], [99], [103], [104] to improve the accuracy of rainfall prediction for daily, monthly and annual. Based on the study findings, the process of rainfall prediction has transitioned from employing data mining techniques to machine learning techniques, and is currently progressing towards utilizing deep learning techniques.

Different approaches have been proposed to rainfall forecasts, and each approach has their advantage and limitations. There are still abundant gaps in this topic for further improvement and progress. The majority of the research listed above was conducted in developed countries, which typically have more access to advanced technologies, infrastructures, and comprehensive data collection systems. Locally, the study conducted by [103], was only used ground station data such as; Max temperature, Min temperature, solar radiation, humidity, and wind speed of the atmosphere over the south western Oromia. However, the study used only station data, as well as the correlations between each variable with rainfall did not measured to determine the impact of the weather features on the rainfall. Other local research conducted by [107], was to identify the atmospheric variables that contribute to rainfall and utilize machine learning techniques to predict the intensity of daily rainfall. As well, the study used only station data.

To overcome the limitations and address the research gaps highlighted in this section, this research employed a 30 years of combined satellite and station data, and it has used new features such as air pressure, cloud cover, and so on. This research has specifically addressed data preparation, handling missing values, and employed four deep learning approaches (MLP, CNN, LSTM, and BiLSTM) for daily rainfall prediction. The study selects the best performing approach to optimize model performance. Therefore, this research contributes to filling the research gaps regarding the prediction of rainfall using a combination of satellite and station data through deep learning methods.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1. Introduction

This chapter presents the research methodology employed in this study, including the process model used to organize its activities. The study follows the design science research methodology, with each step being thoroughly explained and detailed. Furthermore, the adopted processes for this study are discussed extensively.

3.2. Research approach

Research methodology is thought to be concerned with a researcher's approach to the examination of a particular topic of interest. A research methodology can also be thought of as an action plan, strategy, process, or design that guides the selection of research methodologies and links their application [110]. Fundamentally there are different categories of research approaches. For this thesis, a design science approach was proposed to answer the identified research questions. R. Buckminster Fuller coined the term "Design Science" in 1957; it is seen as a systematic form of designing and is concerned with knowledge acquisition related to designs and their activity [110]. Engineering and the science of artifacts are the foundations of the design science paradigm. The core focus of this study lies in problem-solving through the development of innovative ideas, practices, technical capabilities, and products that can be reached through analysis, design, implementation, and information system use [111]. Recently, made compelling arguments for the legitimacy and significance of design science (DS) as a research paradigm within the field of Information Systems (IS) succeeded in integrating design research into the IS research community ([112]; [113]) and actually incorporating design as major element of research [114].

"Design Science creates and evaluates IT artifacts intended to solve identified organizational problems" [[115], p. 77]. Design science research involves a rigorous process of developing artifacts to address identified problems, contributing to research, evaluating the designs, and effectively communicating the findings to relevant audiences [115]. Hevner in this work [115] provided seven principles outlining the characteristics of well-executed research, offering best practices for conducting design science research within the Information Systems discipline.

In Information System (IS), Design science research is widely recognized as a crucial component of the IS research landscape, representing its most significant aspect. The study of [114], Propose five criteria for the design science artifacts evaluation in addition to outlining the design science artifact development approach. According to these criteria, design science research: investigates a significant phenomenon in information systems; makes a significant contribution to the domain; generates testable and implementable artifacts; outperforms existing systems in terms of solutions; and gains generalizable experience from the system development process.

3.3. Design Science Research Process (DSRP)

Design Science Research (DSR) process is a systematic and iterative approach for conducting studies that aims to generate and evaluate innovative artifacts to address real-world problems in the information systems fields. In the context of rainfall prediction, DSR can be utilized to design and create predictive models, algorithms, and systems that enhance our ability to forecast rainfall patterns accurately. Design science introduced in the early 1990s to the IS community by [113] and [103], [114]. As [116] supported the integration of system development in to the research process, by proposing a multi methodological approach that would include theory building, systems development, experimentation, and observation. As well as [116] advocated for the incorporation of system development into the research process through a multi-methodological method, which encompasses systems development, theory building, experimentation, and observation.

As [116] introduced a DSRP framework specifically for collaborative systems research in their paper "Facilitating Group Creativity: Experience with a Group Decision Support System." Their model comprises three primary stages: Problem Formulation stage: systematically examining the problem domain, understanding the context for the issue, and figuring out the key issues that must be resolved. The second stage is; System Design and Development; researchers focus on creating the artifact or solution that solves the problem that was identified. To envision the artifact and develop prototypes, they use design principles, methods, and techniques. The third stage is; Demonstration and Evaluation: in this stage the artifact is evaluated for performance, functionality, usability, and effectiveness in solving the problem using appropriate techniques and metrics.

The research work done by [117], also include problem identification as a basic process step, as do [112] in their general guidelines for DS research. Since the process is iterative, it can be improved and refined continuously, resulting in artifacts that are useful, efficient, and in line with actual needs. In Table 3.1 below, we reviewed four research papers that used to develop design science research process models for information system research.

Table 3.1| Design Science process elements from IS and other Discipline ([118])

Objectives for DSR process model	(Nunamaker et al.1991) [116]	(Walls et al.1992) [113]	(Rossi et al.2003) [117]	(Hevner et al.2004) [112]
Problem identification and motivation.	Construct a conceptual framework	Meta-requirements kernel theories	Identify a need	Significant and relevant problems
Objective of a solution.				Implicit in 'relevance'
Design and Development	Develop a system architecture Analyze and design the system Build the system	Design method Meta Design	Build	Iterative search process Artifact.
Demonstration	Experiment and evaluate the system			
Evaluation		Testable Design process/product hypothesis	Evaluate	Evaluate
Communication				Communication

A different model was proposed forward in 2008 by Peffers, Tuuanen, Rothenberger, & Chatterjee [119], which is most frequently cited. The design science research (DSR) process encompasses four distinct entry points: problem-centered initiation, objective-centered solution initiation, design and development initiation, and client/context initiation. Additionally, it involves six steps: problem identification and motivation, defining solution objectives, design and development, demonstration, evaluation, and communication.

Peffers et al.[119], defined the process as;

- Problem Identification and Motivation: It involves precisely defining the research problem and providing justification for the significance of finding a solution.
- Objectives of a Solution: The goals of a solution are inferred from the definition of the problem itself.

- Design and Development: Create the artifactual solution. Such artifacts are potentially, with each defined broadly, constructs, models, methods, or instantiations [112].
- Demonstration: The effectiveness of the artifact in solving the problem is demonstrated through activities such as experimentation, simulation, case studies, proofs, or other suitable methods.
- Evaluation: Observation and measurement of how well the artifact supports the problem solution involves comparing the proposed objectives of the solution to the actual results.
- Communication: This communication should target researchers and other relevant audiences, including practicing professionals, when appropriate.

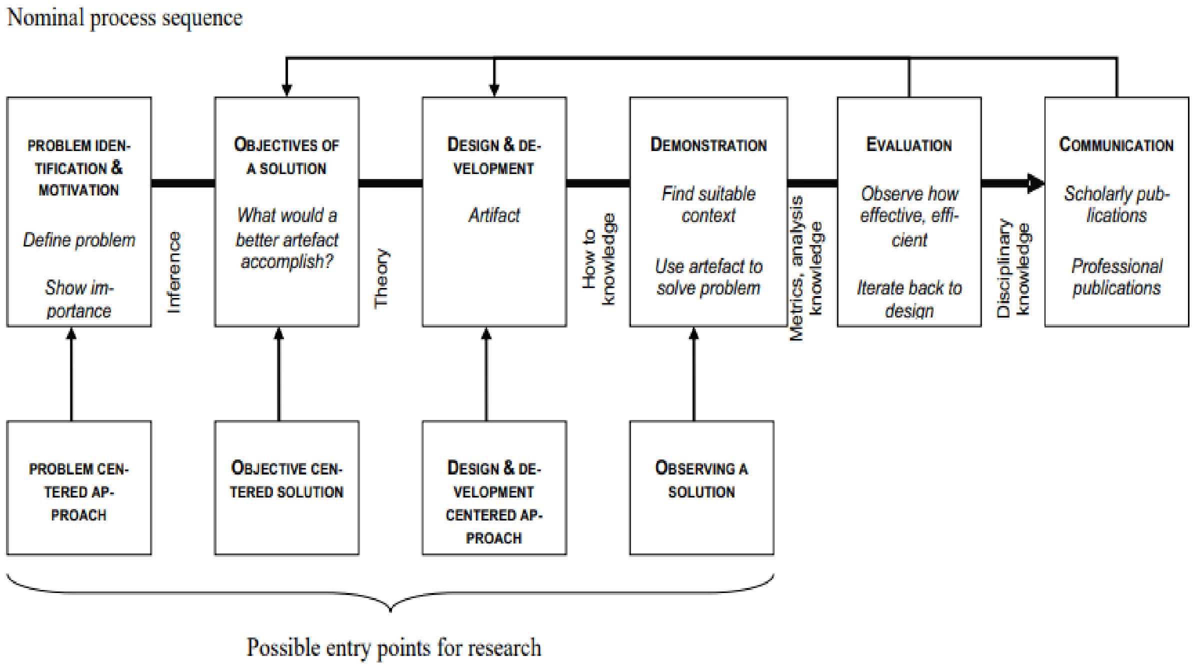


Figure 3.1| Design science research process (DSRP) model (Peffers, et al.[118])

3.4. Proposed Design Science Research Process (DSRP)

As depicted in Figure 3.1 and summarized in Table 3.1, there are several researchers have proposed design science research processes for information system research. Nunamaker et al. (1991), Walls et al. (1992), and Rossi et al. (2003) put forward three such processes. Hevner et al. (2004) expanded on these processes by including objectives of the solution and emphasizing the communication process. Peffers et al. (2008) further enhanced the design science research process by incorporating the demonstration phase. This phase involves utilizing simulations, live prototypes, or mock-ups to effectively showcase the capabilities and functionalities of the developed artifact in addressing the identified problem.

In this study, we adopted the design science research process for rainfall prediction from Nunamaker et al., Walls et al., Rossi et al., Hevner et al., and Peffers et al. These scholars have made significant contributions to the field of design science and information systems, and their works have helped to shape the understanding and implementation of DSRP. The adopted DSR process contains three key processes such as Problem identification and Motivation, Designing and Developing Solution, and Evaluation. Because, these three processes provides a complete coverage of the key activities elaborate in performing design science research. According to the articles listed in Table 3.1 and Figure 3.1, the essential steps from problem identification and understanding the relevance of the research, to designing and developing the artifact, and finally, carefully evaluating the artifact to ensure its effectiveness and contribution are listed. The figure 3.2 shows the adopted design science research process for this study.

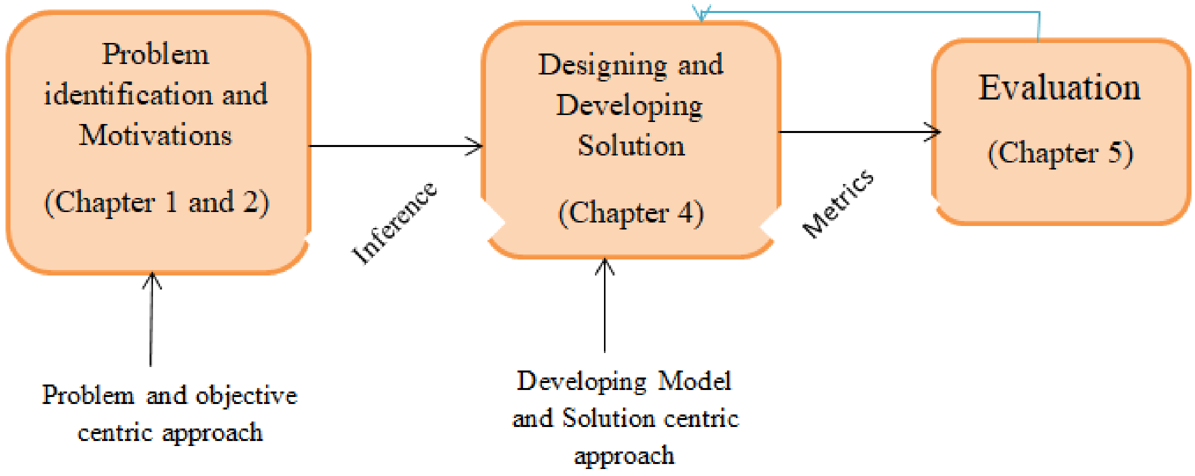


Figure 3.2| Adopted Design Science Research model

Detailed description of the adopted design process model is described below.

Problem Identification and Motivation: This process contains identifying and understanding the problem or opportunity that needs to be addressed and motivates the values of a solution. This step includes examining the existing literature, understanding the context and needs of stakeholders, and identifying the gap or challenge this study aims to experiment

[98], [106]. The research gap in this study is that no attempts were made to improve the performance of rainfall forecasting models using deep learning techniques on the one hand by combining station and satellite data to increase the accuracy of rainfall prediction in Ethiopia. Another problem is; Ethiopia is using the NWP model to predict rainfall and other weather variables. This NWP model requires large computational resources and the model is sensitive to the accuracy of the first conditions. Also, it isn't easy to use this tool for each specific area, because of its high requirements in resources and costs.

In the problem justification phase of the design science research methodology, it is crucial to provide a rationale for the objective of the solution [105]. In alignment with this, the objective of the current study is to design and develop a predictive model for rainfall prediction using deep learning approaches. The ultimate aim is to enhance agricultural productivity, which is instrumental in ensuring food and water supply of high quality for the country's citizens. This developed model will not only facilitate understanding the relationship between rainfall and other weather parameters but also contribute to maximizing the accuracy of predictions and overall agricultural productivity. Detailed descriptions of this phase are presented in chapters one and two of this research.

Designing and developing Solution: This process focuses on designing and developing a solution or artifact to address the identified problem or opportunity. This process involves applying design principles, theories, and methodologies to create a new or improved information system artifact [119]. During the design and development phase of the design science research methodology, it is essential to identify the necessary functionalities and architecture of the artifact. This process involves determining the desired functionality and structure of the artifact, which can encompass a wide range of constructs, models, methods, or instantiations [112]. Subsequently, the actual creation of the artifact takes place, bringing the envisioned design to life. This activity focuses on translating the identified functionalities and architectural specifications into the tangible artifact itself.

In this study, the artifact is a Rainfall prediction model which is designed and developed based on the above mentioned objectives. The model developed based on deep learning techniques such as MLP, CNN, LSTM, and BiLSTM. The deep learning approaches are chose in this study because of; deep learning handles complex patterns which mean that it analyze large amounts of data with complex temporal and spatial patterns in rainfall prediction. The second reason is deep learning models are capable of processing huge datasets with a variety of inputs. The third reason is non-linear relations between the input variables and the rainfall patterns are frequently seen in rainfall prediction. Compared to conventional machine learning algorithms, which are often limited by their linear assumptions, deep learning models are capable to represent complicated non-linear functions, and are better able to capture these non-linear interactions. Readers can refer chapter four of this thesis to understand the details of the solution design phase.

Evaluation: The evaluation phase plays a critical role in measuring the effectiveness of the model and its association with the problem-solving purposes. To authenticate the novelty and establish the research significance of the designs produced, prior knowledge sources such as scientific theories, methodologies, practical experience, domain expertise, and existing design products and processes are employed. This approach ensures that the designs created go beyond routine practices and contribute to the existing body of knowledge. Different measures have been employed to evaluate the models developed in this research. This activity involves comparing the intended objectives of the solution to the actual observed results obtained from the utilization of the artifact during the demonstration phase. By examining the extent to which the artifact aligns with the desired outcomes, the evaluation process helps determine the efficacy of the model in addressing the identified problem.

Deep learning model accuracy and performance metrics are used to measure how well a model is performing on a given task, such as classification or regression [120]. This study is predicting the amount of rain based on different weather stations and satellite data, which means that the output is a form of estimation. We can’t measure estimation or regression using precision, recall, F1Score, and with confusion metrics. While they are valuable for assessing the performance of models in classification problems where the goal is to assign instances to specific categories or classes, they may not be directly applicable to rainfall prediction. In classification, the model predicts a class label while our model predicts a numeric value in regression [121]. Therefore, we used regression metrics such as Root Mean Square Error (RMSE), Mean Square Error (MSE), and Mean Absolute Error (MAE). These metrics focus on measuring the accuracy of continuous predictions by calculating the differences between the predicted rainfall and observed values [122]. These metrics enable researchers to evaluate the accuracy and dependability of the model's predictions and make well-informed conclusions regarding the model's performance by comparing the model's predictions with the actual observed rainfall amounts. The designed models were evaluated based on the test datasets.

Dataset used to evaluate the model: To develop machine learning and deep learning models, it is common practice to divide the dataset into two parts: the training set and the testing set [123]. The training set is utilized to train the model and guess the unknown parameters, while the testing set is used to evaluate the model's accuracy. This separation is necessary to prevent overfitting, which occurs when the model becomes too specific to the training data and performs poorly on new, unseen data. In this study, the final pre-processed dataset, consisting of 11,324 instances and 8 columns, was divided into training and test datasets. Following established research [124], an 80% was allocated for training the models, and the remaining 20% was designated for testing. This consistent split was applied across all models to ensure fairness and comparability in evaluating their performance. Detailed explanations of each metric used for model evaluation are provided in Chapter Five of this research.

3.5. Chapter Summary

The overall chapter discussed about the research methodology, the research approaches and the design science research processes. Furthermore, different scholars design science research processes are review and discussed. Then the adopted process such as identification and understanding of the problems, design and developing solution, and evaluations of the developed model are discussed.

CHAPTER FOUR

Solution Design

4.1. Introduction

The solution in this research is a model which is a common artefact in design science research. Hence, we design a rainfall prediction model in this chapter using deep learning architecture, which can estimate the amount and intensity of rainfall for a specific location or region. In this chapter, the model design process and the solution architecture is discussed. Moreover, justification of the data acquisition and understanding, data engineering, and model building, also the optimal parameters for developing rainfall prediction are selected for all proposed architectures.

4.2. Model Architecture

Figure 4.1 shows the proposed solution architecture to design rainfall prediction model. We followed a common data analytics process to build the model. In the architectures, the first phase is data acquisition and understanding. The key purpose of this phase is extracting data from different station data sources and satellite sources. The second phase is data engineering which involves handling missing instances, integrating the two data sources and handling data redundancy and inconsistency issues. There is also an issue of feature extraction and dimensionality reduction in this phase. Different techniques of data engineering are applied.

The third phase is model building which is based on deep learning AI techniques. The use of deep learning approaches offers the ability of identifying non-relationship between the predicted and independent parameters and also detects new patterns of rainfall in the given training datasets. The deep learning architecture encompasses the different stages of the deep learning cycle and involves the key steps in transforming raw data into training datasets that enable system decision-making. The final phase of this architecture is model evaluation, which aims to assess the performance of the trained model on test data.

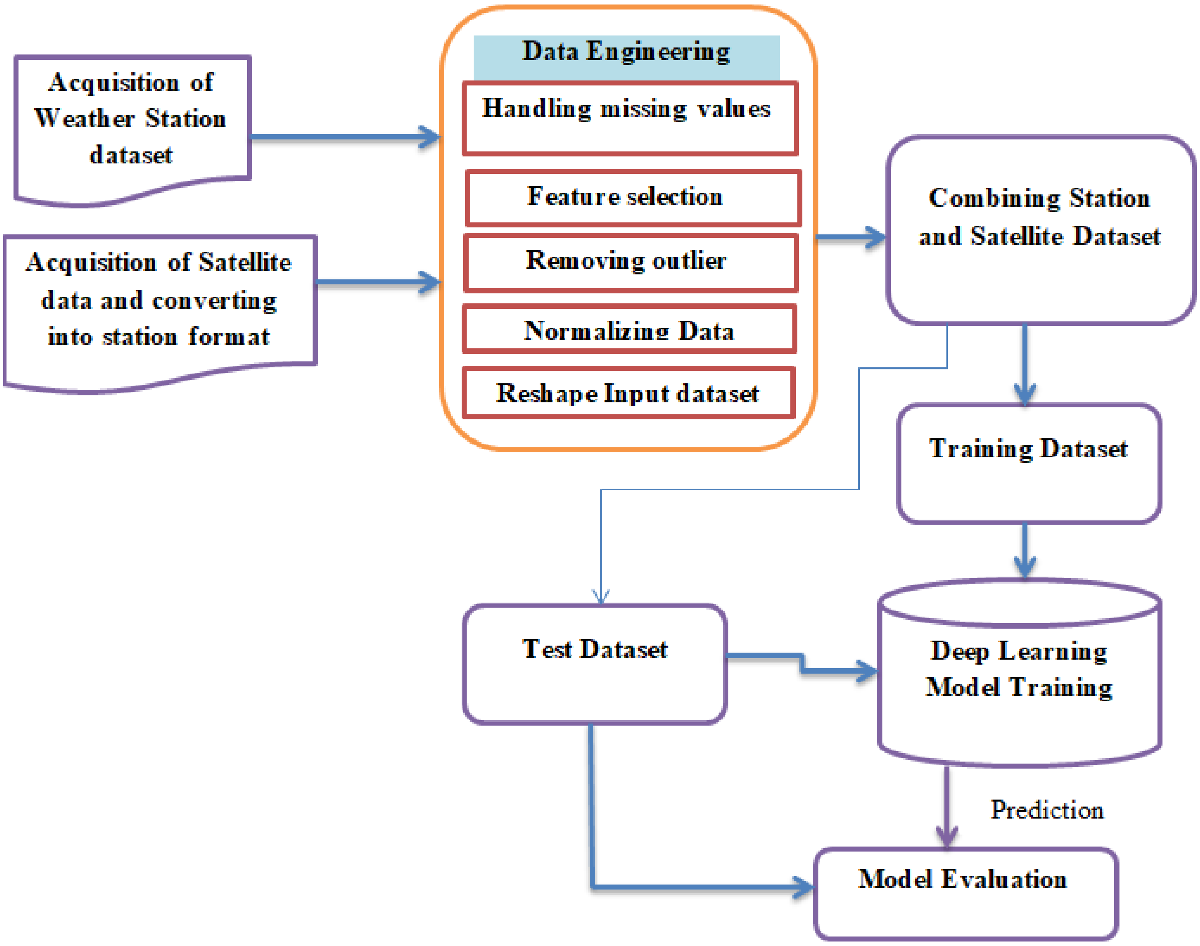


Figure 4.1| Architecture of rainfall prediction model

4.2.1. Data Acquisition and Understanding

Data is essential for developing deep learning models. In this section we discussed about the study area and data collection for both weather station and satellite data that used in this thesis. The study area includes Addis Ababa to Adama, which is a large geographical area in the east showa of Ethiopia as shown in (Figure 4.2). The city is located at 8.981°N 38.798°E near of the Entoto Mountains and vertical 7,726 feet (2,354 meters) above sea level. Adama is a city in the central Oromia Region, Ethiopia. Adama, positioned within the East Showa Zone and forming a Special Zone of Oromia, can be found at 8.557°N 39.284°E. The city located at an elevation of 1648 meters and is situated 99 km southeast of Addis Ababa, the capital of Oromia.

The study area has significant importance in terms of its geography, ecology, and economic activities. The study area has a variety of uses, including; the area primarily used for agriculture, with crops such as wheat, teff, maize, barley, and beans grown in the fertile plains of the Awash River basin. As well as the area is home to large industrial parks, including the Eastern Industrial Zone, the Bole Lemi Industrial Park and etc. The study area is a major transportation hub in Ethiopia, with the Addis Ababa-Adama highway and the Addis Ababa-Djibouti railway passing through the region. The region is also home to the Bole International Airport, which serves as the main airport for the country.

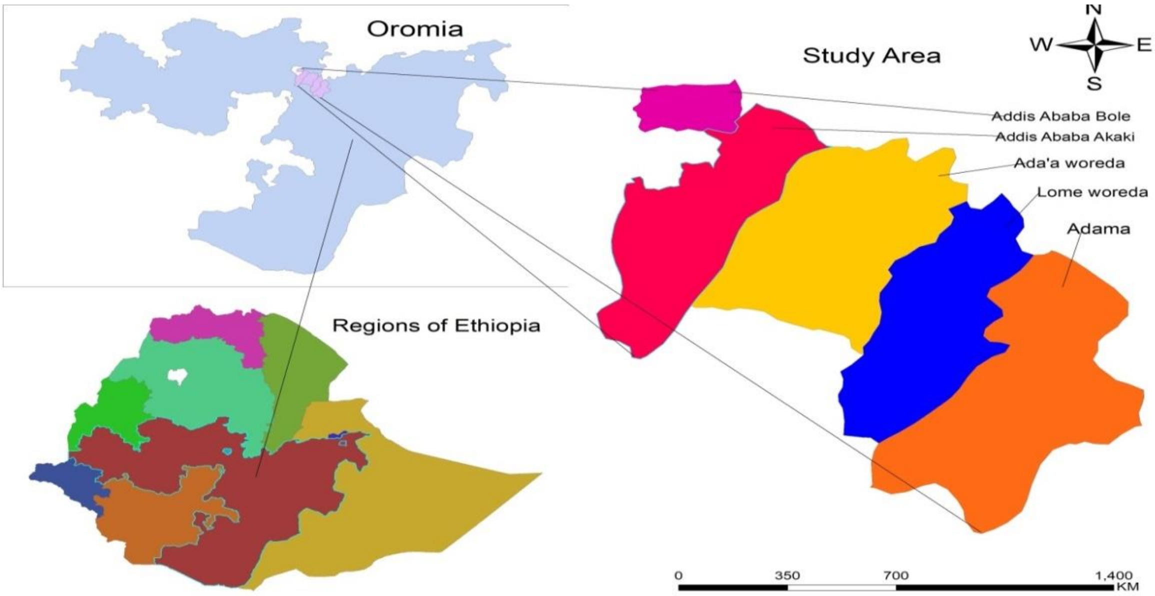


Figure 4.2| The maps of the study area.

In order to build model to be used in the research area, two types of data is used. The first data is station data. For this research, the daily accumulated weather station data acquired from the national meteorological agency of Ethiopia, that covers a period of 30 years from (1990 - 2020). The data comprises twelve variables, which consisted of the geographic variables: “longitude”; “latitude”; “elevation”; and “station name”; the time variables: “year”; “month”; “day”, and the meteorological variables: “maximum temperature”; “minimum temperature”; “relative humidity”; “wind speed”; “solar radiation”; and “rainfall”.

The second data source is Satellite data. Satellite data refers to the information collected by satellites orbiting the Earth. These satellites capture various types of data, such as images, measurements, and other observations, which are useful for a wide range of applications, including rainfall prediction. Satellite data plays a crucial role in rainfall prediction by providing valuable information about weather patterns and atmospheric conditions. The original satellite data downloaded for this study is in gridded format for the selected locations. Daily based minimum and maximum Temperature, rainfall, relative humidity, air pressure, and cloud cover of the past 30 years (from 1990 to 2020) were collected with the same period of station data. The Precipitation data were downloaded from TAMSAT v3.1 for daily gridded based on the study location by netCDF format. The TAMSAT v3.1 is a rainfall estimation system that utilizes satellite and ground-based observations. Developed by the Tropical Applications of Meteorology using a Satellite and ground-based observations (TAMSAT) project, it offers precise precipitation estimates for Africa. The system provides high-resolution rainfall data with a spatial resolution of 0.25° and a temporal resolution of 3 hours. Figure 4.3 shows the sample image of a precipitation that downloaded from satellite for the study area.

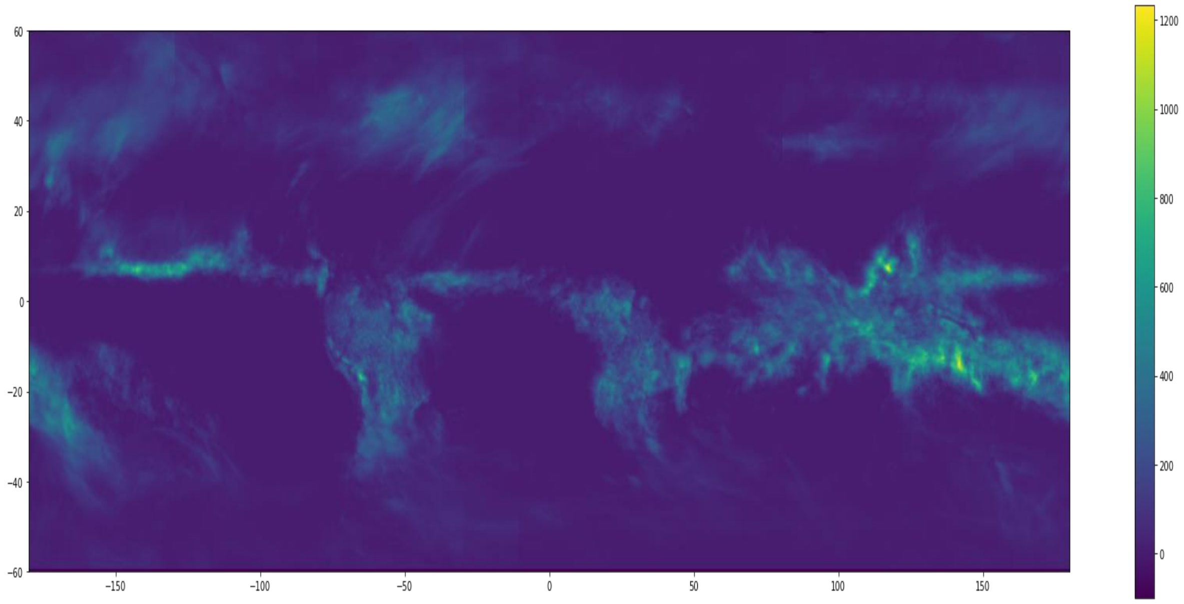
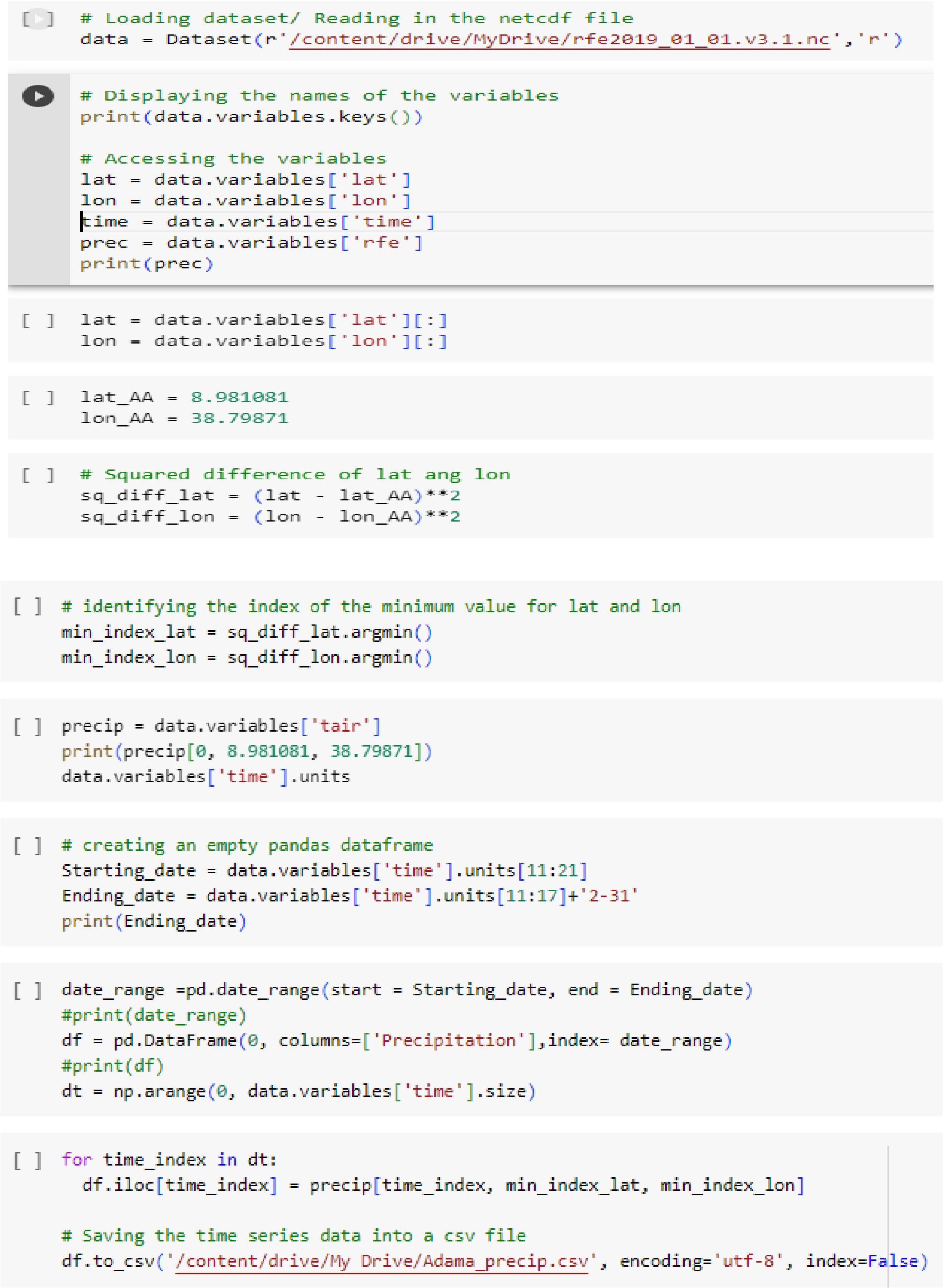


Figure 4.3| Sample satellite image for rainfall

The remaining Satellite datasets were downloaded from JRA-55 that were collected at the interval of three hours, and then time series aggregation was performed using Climate Data Tools (CDT) v7.0 into daily data. CDT is a software tool designed to assist users in accessing and processing weather and climate data from a variety of sources. The collected satellite data is in a netCDF format. Then we used python code to extract the values of satellite data into station format. The satellite data contains; “Date”; “longitude”; “latitude”; “maximum temperature”; “minimum temperature”; “relative humidity”; “wind speed”; “Air pressure”; “cloud cover” and “rainfall” Figure 4.4 show the python code that used to extract the netCDF format into station formats.



```
# Loading dataset/ Reading in the netcdf file
data = Dataset(r'/content/drive/MyDrive/rfe2019_01_01.v3.1.nc','r')
```

```
# Displaying the names of the variables
print(data.variables.keys())

# Accessing the variables
lat = data.variables['lat']
lon = data.variables['lon']
time = data.variables['time']
prec = data.variables['rfe']
print(prec)
```

```
lat = data.variables['lat'][:]
lon = data.variables['lon'][:]
```

```
lat_AA = 8.981081
lon_AA = 38.79871
```

```
# Squared difference of lat ang lon
sq_diff_lat = (lat - lat_AA)**2
sq_diff_lon = (lon - lon_AA)**2
```

```
# identifying the index of the minimum value for lat and lon
min_index_lat = sq_diff_lat.argmin()
min_index_lon = sq_diff_lon.argmin()
```

```
precip = data.variables['tair']
print(precip[0, 8.981081, 38.79871])
data.variables['time'].units
```

```
# creating an empty pandas dataframe
Starting_date = data.variables['time'].units[11:21]
Ending_date = data.variables['time'].units[11:17]+'2-31'
print(Ending_date)
```

```
date_range =pd.date_range(start = Starting_date, end = Ending_date)
#print(date_range)
df = pd.DataFrame(0, columns=['Precipitation'],index= date_range)
#print(df)
dt = np.arange(0, data.variables['time'].size)
```

```
for time_index in dt:
  df.iloc[time_index] = precip[time_index, min_index_lat, min_index_lon]

# Saving the time series data into a csv file
df.to_csv('/content/drive/My Drive/Adama_precip.csv', encoding='utf-8', index=False)
```

Figure 4.4| Python code that used to extract satellite data

Table 4.1 describes the experimental data in detail. The meteorology station records consist of daily values of weather variables collected directly from the station's devices. These data are recorded in a tabular format using Microsoft Excel files.

Table 4.1| Variables used in our study, their description and corresponding units.

Feature Name	Description	Unit/measurement
NAME	Name of the station	
Elevation	Distance above sea level	
Lat	Latitude of the area	
Lon	Longitude of the area	
DATE	Year, month and days of observation	
TMPMAX	Daily Maximum Temperature considered	°C(Degree Celsius)
TMPMIN	Daily Minimum Temperature considered	°C(Degree Celsius)
WINDLY	Daily Wind run in meters per second	Meters per second (m/s)
RELHUM	Daily Relative humidity considered	Percentage (%)
Cloud Cover	Sky covered by clouds	Percentage (%)
Air Pressure	The Pressure of the air	Millibars (mb)
SUNHRS	Daily sunshine hours	Hours (hr)
PRECIP	Daily Precipitation or rainfall amount considered	Millimeters (mm)

4.2.2. Data Engineering

Data engineering is one of the most important phases of model building based on machine learning methods and deep learning techniques [125] and ensures that the available data is properly handled and organized for further analysis and utilization in succeeding phases. Here, noise or missing data typically have a negative impact on the raw data. The data engineering phase is essential for improving the prediction process, by ensuring the data are regularized and filtered beforehand [126], [127]. But the collected data itself cannot directly be used for modeling. This is because;

- Data might contain errors, outliers, missing values, and other quality problems which need to be resolved beforehand.

- There might be a necessity of deriving attributes and even tables from the existing data.
- A need of integrating data from two or more sources would become mandatory, and
- The original data may not be in a format and structure, which are acceptable to the modeling tools and techniques.

The Data engineering phase includes properly handling missing values, detecting and resolving outlier, feature extraction, and normalizing the data techniques are applied in this study.

Handling Missing Values- Missing values in data can be caused by various reasons, and their occurrence can introduce several problems and reduce the statistical power, actually leading to inaccurate hypothesis evaluation. Depending on the context, the extent of missing values, and the specific requirements of the analysis, the missing values of weather data can be handled in various ways. Based on the mechanisms described above, researchers have employed various methods to handle missing datasets, including the approach of dropping the missing values, imputing missing values either of Mean, Median, mode, Forward-fill, or Backward-fill imputation, and using machine learning algorithms to predict missing values. For numerical variables that can take any values within their maximum and minimum missing values ranging from 2% to 30% were imputed using the mean values. This approach is based on the assumption that the mean value is a reasonable approximation for randomly nominated observations from a normal distribution.

The weather station data obtained from the NMA of Ethiopia that covers a period of 30 years has contains some missing values. In this study, three different cities' data were pre-processed and used to train the model to predict rainfall. For Addis Ababa, Adama, and Bishoftu the station data have some missing values as shown below in Table 3.2, for each dependent as well as independent weather variable that is used in this study.

Table 4.2| Details of the missing values for station datasets

Feature	**Missing values in %**		
	Addis Ababa	**Adama**	**Bishoftu**
TMPMAX	3.75 %	3.86 %	38.24 %
TMPMIN	3.39 %	4.53 %	34.88 %
WINSPD	12.47 %	46.18%	37.06 %

RELHUM	25.28 %	38.22 %	33.09 %
SUNHRS	33.32 %	26.24 %	43.30 %
PRECIP	2.27 %	2.94 %	3.52 %

Satellite data pre-processing includes a series of steps to make the raw data and extract information to use as input for deep learning algorithms to develop models for rainfall predictions. This step was done by using the tools provided by the CDT which is the quality control module and the result shows that the data are in a good status. Then by using CDT we extracted values of each variable for the study area by using the latitude and longitude of the area. Table 3.4, shows the number of missing values for each variable after the value was extracted;

Table 4.3| Details of the missing values for satellite data

Features	Missing values in %		
	Addis Ababa	Adama	Bishoftu
TMPMAX	9.35 %	9.25%	10.26 %
TMPMIN	9.22%	8.50%	9.84%
WINSPD	15.65%	15.25%	16.56 %
RELHUM	20.46%	14.50%	22.34 %
Cloud	4.23%	4.23%	4.50 %
PRECIP	0.19%	0.26%	0.26 %

During the cleaning process of this study, it was observed that the collected data contains missing values symbolized by characters such as 'nan' and “”. These missing values can introduce errors in the prediction process. Hence, it is essential to address and handle these missing values properly. In order to handle missing values for both dataset, we used mean imputation method. Here, we done two experiments, in the first experiment the missing values were handled by substituting them with the mean calculated by all datasets and in the second experiment, first the dataset was divided into three groups (from Jun-Sep, Oct-Jan, and Feb-May) based on the Ethiopian season that was discussed in chapter one then three different mean was calculated based on the season and replaced based on the season. To address the missing values based on the season, a mean average mechanism is employed. This includes calculating the mean average for each group of the selected attribute within the dataset. The mean average is determined by summing all instances of each group and dividing the sum by the number of records in that group. By using this method, the missing values are populated with the mean average values specific to their respective seasons. The second experiment performed well in this study and we used seasonal based mean imputation for missed values. For every process of data engineering we used python code.

Outlier detection: - is the process of identifying data points that diverge significantly from the rest of the dataset. In the context of deep learning algorithms, outliers can have a negative influence on the model's performance, as they can skew the model's training and result in incorrect predictions [128]. Outliers can arise in machine learning and deep learning algorithms for various reasons, such as data collection errors, measurement errors, or simply due to the natural variability of the data. Detecting and resolving outliers can help to improve the accuracy and consistency of the deep learning algorithm. There are numerous techniques that can be used for outlier detection and resolution in deep learning algorithms. One common approach is to use statistical methods such as z-scores or interquartile range (IQR) to identify outliers [129]. Another approach is to use machine learning algorithms, such as clustering or anomaly detection, to identify unusual data points.

In this paper, the outliers were detected using a Z-score, which is used to identifying data points that deviate significantly from the mean. If a data point has a high z-score (i.e., is far from the mean), it is considered an outlier [129]. A commonly used threshold is 3, which indicates data points that are more than 3 standard deviations from the mean.

$$\mathrm{Z-score} = \frac{(X - \mu)}{\sigma}$$

Where X denotes the data point, μ - mean of the dataset, and σ - standard deviation of the dataset.

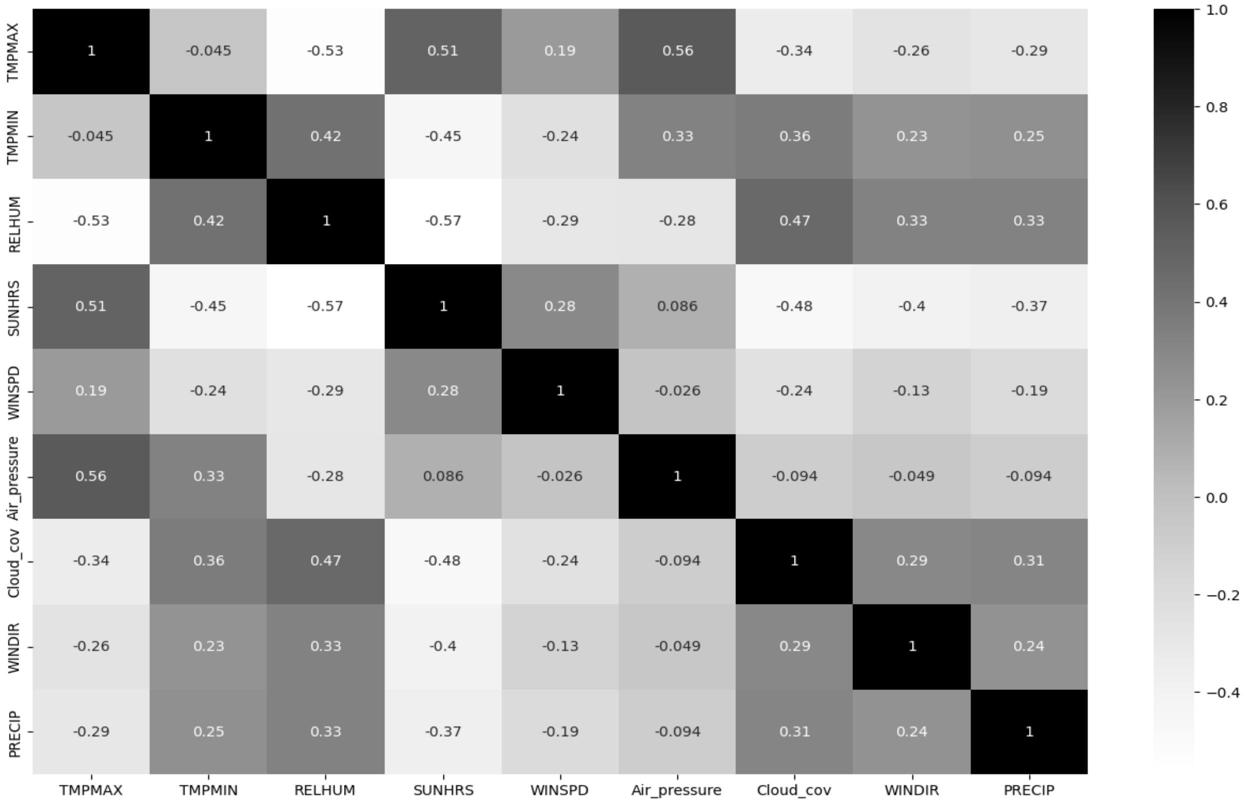
Once outliers have been identified, there are several ways to resolve them. One approach is to remove the outliers from the dataset entirely. Another approach is to replace the outlier values with a more reasonable estimate, such as the mean or median value of the surrounding data points. After the outliers were detected, we used reasonable estimation method which is using mean technique.

Data combination: The station dataset contains 11324 instances and 12 columns including geographical, time, and meteorological variables, and the satellite dataset also contains 11324 instances and 11 columns that include geographical, time, and meteorological variables after being converted into station format. After converting the satellite data into station format and performing data pre-processing for both datasets, the station and satellite data were combined by taking the mean for the same variable and incorporating the remaining variable as it is into a single dataset. After both datasets were combined the final dataset contains 11324 instances and 8 meteorological variables. This process allowed for integrating information from both sources, ensuring that the resulting dataset contains the averaged values for shared variables while retaining the original values for the non-shared variables.

Selecting features: - is the process of identifying the most relevant features or input variables from a dataset that are useful for building a model. In this study, feature selection involves identifying the weather parameters that have the strongest correlation with rainfall prediction and including them as input features to the model. The reasons for performing feature selection in predicting rainfall includes; improving model accuracy by eliminating irrelevant or redundant variables from the dataset, reducing computational complexity, and increasing the interpretability of the model. Correlation Analysis between data was done to analyze the correlation between the independent and dependent variables. Additionally, it supports in identifying any potential issues of multicollinearity, which refers to high correlations among the independent features. By examining the correlation coefficients, valuable insights can be gained regarding the interdependencies and associations among the variables.

To analyze the correlation between both continuous and categorical variables in the study, the 'Spearman' correlation method was employed. Additionally, a correlation matrix, presented in Table 4.4, provides a detailed overview of the correlation coefficients between each pair of variables.

Table 4.4| Correlation coefficient between independent variables and dependent variable



	TMPMAX	TMPMIN	RELHUM	SUNHRS	WINSPD	Air_pressure	Cloud_cov	WINDIR	PRECIP
TMPMAX	1	-0.045	-0.53	0.51	0.19	0.56	-0.34	-0.26	-0.29
TMPMIN	-0.045	1	0.42	-0.45	-0.24	0.33	0.36	0.23	0.25
RELHUM	-0.53	0.42	1	-0.57	-0.29	-0.28	0.47	0.33	0.33
SUNHRS	0.51	-0.45	-0.57	1	0.28	0.086	-0.48	-0.4	-0.37
WINSPD	0.19	-0.24	-0.29	0.28	1	-0.026	-0.24	-0.13	-0.19
Air_pressure	0.56	0.33	-0.28	0.086	-0.026	1	-0.094	-0.049	-0.094
Cloud_cov	-0.34	0.36	0.47	-0.48	-0.24	-0.094	1	0.29	0.31
WINDIR	-0.26	0.23	0.33	-0.4	-0.13	-0.049	0.29	1	0.24
PRECIP	-0.29	0.25	0.33	-0.37	-0.19	-0.094	0.31	0.24	1

As shown in Table 4.4 thus the results of correlation also helped in feature selection. The variable Relative Humidity has the most connection with the target variable "Precipitation or rainfall," positively and solar radiation (SUNHRS) has high negative correlation with target variable, as it can be seen from the heatmap of correlations above. There is no multicollinearity (high inter-correlation) among two or more independent variables. The correlation represents the degree to which one variable is influenced by another. The interpretation of the correlation values with their range is displayed in the below table.

Table 4.5 Spearman coefficient ranges and interpretations [130]

Spearman coefficient	**Interpretation**
0.90 to 1.00 (- 0.90 to -1.00)	Very high correlation
0.70 to 0.90 (- 0.70 to - 0.90)	High correlation
0.50 to 0.70 (- 0.50 to - 0.70)	Moderate correlation
0.30 to 0.50 (- 0.30 to - 0.50)	Low correlation
0.00 to 0.30 (0.00 to - 0.30)	Negligible correlation

Based on the result presented in Table 4.5, it can be observed that the dependent variable (Rainfall) exhibits weak correlations with the seven independent weather variables. In order to identify the most influential features for predicting rainfall, feature selection techniques, also referred to as variable selection, attribute selection, or variable subset selection, and are employed. In this study, the correlation method was used to determine the optimal predictive features for rainfall prediction. Feature selection plays a critical role in improving the accuracy of a model by improving its training efficiency and reducing complexity. In this study, Random Forest was employed to identify the most influential features for feature selection, based on insights from the existing literature. Random Forest is an ensemble learning method widely used to determine the optimal set of features that contribute significantly to predicting the target variable.

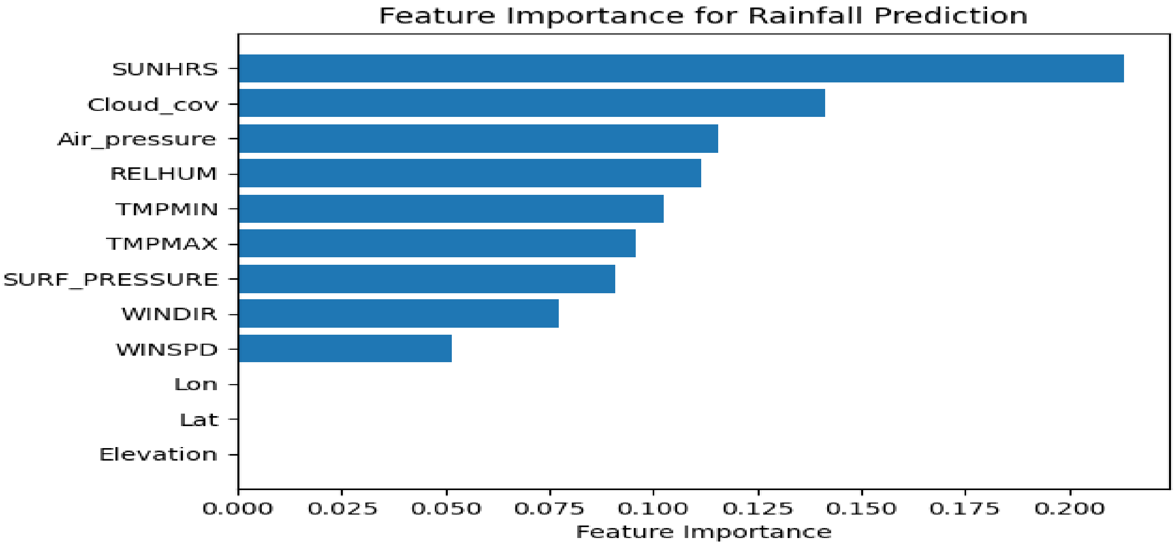


Figure 4.5| Feature importance's for rainfall prediction

Based on the graph depicted above, it is evident that the most important variable in predicting rainfall is SUNHRS (solar radiation). This is followed by Cloud cover, Air pressure, RELHUM (relative humidity), and other factors in descending order of importance. These results highlight the importance of these specific variables in accurately forecasting the target variable in the context of rainfall prediction.

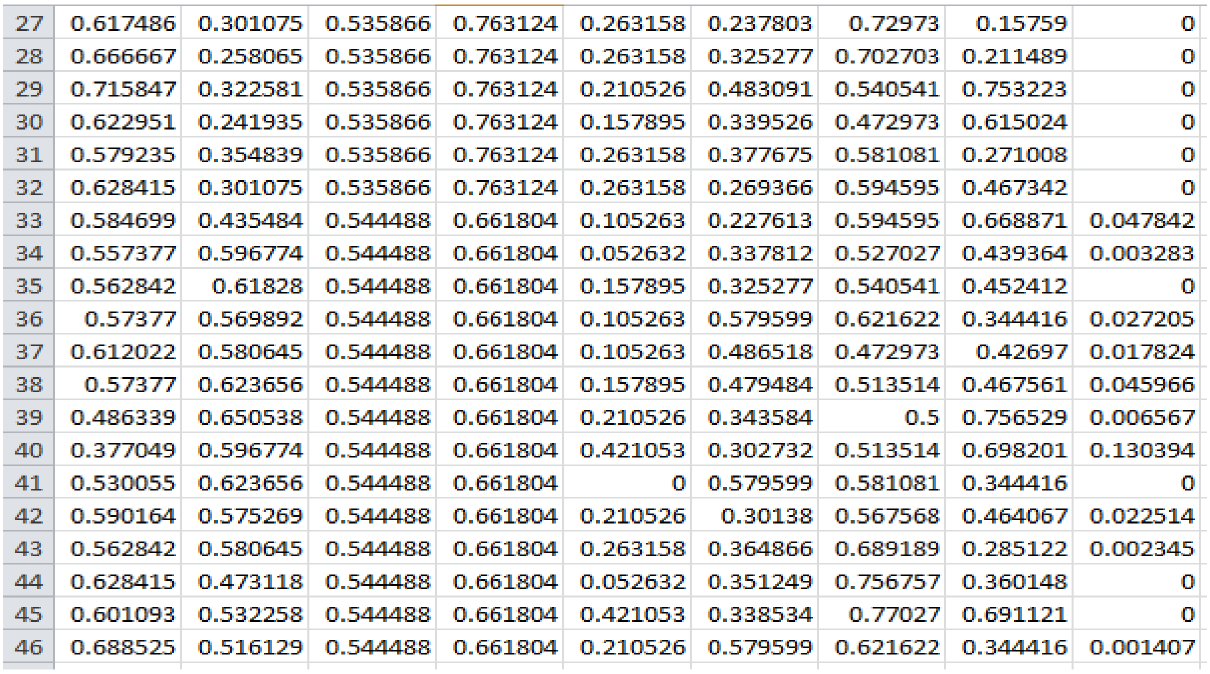
Data Normalization: is a process that aims to normalize the dimensions of the data to ensure they are approximately on the same scale [131]. In the context of training neural networks, input data normalization plays important role in achieving best results. According to [132], the choice of activation function in a BPNN model determines the specific standardization requirements for input data. For example, when using a standard sigmoid function, each input variable should be normalized between 0 and 1. On the other hand, if a hyperbolic tangent function is employed, the input variable should be normalized between -1 and 1. Failing to follow to these normalization ranges may result in fixed values such as -1, 1, or 0 for many hidden neurons in the network. Hence, proper data normalization is essential for ensuring the effective training and performance of the deep learning model.

To assess the normality of continuous variables, the skewness and kurtosis continuous variables were calculated. If these values lay outside the range of [0, 1], it indicated that the variable was not normally distributed. Furthermore, the normality of variables could also be visually assessed using histogram plots. Both the weather station and satellite datasets were normalized using the sklearn MinMaxScaler () function, which rescales the attribute values from their original domain to a new range, typically between 0 and 1 [133]. This technique, known as min-max scaling, was applied to specific variables in the dataset to ensure data normalization. By using the min-max scaler, the weather parameters were transformed to obtain new scaled values, denoted as "v." This normalization process helps to bring the variables to a standardized range and facilitates further analysis and modeling.

$$v = \frac{X - \mathrm{Min}(X)}{\mathrm{Max}(X) - \mathrm{Min}(X)}$$

Where X - scaled data value, Max(x) - maximum value, and Min(x) - minimum value from input.

Table 4.6| Sample dataset after data engineering performed.



27	0.617486	0.301075	0.535866	0.763124	0.263158	0.237803	0.72973	0.15759	0
28	0.666667	0.258065	0.535866	0.763124	0.263158	0.325277	0.702703	0.211489	0
29	0.715847	0.322581	0.535866	0.763124	0.210526	0.483091	0.540541	0.753223	0
30	0.622951	0.241935	0.535866	0.763124	0.157895	0.339526	0.472973	0.615024	0
31	0.579235	0.354839	0.535866	0.763124	0.263158	0.377675	0.581081	0.271008	0
32	0.628415	0.301075	0.535866	0.763124	0.263158	0.269366	0.594595	0.467342	0
33	0.584699	0.435484	0.544488	0.661804	0.105263	0.227613	0.594595	0.668871	0.047842
34	0.557377	0.596774	0.544488	0.661804	0.052632	0.337812	0.527027	0.439364	0.003283
35	0.562842	0.61828	0.544488	0.661804	0.157895	0.325277	0.540541	0.452412	0
36	0.57377	0.569892	0.544488	0.661804	0.105263	0.579599	0.621622	0.344416	0.027205
37	0.612022	0.580645	0.544488	0.661804	0.105263	0.486518	0.472973	0.42697	0.017824
38	0.57377	0.623656	0.544488	0.661804	0.157895	0.479484	0.513514	0.467561	0.045966
39	0.486339	0.650538	0.544488	0.661804	0.210526	0.343584	0.5	0.756529	0.006567
40	0.377049	0.596774	0.544488	0.661804	0.421053	0.302732	0.513514	0.698201	0.130394
41	0.530055	0.623656	0.544488	0.661804	0	0.579599	0.581081	0.344416	0
42	0.590164	0.575269	0.544488	0.661804	0.210526	0.30138	0.567568	0.464067	0.022514
43	0.562842	0.580645	0.544488	0.661804	0.263158	0.364866	0.689189	0.285122	0.002345
44	0.628415	0.473118	0.544488	0.661804	0.052632	0.351249	0.756757	0.360148	0
45	0.601093	0.532258	0.544488	0.661804	0.421053	0.338534	0.77027	0.691121	0
46	0.688525	0.516129	0.544488	0.661804	0.210526	0.579599	0.621622	0.344416	0.001407

Reshaping data: After the process of data normalization is conducted, and then the data can be reshaped in 3 dimensions. Preparing data for deep learning algorithms involves formatting the input data in a manner suitable for the neural network architecture. This includes changing the dimensions of the original data [134]. The process of getting sequence data ready for LSTM and BiLSTM models can be challenging. To address this challenge, we convert the input data sequence from a 2D matrix to the necessary 3D format for the LSTM and BiLSTM input layer. This transformation is carried out after removing empty records, resolving missing values, and employing the min-max scaler.

4.2.3. Model Building

Various models were developed by different researchers using big data techniques [135] and conventional machine learning approaches [107]. However, in this study, we developed a deep learning model to overcome the limitations of statistical and machine learning algorithms. The main reason why we used deep learning approaches is; rainfall patterns are nonlinear relationships with other meteorological variables and the models can capture these complex connections more effectively than linear models commonly used in traditional statistical and machine learning approaches. Also, deep learning algorithms have gained acceptance in recent years due to their capability to automatically learn and handle complex patterns in large datasets, as well as detect the temporal and spatial patterns in rainfall prediction.

The research focused on building models using four well-established deep learning algorithm architectures that have proven effective in rainfall prediction. These architectures include the Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short Term Memory (LSTM), and Bidirectional Long Short Term Memory (BiLSTM). In this research we selected these deep learning architectures because of; the main reason why we used only these four architectures for this study is; one each model has its own characteristic that makes them suitable for different types of data, such as for station, satellite and for combined both data. Deep learning algorithms typically need to careful design the architecture, deciding on types and number of layers and activation functions, because of its multiple layers and complex architecture. This needs a deep understanding of the problem area, as well as expertise in neural networks.

Another reason is each approaches has recommended by different researchers as a best approach for rainfall prediction using satellite and station data separately. MLP is proposed by [77], and [81] as best rainfall prediction model than other machine learning algorithms. Also, [105] proposed LSTM model on a monthly basis and on an yearly basis for predict the amount of rainfall. Others such as [67] suggested LSTM for temperature prediction in weather forecast, [103] proposed for daily rainfall prediction. Others such as [83] and [99] proposed CNN architecture for satellite image based and for station based format data to predict rainfall. Also, BiLSTM used for rainfall prediction that has forward and backward neurons [104]. On the work of [136] and [137], proposed BiLSTM as one of the best approach for rainfall prediction.

Multilayer Perceptron (MLP) is a basic feed forward neural network architecture, which is used to model complex relationships between independent and dependent variables. CNNs are particularly useful for capturing spatial patterns in satellite data, such as identifying patterns of weather conditions. However, it can also be adapted to handle sequential data like time series rainfall. LSTMs are well-suited for time series analysis and can effectively model temporal and captures long-term dependencies in sequential data. They can learn to remember information over long time periods, making them useful for predicting rainfall patterns based on historical data. BiLSTMs can capture contextual knowledge’s from both past and future periods. This can be valuable in rainfall prediction, as it allows the model to consider the influence of both past and future observations. Multiple experiments were performed with different datasets and various architectures for each model in this work.

Once the four major architectures are selected, the best algorithm from the four is selected by experimentally evaluating their performance. However, before trying to compare the four algorithms, another mini-experiment is conducted to find the optimal parameters within each algorithm. To find the optimal parameters that use algorithms to predict rainfall accurately, different experiments are conducted for each algorithm as described below.

To determine the optimal values for the parameters in the selected deep learning techniques, we conducted a series of experiments on the pre-processed dataset that is prepared for model development. The goal is to assess the performance of the models under various parameter settings and identify the configurations that generate the best results. We systematically varied the parameter values, such as learning rate, batch size, number of layers, activation functions, regularization techniques, and other hyperparameters specific to the chosen deep learning techniques. For each combination of parameter values, we trained the models on the dataset and evaluated their performance using appropriate evaluation metrics, such as RMSE, MAE, and MSE. Then each trained models are compared and well performed values are chosen as optimal values for parameters as described below.

For MPL experiments were conducted with a random state of 1, with Maximum iteration of 200, 1000, 5000, and other setting default. Here, the default maximum iteration of MLPRegressor from Scikit-learn is 200 iterations. But, for our study, parameters with a random state of 1, with a Maximum iteration of 1000, and other parameters as a default are the best for building an MLP model. In deep learning, neural networks have input, hidden, and output layers that are all made up of different layers. Extracting features and representations from the input data is the responsibility of the hidden layers. For the CNN, LSTM, and BiLSTM experiments were conducted with hidden layers starting from one hidden layer up to four hidden layers to select the best suitable numbers of hidden layers for the models. Three hidden layers with 128 neurons for first layer, 64 neurons for second layer, and 32 neurons for the third layer respectively are good parameters for each model. To select the best activation functions for the models, we tried using “relu”, “Tanh”, and “Sigmoid” functions, and the “relu” activation function is the best than others. Also, Loss functions were selected by experiment, from “rmse”, “mse”, and “mae”. Because of the regression problems, we performed experiments to select loss functions from “rmse”, “mse”, and “mae” and mse has outperformed. On the other hand, the Adam optimizer was used on all models, because the Adam optimizer adapts the learning rate for individual parameters by considering estimates of both the first and second moments of the gradients.

Selecting an optimal number of epochs may vary depending on the specific characteristics of the datasets, model architecture, and the problem complexity. It's recommended to experiment with different epoch values and assess their influence of the model's performance to find the best choice for the rainfall prediction task. In this research, we performed an experiment with 100, 50, 30, and 10 epochs for LSTM, BiLSTM, and CNN algorithms. However, the experiment with 30 epochs is the best to develop the model with the listed algorithms. These all experiments were conducted based on the training dataset and the performances were evaluated. The combined dataset contains 11324 instances and 8 meteorological variables. The training datasets that used to develop models are 80% of all datasets and the remaining 20% is used for testing or evaluation purposes.

After different experiments were applied; the following hyper-parameters are selected to design and develop the best rainfall models.

Table 4.6| The summary of the selected optimal parameters

Multi-Layer Perceptron (MLP)	Optimal values for parameters
Random state	1
Maximum iteration	1000
Other setting	Default
Convolutional Neural Network (CNN)	
Three hidden layers	The First, the second, and the third layers are 128, 64, 32 respectively
Activation function	Relu
Kernel size	2
Loss function	mean_squared_error (mse)
Optimizer	Adam
Epochs	30
Long Short Term Memory (LSTM) and BiLSTM	
Three hidden layers	The First, the second, and the third layers are 128, 64, 32 respectively
Activation function	Relu

Dropout	0.3
Loss function	mean_squared_error (mse)
Optimizer	Adam
Epochs	30
Batch size	32

4.3. Chapter Summary

The above sections of this chapter discuss each step of data collection and data preprocessing techniques (i.e. handling missing values, data normalization, data splitting and etc.) in detail. These are aimed at achieving the final findings. There are a lot of challenges that, this study faced in developing the best model for rainfall prediction by the given datasets. Specially, it is difficult to collect satellite data from global models, as well as to extract needed values from the image that helps to improve the accuracy of the rainfall prediction model. Unclean and noisy data made the whole process extremely exhausting for the data engineering phase, it is time-consuming specially, to integrate and transform the dataset. To select best algorithm for rainfall prediction from the four is selected by experimentally evaluating their performance. Also, different experiments were performed to select the best hyper-parameters for selected techniques. The next chapter will cover the experiment and evaluations of the model.

CHAPTER Five

Experiment and Evaluation

5.1. Introduction

In this section, we have discussed on the experimental setup, tools and evaluation metrics as well as the experimental protocols which helps to develop models for rainfall prediction. Also, discussions of experiment of the results for the selected four deep learning architectures are performed. Each model was developed based on the parameters listed in chapter four. Then each developed model was evaluated based on RMSE, MSE, and MAE to identify the best-predicting model for the given datasets. Furthermore, models are developed for weather station and satellite data separately. Finally, best model for rainfall prediction is selected.

5.2. Experimental Setup

5.2.1. Implementation Tools

For the implementation of the rainfall prediction model, many tools and techniques are used. Different libraries and editors that used in this model building process are discussed below; Anaconda is used for data preprocessing (from rewriting the Excel file up to handling missing values and outlier detection and resolving) purpose. To pre-process weather station and satellite data for the coding aspect, we used the Jupyter Notebook. For the model implementation, we used **Google Colab** which is a free cloud-based platform developed by Google that permits users to write and run Python code collaboratively. The implementation has used Google Colab with GPU of hardware accelerator 2.1GB. Google Colab comes pre-installed with a wide range of Python libraries, including common data science and machine learning libraries such as Pandas, Scikit-learn, NumPy, TensorFlow, etc. [138].

TensorFlow is used to develop models that take into account various factors that influence rainfall prediction [139]. **Keras** provides support for a widespread range of neural network approaches, including feed-forward, recurrent, convolutional, and combinations of models [140]. It also supports both CPU and GPU acceleration, which allows for faster training of models on large datasets. **Scikit-learn** offers a range of tools for data engineering, feature selection, and model evaluation, as well as a variety of machine learning and deep learning algorithms. It involves converting the raw data into a format suitable for deep learning algorithms and selecting relevant features that are likely to have a important influence on the predictions. The library is built upon SciPy (Scientific Python) and includes Pandas, NumPy, Matplotlib, and etc.

5.2.2. Evaluation Metrics

Each developed model was evaluated based on RMSE, MSE, and MAE to identify the best-predicting model for the given datasets. These are the common evaluation metrics for regression problem to assess the performance of developed models. They are also applied in similar works [108], [101], [104], [107]. The metrics are discussed below;

Root Mean Squared Error (RMSE): is a widely used metric for assessing regression models. It measures the average deviation between the predicted rainfall values and the real observed values, taking into account the squared differences between them. RMSE penalizes large errors more heavily, making it sensitive to outliers [108]. By calculating the square root of the mean squared errors, it provides a measure of the average absolute deviation between predicted and observed rainfall, giving an indication of the model's overall accuracy.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2}$$

Where, n = total number of data points, x_i = actual rainfall values and y_i = predicted rainfall values

Mean Squared Error (MSE): s a widely adopted evaluation metric for regression tasks. It calculates the average of the squared differences between the observed rainfall values and the corresponding predicted values. Like RMSE, MSE also penalizes large errors, but without taking the square root. MSE is calculated by taking the average of the squared differences between the predicted rainfall and actual values.

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

Where, n = total number of data points, x_i = actual rainfall value, y_i = predicted rainfall value

Mean Absolute Error (MAE) is a regression metric that calculates the average absolute difference between the corresponding predicted values and the actual rainfall values. MAE is less sensitive to outliers and provides a direct indication of the average absolute deviation between predictions and observations [104]. MAE is calculated by taking the average of the absolute differences between the predicted rainfall value and the actual values.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|x_i - y_i|$$

Where, N = total number of data points, x_i = actual value and y_i = predicted value

When the RMSE, MSE, and MAE values are small, it indicates that the model excels at prediction. Equally, larger RMSE, MSE, and MAE values suggest that the model might encounter difficulties in specific areas. A perfect predictor of output for a given input data would yield an RMSE, MSE, and MAE of 0.

5.2.3. Experimental Protocol

The experiment in this chapter is presented in the following protocol. The first is to demonstrate to what extent combining station data with satellite data improves the performance of the model. The developed models are calculated based on the above metrics to identify the best predictive model for rainfall prediction. Moreover, another experiment is conducted by taking each of the sources separately. In these cases, good performing models i.e. LSTM and BiLSTM is presented, and the results of the remaining two approaches are presented in the appendix. The second experiment is meant to select then best deep learning model from the four models. Within this, another experiment is first undertaken to find an optimal parameters.

5.3. Discussion of Experimental Results

The discussion begins by presenting what has been identified as critical finding in the experiments. We experimented with the combinations of both datasets to select well-performing models. To merge or combine both station and satellite data we used two approaches; the first experimental approach is replaced the missing value in station data with satellite data, and the rest of the parameters were merged into station data Excel file. The second approach is taking their mean for the same parameters and combining the rest into one Excel file. From both data, the second approach which is mean-based imputations gives a good result when compared with the first way. Finally, we used a dataset that combined the mean from the two datasets. In this section, four different deep learning models were developed using the selected parameters.

5.3.1. Model 1: Multi-Layer Perceptron (MLP)

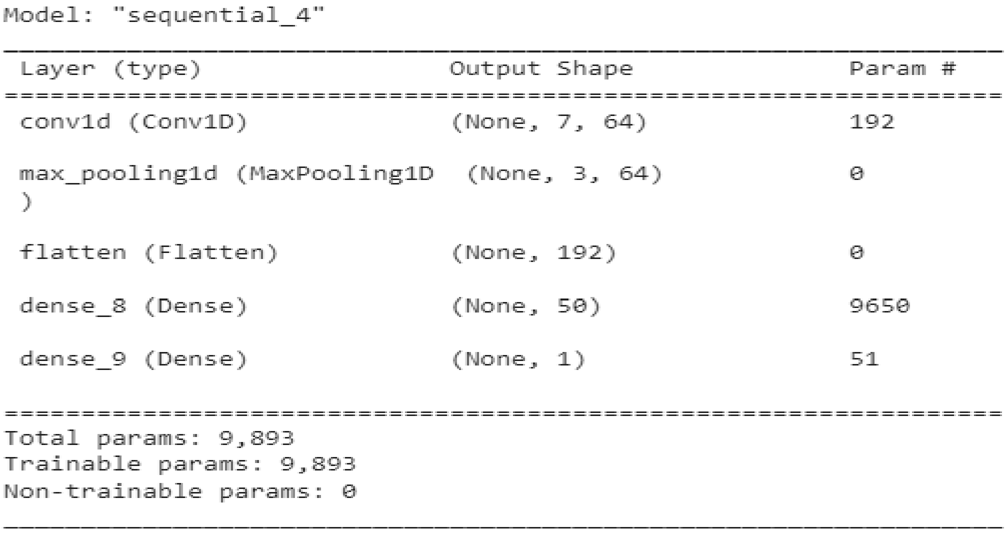
In the case of the MLP configuration, the best result was adjusted with the following combination of values: MLPRegressor form sklearn or Scikit Learn, one random state, and

1000 maximum iterations of training were used. MLPRegressor is a class in sklearn that implements a multi-layer perceptron (MLP) model for regression tasks. The model was trained and tested using only significant variables, while ignoring any insignificant parameters.

The multilayer perceptron (MPL) was evaluated based on Root Mean Square Error (RMSE), Mean Squared Error (MSE), and Mean Absolute Error (MAE). The model performed; with MSE value of 0.000727, MAE value of 0.0158, and RMSE value of 0.0269 using Addis Ababa weather data, with MSE value of 0.00357, MAE value of 0.0344, and RMSE value of 0.0613 using Adama weather data, and with MSE value of 0.00399, MAE value of 0.0308, and RMSE value of 0.0609 using Bishoftu weather data.

5.3.2. Model 2: Convolutional Neural network (CNN)

A Convolutional Neural Network (CNN) is a deep learning model renowned for its exceptional capability in analyzing visual data, including images or spatially structured data. CNNs are specifically designed to extract meaningful features and patterns from such data, making them highly effective for tasks involving visual analysis and recognition. However, it can also be applied to other types of data with spatial or temporal structure, such as weather data, including rainfall prediction. The approach used Max pooling, to down-sampling operation in CNNs utilize techniques that reduce the spatial dimensions of feature maps while preserving essential features. In this study, a CNN is well-suited for analyzing spatial and temporal patterns in data to predict rainfall using defined parameters in chapter four;



```
Model: "sequential_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 7, 64)             192

 max_pooling1d (MaxPooling1D  (None, 3, 64)            0
 )

 flatten (Flatten)           (None, 192)               0

 dense_8 (Dense)             (None, 50)                9650

 dense_9 (Dense)             (None, 1)                 51

=================================================================
Total params: 9,893
Trainable params: 9,893
Non-trainable params: 0
_________________________________________________________________
```

Figure 5.1| The summary for the CNN model

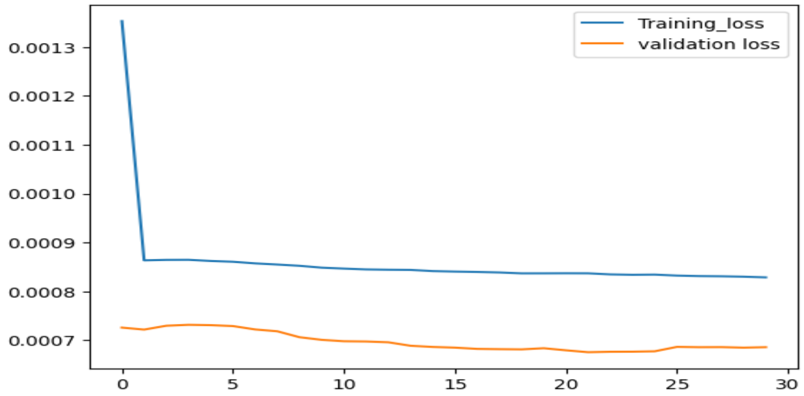
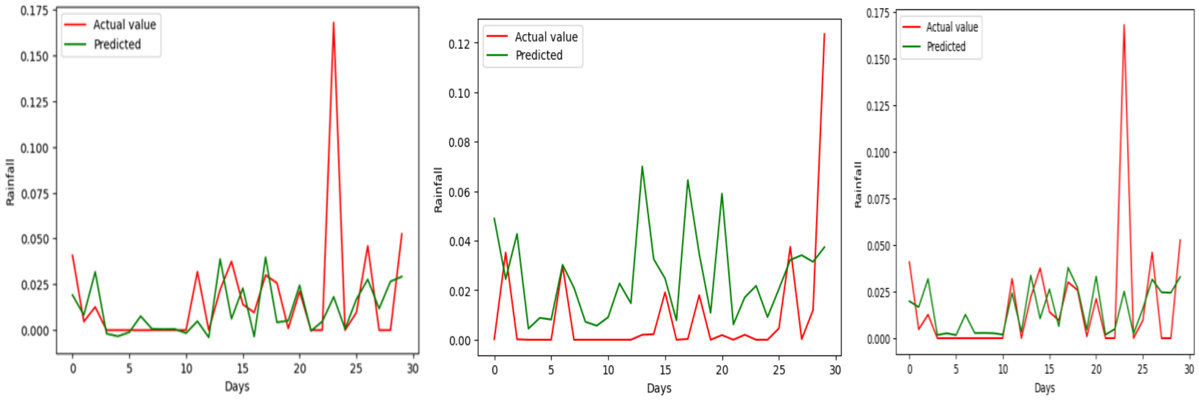


Figure 5.2| Validation and training using MSE for the CNN model

The training loss metric evaluates the performance of the model on the training data, while the validation loss metric assesses the model's generalization ability on unseen data. In Figure 5.2, the training loss is higher than the validation loss, indicating that the model is not overfitting to the training dataset. Consequently, we can proceed with developing the model using the aforementioned parameters. Figure 5.3 below illustrates the comparison between the actual values and the corresponding predicted values for each data point.



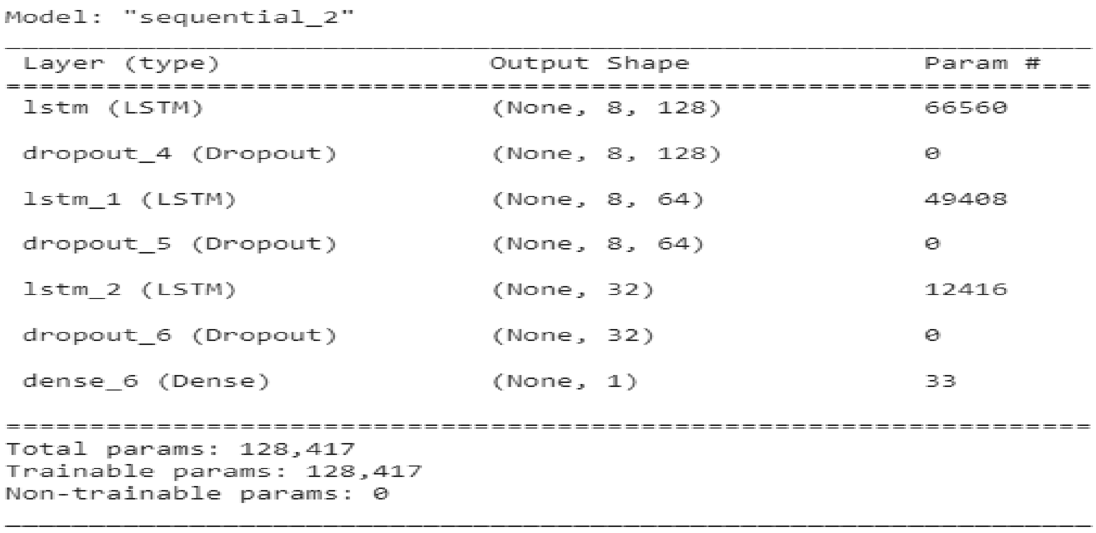
a) Addis Ababa b) Adama data c) Bishoftu data

Figure 5.3| Evaluation of CNN model with testing data

As indicated in Figure 5.3 the CNN model prediction performance metrics, with an MSE value of 0.00069, MAE value of 0.0127, and RMSE value of 0.02626 using the Addis Ababa dataset, with MSE value of 0.00345, MAE value of 0.0336, and RMSE value of 0.0591 using Adama dataset, and with MSE value of 0.00407, MAE value of 0.0380, and RMSE value of 0.0641 using Bishoftu dataset.

5.3.3. Model 3: Long Short Term Memory (LSTM)

LSTM is a specific architecture designed to tackle the vanishing gradient problem and capture long-term dependencies within sequential data. In this study, different components of parameters were linked together to train the LSTM model to predict accurately rainfall; Rectified Linear Unit (ReLU) introduces non-linearity to the network, allowing it to learn complex relationships in the data, Dropout function is a regularization technique used to prevent overfitting in neural networks, Adam optimizer is an optimization algorithm that automatically adjusts the learning rate during training, which can speed up convergence and improve training performance, and Dense layer is typically added after the LSTM layer to process the output of the LSTM units and map it to the desired output dimension. Figure 5.4 shows the model layer and parameters for LSTM.



```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 8, 128)            66560

 dropout_4 (Dropout)         (None, 8, 128)            0

 lstm_1 (LSTM)               (None, 8, 64)             49408

 dropout_5 (Dropout)         (None, 8, 64)             0

 lstm_2 (LSTM)               (None, 32)                12416

 dropout_6 (Dropout)         (None, 32)                0

 dense_6 (Dense)             (None, 1)                 33

=================================================================
Total params: 128,417
Trainable params: 128,417
Non-trainable params: 0
_________________________________________________________________
```

Figure 5.4| The summary layer of LSTM model

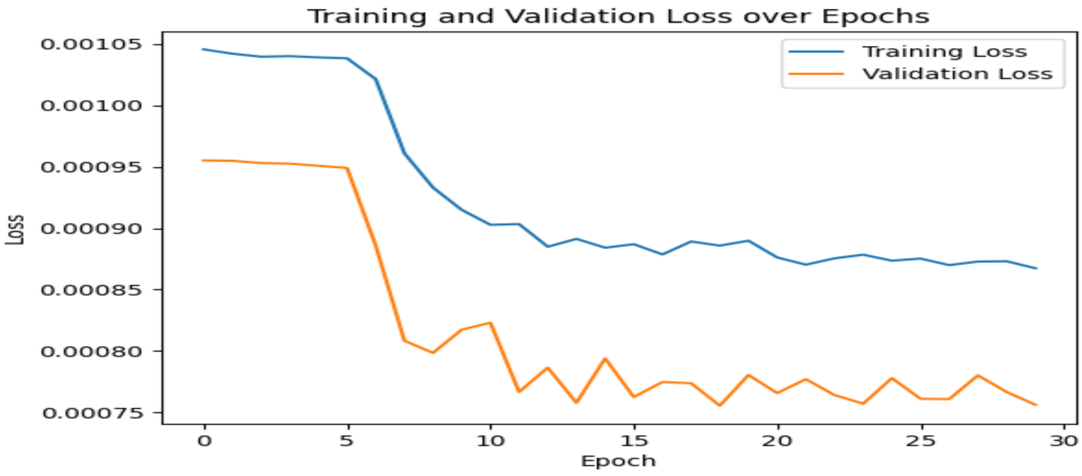
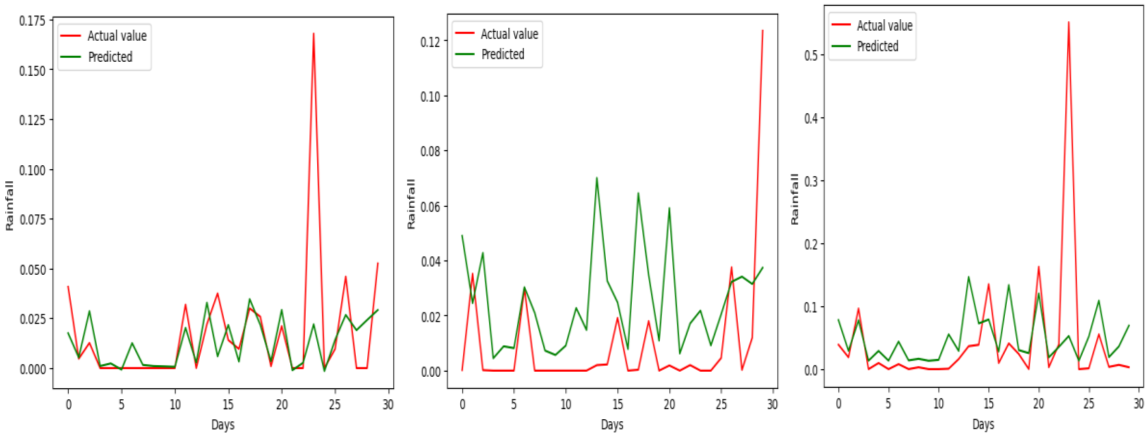


Figure 5.5| Validation and training using MSE for the LSTM model

Here also the training loss is greater than the validation loss, which means that the model is not overfit for the training data. Therefore, we can develop the model for rainfall prediction using the training data. Figure 5.6 shows the comparison between Actual values and the predicted values for each data.



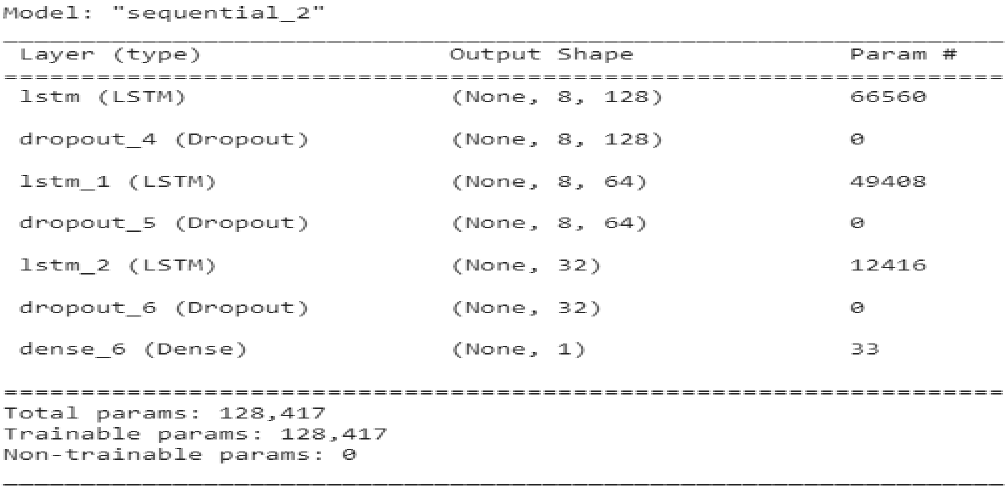
a) Addis Ababa data b) Adama data c) Bishoftu data

Figure 5.6| Evaluation of LSTM model with testing dataset

As indicated in the figure 5.6 the LSTM model prediction performance metrics, with MSE value of 0.00066, MAE value of 0.0110, and RMSE value of 0.0260 using Addis Ababa dataset, with MSE value of 0.0033, MAE value of 0.0299, and RMSE value of 0.058 using Adama dataset, and with MSE value of 0.0385, MAE value of 0.0268, and RMSE value of 0.0620 using Bishoftu dataset.

5.3.4. Model 4: Bidirectional LSTM (BiLSTM)

Bidirectional LSTM maximizes data utilization by processing time steps in both the forward and backward directions. It achieves this by duplicating the initial recurrent network in the architecture, resulting in two adjacent layers. The input is passed as it is to the first layer, while a reversed copy of the input is provided to the second layer. In this study, different parameters were used such listed above in chapter four. In BiLSTM; the same hyperparameters were used in order to develop the best model. Each BiLSTM layers and dropout layers were summarized in the Figure 4.7.



```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 8, 128)            66560

 dropout_4 (Dropout)         (None, 8, 128)            0

 lstm_1 (LSTM)               (None, 8, 64)             49408

 dropout_5 (Dropout)         (None, 8, 64)             0

 lstm_2 (LSTM)               (None, 32)                12416

 dropout_6 (Dropout)         (None, 32)                0

 dense_6 (Dense)             (None, 1)                 33

=================================================================
Total params: 128,417
Trainable params: 128,417
Non-trainable params: 0
_________________________________________________________________
```

Figure 5.7| The summary layer of BiLSTM model

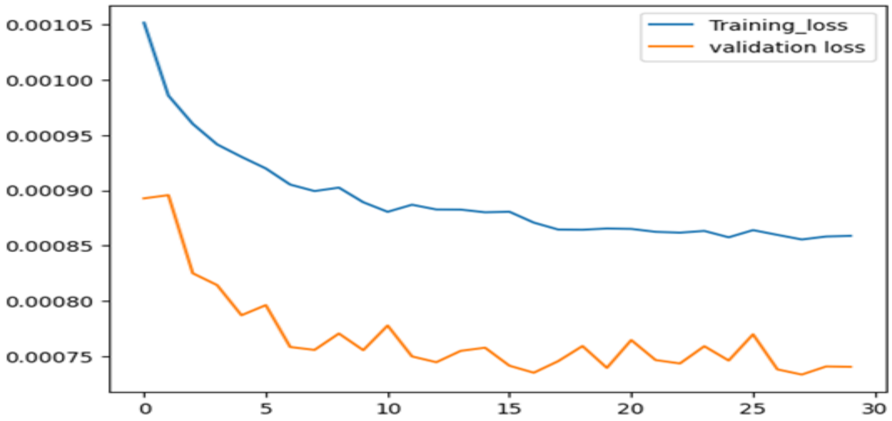
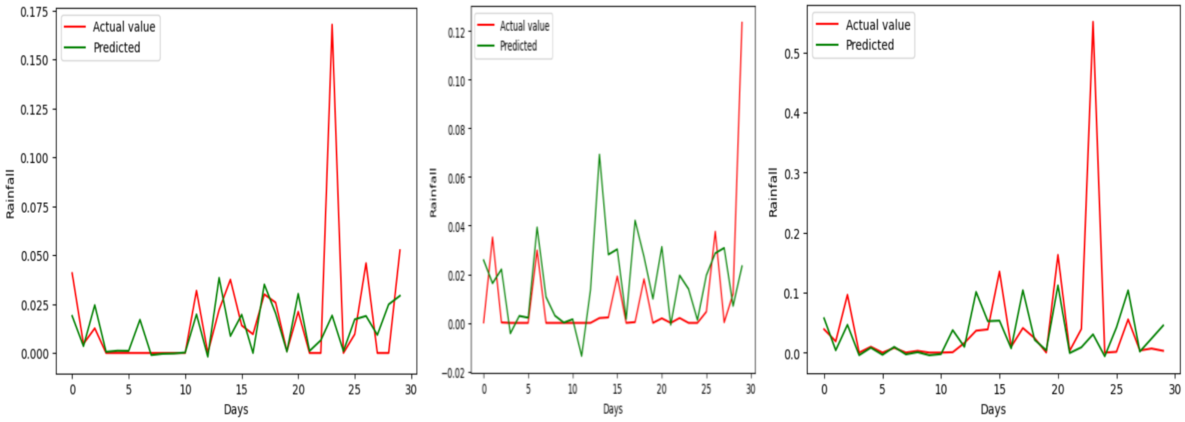


Figure 5.8| Validation and training using MSE for the BiLSTM model

Also in BiLSTM, the training loss is greater than validation loss, which means that the model is not overfit for the training data. Accordingly, we can develop the model for rainfall prediction using the training data. Figure 5.9 shows the comparison between Actual values and the predicted values for each data.



a) Addis Ababa b) Adama c) Bishoftu data

Figure 5.9| Evaluation of BiLSTM model with testing dataset.

As indicated in the Figure 5.9 the BiLSTM model prediction performance metrics, with MSE value of 0.00064 MAE value of 0.0105, and RMSE value of 0.0254 using Addis Ababa dataset, with MSE value of 0.0032, MAE value of 0.0253, and RMSE value of 0.0562 using Adama dataset, and with MSE value of 0.0038, MAE value of 0.0266, and RMSE value of 0.060 using Bishoftu dataset.

To compare developed rainfall prediction models based on their values of evaluation metrics, as shown in the Table 5.1 below from the experiment performed above. In the table, A.A represents Addis Ababa data, AD represents Adama data, and BSH represents Bishoftu data.

Table 5.1| The RMSE, MSE, and MAE values for deep learning methods

Evaluation Metrics	RMSE			MSE			MAE		
	A.A	AD	BSH	A.A	AD	BSH	A.A	A.D	BSH
MLP	0.0270	0.0613	0.0609	0.000727	0.00357	0.00399	0.0158	0.0344	0.0308
CNN	0.0263	0.0591	0.0641	0.00069	0.00345	0.00407	0.0127	0.0316	0.038
LSTM	0.0260	0.0580	0.0620	0.00066	0.00330	0.00385	0.0110	0.0299	0.0268
BiLSTM	0.0254	0.0562	0.060	0.00064	0.0032	0.0038	0.0105	0.0253	0.0266

As shown in Table 5.1 results were obtained from each experiment (selected deep learning techniques) based on three different data sets. In order to select the best model for rainfall prediction for the study area the average of the evaluation metrics for three places were taken to generalize the models. Here, the average values for each model using evaluation metrics RMSA, MSE, and MAE for MLP are 0.0497, 0.00276, and 0.027 respectively. For CNN, the average values for RMSE, MSE, and MAE are 0.0495, 0.00274, and 0.0269 respectively. For LSTM the average values for RMSE, MSE, and MAE are 0.04866, 0.0026, and 0.0226 respectively. For BiLSTM the average values for RMSE, MSE, and MAE are 0.0472, 0.0025, and 0.021 respectively.

Table 5.2| The average RMSE, MSE, and MAE values for deep learning methods

Deep Learning Architectures	RMSE	MSE	MAE	Remark
MLP	0.0497	0.00276	0.027	4
CNN	0.0495	0.00274	0.0269	3
LSTM	0.04866	0.0026	0.0226	2
BiLSTM	0.0472	0.0025	0.021	1

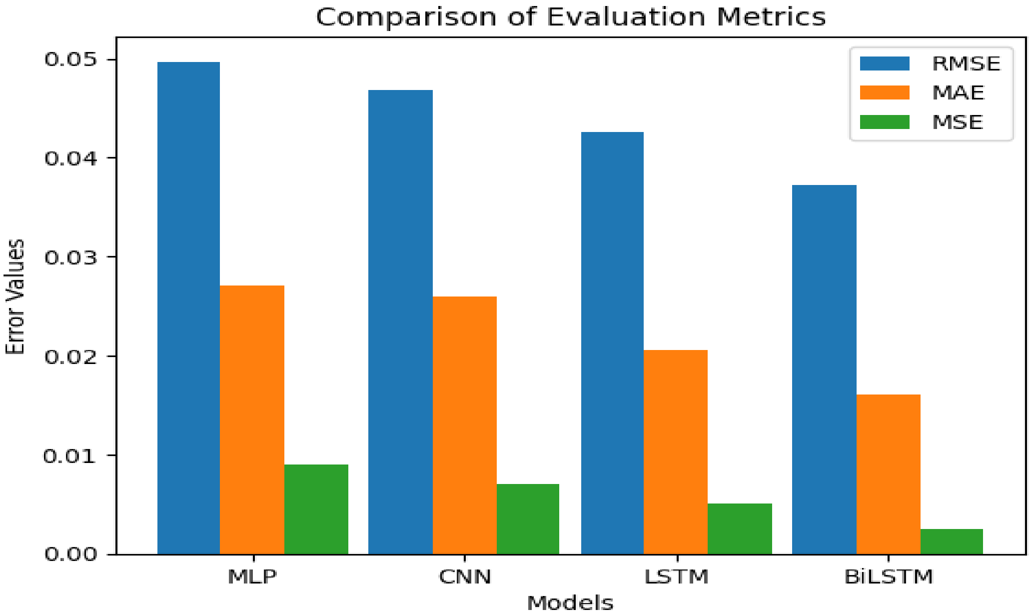


Figure 5.10| Deep learning models comparison graph

The above experiments were conducted to find the best deep learning model for predicting the rainfall for the Addis Ababa, Bishoftu, and Adama areas. In those experiments, the same set of experiments was performed with the combined datasets (weather station and satellite data). As the average values from Table 5.2 and Figure 5.10 shows, BiLSTM outperformed LSTM, CNN, and MLP by 0.0015, 0.0023, and 0.0025 respectively for RMSE, 0.0001, 0.00024, and 0.0026 respectively for MSE, and 0.0016, 0.0059, and 0.006% respectively for MAE.

When compared to the results, the BiLSTM (Bidirectional LSTM) provides good results than the other deep learning algorithms for predicting rainfall for sequential datasets. Even if BiLSTM is the best approach to predict rainfall, LSTM is also a good approach for rainfall prediction than CNN and MLP. Therefore, when dealing with large datasets and complex data, BiLSTM tends to outperform other deep learning algorithms in terms of accuracy.

5.3.5. Experiments with Station and Satellite Data Separately

In this experiment, we used weather station data and satellite data separately. The section shows the results of the two outperformed models in the above experiment. These are LSTM and BiLSTM models, in order to show the differences in the model developed with station and satellite separately. Figure 5.11 shows the validation and training loss based on the loss function of mse by using station dataset.

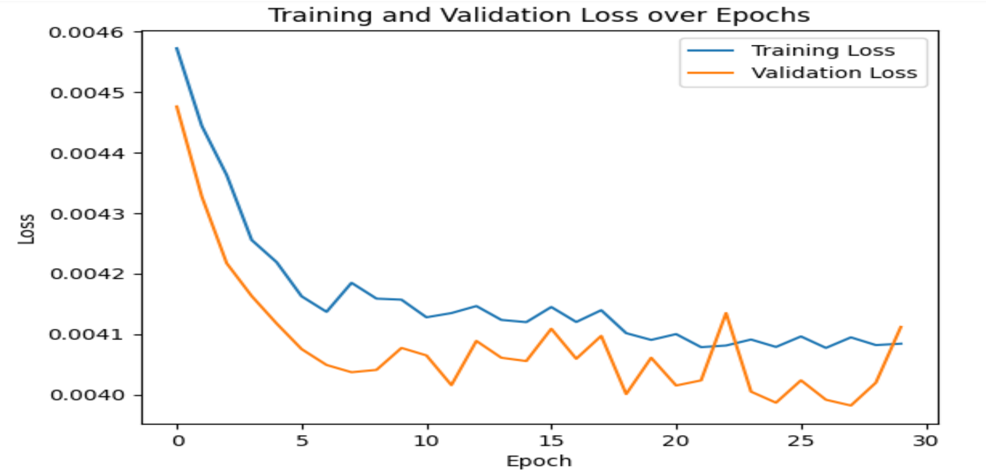


Figure 5.11| Validation and training loss using MSE for the BiLSTM model

Figure 5.11 shows that training and validation loss errors are low and the model is not overfitting so that we can develop a deep learning model using station data. Figures 5.12 a) and b) show the sample model results for Actual and Predicted values of station data using LSTM and BiLSTM models.

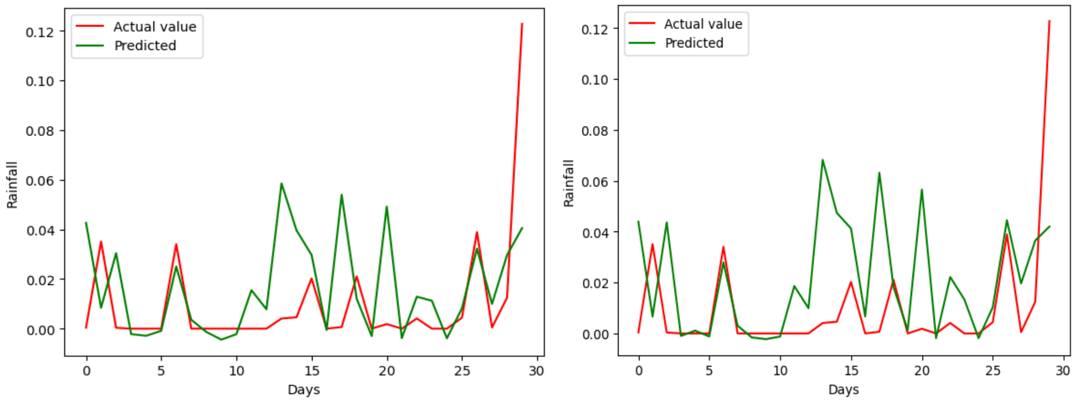


Figure 5.12| Comparison of actual and predicted rainfall values for station data

To developed models with station data we used the optimal parameters selected in chapter four. The evaluation LSTM model with n MSE value of 0.00338, MAE value of 0.0246, and

RMSE value of 0.0582 and BiLSTM model with MSE value of 0.0032, MAE value of 0.0278, and RMSE value of 0.057

Here also LSTM and BiLSTM models were displayed in order to show the differences in model with satellite data. Figure 5.13 shows the validation and training loss based on the loss function of mse by using satellite dataset.

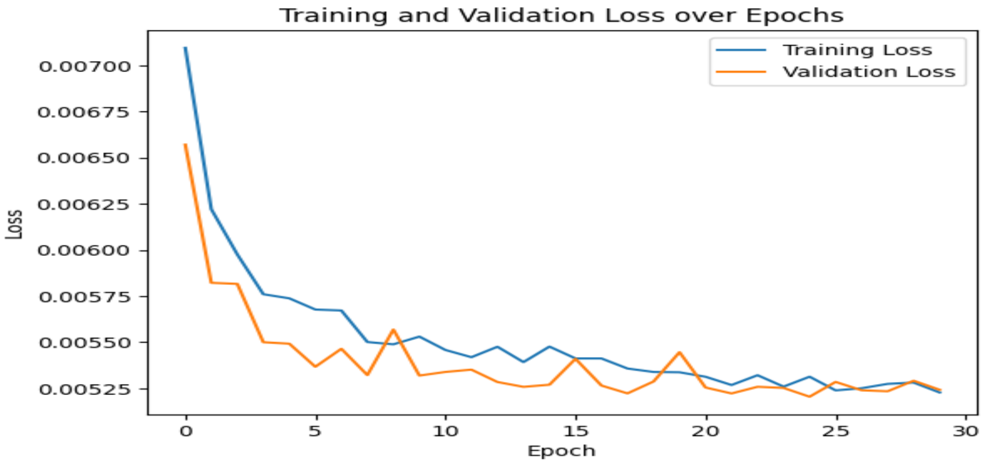
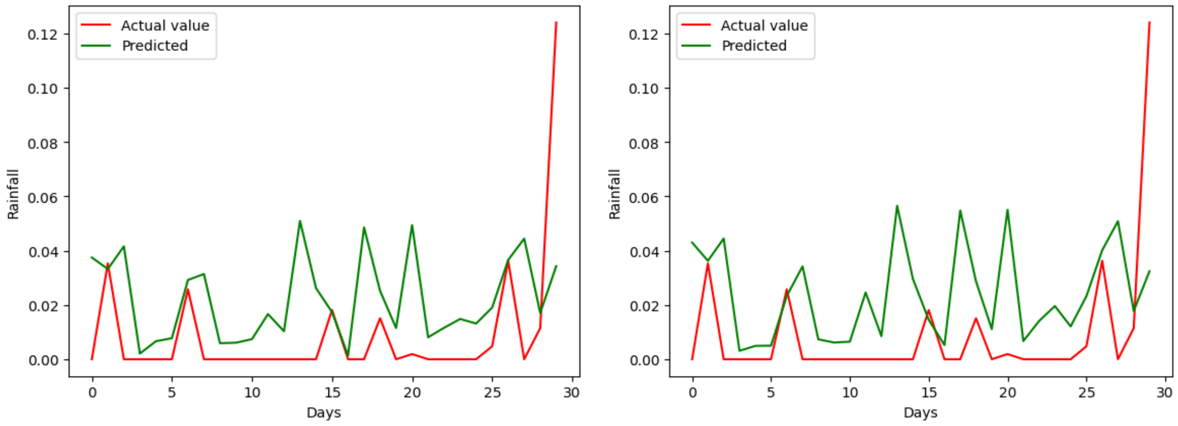


Figure 5.13| Validation and training using MSE for the BiLSTM model

Figure 5.13 shows that training and validation errors are low and the model is not overfitted so that we can develop a deep learning model with satellite dataset. Figures 5.14 a) and b) show the sample model results for Actual and Predicted rainfall values of satellite data using LSTM and BiLSTM models.



a) LSTM b) BiLSTM

Figure 5.14| Comparison of actual and predicted rainfall values for Satellite data.

To developed models with station data we used the optimal parameters selected in chapter four. The evaluation of LSTM model with an MSE value of 0.0042, MAE value of 0.0359, and RMSE value of 0.065 and BiLSTM with MSE value of 0.00372, MAE value of 0.0304, and RMSE value of 0.061.

This Experiment indicates that; for the data collected from the meteorological station LSTM and BiLSTM Models are the best and most appropriate models to predict rainfall. But for weather data collected from satellites, the CNN model outperformed other deep learning approaches and it's an appropriate model for rainfall prediction. For both dataset, the remaining approach the results are presented in the appendix. In General, the table in the appendix shows the summary of the result for both datasets using MLP, CNN, LSTM, and BiLSTM.

5.4. Chapter Summary

The above sections of this chapter are describes the experimental tools, protocols, and evaluation metrics, that help to develop a rainfall prediction model using deep learning and help to evaluate the developed models for their accuracy and performance. Furthermore, the chapter discusses each developed model based on a combined station and satellite data. Also, other models were developed using station and satellite data separately. Finally, the chapter concludes that the BiLSTM model outperformed LSTM, CNN, and MLP for combined weather station and satellite data.

CHAPTER SIX

CONCLUSION AND RECOMMENDATION

This chapter summarizes the conclusions, recommendations and future works of the study. First, it summarises the results of the study and conclusions drawn from the experiment are presented. Lastly, the recommendations and future works are presented.

6.1. Summary and Conclusion

The significance of this kind of research for Smart agriculture, Climate Change Adaptation, transportation, flood warning, and other industrial sectors helps to make more profit. It has become known that predicting rainfall is the most important in developing countries like Ethiopia because agriculture is dependent on rain. Hence, rainfall is influenced by a variety of meteorological variables and is a difficult non-linear system, as well as the most essential part of agriculture. Early access to weather predictions and information about rainfall timely and amount can be highly advantageous for farmers. In this study, a rainfall prediction model was developed by combining weather station data obtained from the National Meteorological Agency of Ethiopia with satellite data from TAMSAT v3.1 and JRA-55. The aim was to provide farmers with valuable insights for better planning and decision-making regarding agricultural activities. These developed models need to achieve high accuracy values by using multiple meteorological parameters. The data engineering techniques have been performed in this research to make the collected data clean and to make it suitable for deep learning models. To train and test the model, the pre-processed data is divided into 80% for training and 20% for testing. Also, this study has applied feature selection, which means selecting important features to develop an accurate rainfall model.

Four deep learning approaches were chosen in this study, such as; Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short Term Memory (LSTM), and Bidirectional LSTM (BiLSTM). The main reason that we chose these algorithms has; the first reason, Ability to handle complex relationships which means allowing them to capture the dependencies and interactions between different input variables more effectively. The second reason, Handling high-dimensional data efficiently and automatically. Especially, LSTMs and BiLSTMs are particularly suitable for capturing temporal dependencies in rainfall prediction. They can model long-term dependencies and memory within sequences of weather station data format. Also, these algorithms have shown strong generalization capabilities, enabling them to learn from large datasets and generalize well to unseen data.

The research questions were answered based on the experiments performed in the above chapter. The first research question is, “Which data engineering techniques should be applied to prepare the dataset?” in the research different techniques were used, such as the missing values were handled using mean imputation based on the three season of Ethiopia, outlier were detected using z-score and resolved using mean imputation, and etc. The second research question is, “What are the most effective meteorological variables for predicting rainfall from the combination of weather station and satellite data?” In this study, SUNHRS (solar radiation), Cloud cover, Air Pressure, Relative Humidity (RELHUM), Minimum temperature, Maximum temperature, and wind speed are the basic weather variables to build a rainfall prediction model. The third research question is, “To what extent do deep learning approaches predict rainfall based on combined weather station and satellite data in Ethiopia?” the experiments we performed shows that, Bidirectional LSTM (BiLSTM) model has the lowest error of RMSE, MSE, and MAE of 0.0472, 0.0025, and 0.021 respectively. So this model was chosen as the best model for predicting rainfall for combined weather station and satellite data. The fourth question is, “Which deep learning model architecture is more suitable to predict rainfall patterns using both station and satellite data?” Here, Bidirectional LSTM (BiLSTM) model gives best results than the other deep learning algorithm for predicting rainfall for the combined weather station and satellite datasets. To conclude, rainfall prediction should be the concern of all (National and regional meteorological agencies, Disaster management authorities, Agricultural sectors (policymakers, farmers), Water resource management, etc.). If the amount of the rain that drops in specific area were not known, then there will be a significant waste or loss of properties. As a result, accurate rainfall prediction model have an adverse and unintended impact.

In this work, we faced lots of challenges such as; integrating weather station and satellite data can be complex due to differences in spatial and temporal resolutions, as well as they have different data formats. In order to solve this problem we used the climate data tool (CDT) and python code to convert the satellite data into station format and to be in the same temporal resolutions and the same format with station data. The low quality of weather station data and the issues of availability of satellite data were other challenges that we faced. Especially, station data has a high amount of missing values, to handle these missing values we used seasonal-based mean imputation techniques. On the other way, we used the dropout technique to mitigate overfitting and enhance the generalization of the models, because deep learning models are prone to overfitting, especially when dealing with complex datasets. Also, Computational resources and time were another challenge hence; the models can be computationally intensive, requiring substantial computational resources, memory, and time.

The contribution of the study is outlined as follows;

- The level of accuracy of rainfall prediction increased by utilizing deep learning models that combine data from meteorological stations and satellites. The model identifies complex relations and patterns in data, which could result in more precise and dependable rainfall prediction.
- The proposed model is useful in agricultural planning and resource allocation, empowering farmers to choose crops, set planting dates, and manage water resources. This can increase agricultural productivity and reduce farmers' sensitivity to variations in rainfall.
- As our country is disposed to seasonal floods, particularly during the rainy season. So the model can contribute to the development of robust flood early warning systems by accurately predicting rainfall patterns.
- This result can be used to enhance reservoir management, irrigation scheduling, and water allocation, ensuring sustainable and efficient utilization of water resources across the country.

6.2. Recommendation

This study recommendation could be used to design and develop a predictive rainfall model to improve the accuracy of rainfall prediction. Finally, based on the research findings, the study has the following recommendations;

- Rainfall patterns and conditions evolve over time, requiring continuous updates to the proposed model. User of this model must establish a mechanism for incorporating new data and retraining the model periodically to maintain its predictive performance.
- The study recommends NMA of Ethiopia to expand the network of collection of weather station data and satellite data. Because, there is a limitation on both data types.
- In order to provide farmers with precise rainfall forecasts, we suggest incorporating the proposed model into smart agricultural platforms and systems. In order to support

crop planning, pest management, irrigation scheduling, and agricultural productivity and resource management.

- In this study, the researcher tested the accuracy of the developed model using three different datasets; station, satellite, and by combining station and satellite data. From those, the combined station and satellite data were performed well, this shows having more data is good to develop a better performing model for rainfall prediction and if the researcher gets more data it should improve the performance of the proposed approach.

6.3. Future Work

Some of future research identified throughout the study may be carried out here;

- In this study, only three city datasets were collected and analysed, in the future, another cities and areas dataset can be collected to develop more accurate model.
- Also, further study is required to handle additional weather variables including regional and global weather trends such as, Evapotranspiration, Sea Surface Temperatures, Water Vapor Content, and Water Vapor Content.

Reference:

[1] K. Fekadu, “Ethiopian Seasonal Rainfall Variability and Prediction Using Canonical Correlation Analysis (CCA),” *Earth Sci.*, vol. 4, no. 3, p. 112, 2015, doi: 10.11648/j.earth.20150403.14.

[2] D. Korecha and A. Sorteberg, “Validation of operational seasonal rainfall forecast in Ethiopia,” *Water Resour. Res.*, vol. 49, no. 11, pp. 7681–7697, 2013, doi: 10.1002/2013WR013760.

[3] A. Geetha and G. Nasira, “Artificial neural networks’ application in weather forecasting – Using RapidMiner,” *Int. J. Comput. Intell. Informatics*, vol. 4, no. 3, pp. 177–182, 2014, [Online]. Available: http://www.periyaruniversity.ac.in/ijcii/issue/Vol4No3December2014/IJCII 4-1-152.pdf.

[4] S. M. Yimer, A. Bouanani, N. Kumar, B. Tischbein, and C. Borgemeister, “Assessment of Climate Models Performance and Associated Uncertainties in Rainfall Projection from CORDEX over the Eastern Nile Basin, Ethiopia,” *Climate*, vol. 10, no. 7, 2022, doi: 10.3390/cli10070095.

[5] X. Ziniu, L. Bo, L. Hua, and Z. De, “Progress in climate prediction and weather forecast operations in China,” *Adv. Atmos. Sci.*, vol. 29, no. 5, pp. 943–957, 2012, doi: 10.1007/s00376-012-1194-9.1.Progress.

[6] Z. Wang and A. B. M. Mazharul Mujib, “The Weather Forecast Using Data Mining Research Based on Cloud Computing.,” *J. Phys. Conf. Ser.*, vol. 910, no. 1, 2017, doi: 10.1088/1742-6596/910/1/012020.

[7] D. Manatsa, W. Chingombe, and C. H. Matarira, “The impact of the positive Indian Ocean dipole on Zimbabwe droughts Tropical climate is understood to be dominated by,” *Int. J. Climatol.*, vol. 2029, no. March 2008, pp. 2011–2029, 2008, doi: 10.1002/joc.

[8] N. National Meterology Service Agency, “Climate & Agro Climate Resources of Ethiopia. NMSA Meteorological Research Report Series,” *Agric. Sci.*, vol. 01, no. 01, p. Addis Ababa., 1996.

[9] J. S. Nick van de Giesen, Rolf Hut, “van de Giesen - The Trans-African Hydro-Meteorological Observatory TAHMO.pdf,” *Wiley Interdiscip. Rev. Water*, vol. 1, no. August, pp. 341–348, 2014.

[10] G. T. Ayehu, T. Tadesse, B. Gessesse, and T. Dinku, “Validation of new satellite rainfall products over the Upper Blue Nile Basin, Ethiopia,” *Atmos. Meas. Tech.*, vol. 11, no. 4, pp. 1921–1936, 2018, doi: 10.5194/amt-11-1921-2018.

[11] C. Le Coz and N. Van De Giesen, “Comparison of rainfall products over sub-saharan

africa," *J. Hydrometeorol.*, vol. 21, no. 4, pp. 553–596, 2020, doi: 10.1175/JHM-D-18-0256.1.

[12] D. Macharia, K. Fankhauser, J. S. Selker, J. C. Neff, and E. A. Thomas, "Validation and Intercomparison of Satellite-Based Rainfall Products over Africa with TAHMO In Situ Rainfall Observations," *J. Hydrometeorol.*, vol. 23, no. 7, pp. 1131–1154, 2022, doi: 10.1175/JHM-D-21-0161.1.

[13] M. Navid, "Multiple Linear Regressions for Predicting Rainfall for Bangladesh," *Communications*, vol. 6, no. 1, p. 1, 2018, doi: 10.11648/j.com.20180601.11.

[14] Z. Ismail, A. Yahya, and A. Shabri, "Forecasting Gold Prices Using Multiple Linear Regression Method Department of Mathematics , Faculty of Science Department of Basic Education , Faculty of Education," *Am. J. Appl. Sci.*, vol. 6, no. 8, pp. 1509–1514, 2009.

[15] T. S. V. Ramesh and P. P. Mujumdar, "Rainfall forecasting using neural networks," *Stoch. Hydraul. Proc. Symp. Mackay, Aust.*, no. April, pp. 706–713, 2015.

[16] W. Groß, S. Lange, J. Bödecker, and M. Blum, "Predicting time series with space-time convolutional and recurrent neural networks," *ESANN 2017 - Proceedings, 25th Eur. Symp. Artif. Neural Networks, Comput. Intell. Mach. Learn.*, vol. 26–28, no. April, pp. 71–76, 2017.

[17] F. M. Simba and M. Chiturumani, "Comparison of Satellite Data and Ground Based Weather Data in Masvingo, Zimbabwe," *Int. J. Environ. Sci. Nat. Resour.*, vol. 8, no. 4, pp. 102–107, 2018, doi: 10.19080/IJESNR.2018.08.555739.

[18] Y. Mohammed, F. Yimer, M. Tadesse, and K. Tesfaye, "Meteorological drought assessment in north east highlands of Ethiopia International Journal of Climate Change Strategies and Article information :," *Int. J. Clim. Chang. Strateg. Manag.*, no. September, 2017, doi: 10.1108/IJCCSM-12-2016-0179.

[19] P. M. All and J. Terms, "Rainfall variability in the Ethiopian and Eritrean highlands and its links with the Southern Oscillation Index," *J. Biogeogr.*, vol. 22, no. 4, pp. 945–952, 2014.

[20] J. Yan, T. Xu, Y. Yu, and H. Xu, "Rainfall forecast model based on the tabnet model," *Water (Switzerland)*, vol. 13, no. 9, 2021, doi: 10.3390/w13091272.

[21] A. M. E. Toth*, A. Brath, "Comparison of short-term rainfall prediction models for real-time flood forecasting," *J. Hydrol.*, vol. 239, no. 3, pp. 132–147, 2000, doi: 10.2166/nh.2014.178.

[22] P. S. Dutta and H. Tahbilder, "PREDICTION OF RAINFALL USING DATAMINING TECHNIQUE OVER ASSAM," *Indian J. Comput. Sci. Eng.*, vol. 5, no. 2, pp. 85–90, 2014.

[23] M. D. A. Pranatha, N. Pramaita, M. Sudarma, and I. M. O. Widyantara, “Filtering Outlier Data Using Box Whisker Plot Method for Fuzzy Time Series Rainfall Forecasting,” *Proceeding 2018 4th Int. Conf. Wirel. Telemat. ICWT 2018*, vol. 32, no. July, pp. 12–13, 2018, doi: 10.1109/ICWT.2018.8527734.

[24] W. K. Luseno, J. G. McPeak, C. B. Barrett, P. D. Little, and G. Gebru, “Assessing the value of climate forecast information for pastoralists: Evidence from Southern Ethiopia and Northern Kenya,” *World Dev.*, vol. 31, no. 9, pp. 1477–1494, 2003, doi: 10.1016/S0305-750X(03)00113-X.

[25] A. J. Zapata-Sierra, A. Cama-Pinto, F. G. Montoya, A. Alcayde, and F. Manzano-Agugliaro, “Wind missing data arrangement using wavelet based techniques for getting maximum likelihood,” *Energy Convers. Manag.*, vol. 185, no. January, pp. 552–561, 2019, doi: 10.1016/j.enconman.2019.01.109.

[26] S. Manandhar, S. Dev, Y. H. Lee, Y. S. Meng, and S. Winkler, “A Data-Driven Approach for Accurate Rainfall Prediction,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9323–9330, 2019, doi: 10.1109/TGRS.2019.2926110.

[27] R. Saunders, “The use of satellite data in numerical weather prediction,” *Weather Sci.*, vol. 76, no. 3, pp. 95–97, 2021, doi: 10.1002/wea.3913.

[28] D. A. Hughes, “Comparison of satellite rainfall data with observations from gauging station networks,” *J. Hydrol.*, vol. 327, no. 3–4, pp. 399–410, 2006, doi: 10.1016/j.jhydrol.2005.11.041.

[29] X. Ren *et al.*, “Deep Learning-Based Weather Prediction: A Survey,” *Big Data Res.*, vol. 23, pp. 100–178, 2021, doi: 10.1016/j.bdr.2020.100178.

[30] K. Namitha, A. Jayapriya, and G. S. Kumar, “Rainfall prediction using artificial neural network on map-reduce framework,” *ACM Int. Conf. Proceeding Ser.*, vol. 10-13-Augu, pp. 492–495, 2015, doi: 10.1145/2791405.2791468.

[31] A. Y. Barrera-Animas, L. O. Oyedele, M. Bilal, T. D. Akinosho, J. M. D. Delgado, and L. A. Akanbi, “Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting,” *Mach. Learn. with Appl.*, vol. 7, no. November, p. 100204, 2022, doi: 10.1016/j.mlwa.2021.100204.

[32] W. Zaw and T. Naing, “Empirical Statistical Modeling of Rainfall Prediction over Myanmar,” *World Acad Sci Eng Technol*, vol. 2, no. 10, pp. 565–568, 2008, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.193.4595&rep=rep1&type=pdf.

[33] S. A. Fayaz, M. Zaman, and M. A. Butt, “Knowledge Discovery in Geographical Sciences—A Systematic Survey of Various Machine Learning Algorithms for Rainfall Prediction,” *Int. Conf. Innov. Comput. Commun.*, vol. 1388, pp. 593–608, 2022, doi:

10.1007/978-981-16-2597-8_51.

[34] M. Wiston and M. KM, “Weather Forecasting: From the Early Weather Wizards to Modern-day Weather Predictions,” *J. Climatol. Weather Forecast.*, vol. 06, no. 02, 2018, doi: 10.4172/2332-2594.1000229.

[35] P. M. De Lima and E. B. Guedes, “Rainfall prediction for Manaus, Amazonas with artificial neural networks,” *2015 Latin-America Congr. Comput. Intell. LA-CCI*, no. October, pp. 1–6, 2015, doi: 10.1109/LA-CCI.2015.7435934.

[36] P. Bauer, A. Thorpe, and G. Brunet, “The quiet revolution of numerical weather prediction,” *Nature*, vol. 525, no. 7567, pp. 47–55, 2015, doi: 10.1038/nature14956.

[37] Z. Pu and E. Kalnay, *Numerical Weather Prediction Basics: Models, Numerical Methods, and Data Assimilation*. 2018.

[38] T. Roomi, “Lectures on Numerical Weather Prediction Thaer Obaid Roomi,” no. September, 2021, doi: 10.13140/RG.2.2.16732.51849.

[39] C. K. Wikle, “Atmospheric Modeling, Data Assimilation, and Predictability,” *Technometrics*, vol. 47, no. 4, pp. 521–521, 2005, doi: 10.1198/tech.2005.s326.

[40] R. Bornstein and Q. Lin, “Urban heat islands and summertime convective thunderstorms in Atlanta : three case studies,” *Atmos. Environ.*, vol. 34, pp. 507–516, 2000.

[41] A. M. Society and A. Meteorology, “American Meteorological Society Local Circulations Developed in the Vicinity of Both Coastal and Inland Urban Areas : A Numerical Study with a Mesoscale Atmospheric Model Author (s): Yukitaka Ohashi and Hideji Kida Published by : American Meteorological,” *J. Appl. Meteorol.*, vol. 41, no. 1, 2005.

[42] M. J. Best, “Representing urban areas within operational numerical weather prediction models,” *Boundary-Layer Meteorol.*, vol. 114, no. 1, pp. 91–109, 2005, doi: 10.1007/s10546-004-4834-5.

[43] A. Provenzale, “Climate models,” *Rend. Lincei*, vol. 25, no. 1, pp. 49–58, 2014, doi: 10.1007/s12210-013-0268-7.

[44] J. R.-V. and A. Jarvis, “Downscaling Global Circulation Model Outputs: The Delta Method Decision and Policy Analysis Working Paper No. 1,” *Int. Cent. Trop. Agric. CIAT*, no. 1, pp. 29–36, 2010.

[45] N. Kayaba *et al.*, “Dynamical regional downscaling using the JRA-55 reanalysis (DSJRA-55),” *Sci. Online Lett. Atmos.*, vol. 12, no. 1, pp. 1–5, 2016, doi: 10.2151/sola.2016-001.

[46] M. Arshad, X. Ma, J. Yin, W. Ullah, M. Liu, and I. Ullah, “Performance evaluation of

ERA-5, JRA-55, MERRA-2, and CFS-2 reanalysis datasets, over diverse climate regions of Pakistan,” *Weather Clim. Extrem.*, vol. 33, p. 100373, 2021, doi: 10.1016/j.wace.2021.100373.

[47] T. F. Hogan *et al.*, “The navy global environmental model,” *Oceanography*, vol. 27, no. 3, pp. 116–125, 2014, doi: 10.5670/oceanog.2014.73.

[48] IPCC, *CLIMATE CHANGE 2013 Climate Change 2013*. 2013.

[49] C. L. Castro, R. A. Pielke, and G. Leoncini, “Dynamical downscaling: Assessment of value retained and added using the Regional Atmopsheric Modeling System (RAMS),” *J. Geophys. Res. D Atmos.*, vol. 110, no. 5, pp. 1–21, 2005, doi: 10.1029/2004JD004721.

[50] F. Giorgi, “Simulation of Regional Climate Using a Limited Area Model Nested in a General Circulation Model,” in *Journal of climate*, 1990, vol. 26, no. 1, pp. 1–9, [Online]. Available: http://journal.um-surabaya.ac.id/index.php/JKM/article/view/2203.

[51] E. Rogers *et al.*, “The NCEP North American Modeling System: Final Eta model / analysis changes and preliminary experiments using the WRF-NMM,” *Proc. 21st Conf. Weather Anal. Forecast. Conf. Numer. Weather Predict.*, vol. 4, no. August, 2005.

[52] Y. Yamazaki and M. D. M. Orgaz, “Forecasting mesoscale precipitation using the MM5 model with the four-dimensional data assimilation (FDDA) technique,” *Glob. NEST J.*, vol. 7, no. 3, pp. 258–263, 2018, doi: 10.30955/gnj.000344.

[53] N. S. Diffenbaugh and M. Ashfaq, “Intensification of hot extremes in the United States,” *Geophys. Res. Lett.*, vol. 37, no. May, pp. 1–5, 2010, doi: 10.1029/2010GL043888.

[54] V. Mody and S. D. Ray, “Downscaling Climate Models:,” *Encycl. Toxicol. Third Ed.*, vol. 120, no. 1, pp. 456–457, 2014.

[55] R. Srinivasan, C. Santhi, R. D. Harmel, and A. Van Griensven, “SWAT: MODEL USE, CALIBRATION, AND VALIDATION,” *Am. Soc. Agric. Biol. Eng.*, vol. 55, no. 4, pp. 1491–1508, 2012.

[56] V. A. Cooper, V. T. V. Nguyen, and J. A. Nicell, “Calibration of conceptual rainfall-runoff models using global optimisation methods with hydrologic process-based parameter constraints,” *J. Hydrol.*, vol. 334, no. 3–4, pp. 455–466, 2007, doi: 10.1016/j.jhydrol.2006.10.036.

[57] X. Wang and Q. He, “Enhancing generalization capability of SVM classifiers with feature weight adjustment,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3213, pp. 1037–1043, 2004, doi: 10.1007/978-3-540-30132-5_140.

[58] A. J. Holden *et al.*, “Reducing the Dimensionality of Data with Neural Networks,” *Am. Assoc. Adv. Sci.*, vol. 313, no. July, pp. 504–507, 2006.

[59] A. Mohamed, G. E. Dahl, and G. Hinton, “Acoustic Modeling Using Deep Belief Networks,” *IEEE Trans. Audio. Speech. Lang. Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[60] S. K. David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, “X-VECTORS : ROBUST DNN EMBEDDINGS FOR SPEAKER RECOGNITION David Snyder , Daniel Garcia-Romero , Gregory Sell , Daniel Povey , Sanjeev Khudanpur Center for Language and Speech Processing & Human Language Technology Center of Excellence The Johns Hopkins Un,” *IEEE Int. Conf. Acoust. Speech Signal Process. ICASSP, IEEE*, pp. 5329–5333, 2018.

[61] Y. Bengio, “Deep Learning of Representations for Unsupervised and Transfer Learning,” *JMLR Work. Conf. Proc.*, vol. 27, pp. 17–37, 2012.

[62] P. Baldi, P. Sadowski, and D. Whiteson, “physics with deep learning,” *Nat. Commun.*, vol. 2, no. July, pp. 1–9, 2014, doi: 10.1038/ncomms5308.

[63] W. Bhimji, S. A. Farrell, T. Kurth, and E. Racah, “Deep Neural Networks for Physics Analysis on low-level whole-detector data at the LHC Deep Neural Networks for Physics Analysis on low-level whole-detector data at the LHC,” *IOP Conf. Ser. J. Phys. Conf. Ser.*, vol. 1085, pp. 20–34, 2018.

[64] F. Arbabzadah, S. Chmiela, K. R. Mu, and A. Tkatchenko, “Quantum-chemical insights from deep tensor neural networks,” *Nat. Commun.*, vol. 9, no. Jan, pp. 6–13, 2017, doi: 10.1038/ncomms13890.

[65] D. Zheng *et al.*, “Estimated GFR and the Effect of Intensive Blood Pressure Lowering After Acute Intracerebral Hemorrhage,” *Am. J. Kidney Dis.*, vol. 17, no. January, pp. 1–9, 2016, doi: 10.1053/j.ajkd.2016.01.020.

[66] J. N. K. Liu and Y. Hu, “Application of feature-weighted Support Vector regression using grey correlation degree to stock price forecasting,” *Neural Comput. Appl.*, vol. 22, no. SUPPL.1, pp. 143–152, 2013, doi: 10.1007/s00521-012-0969-3.

[67] Z. Karevan and J. A. K. Suykens, “Spatio-temporal Stacked LSTM for Temperature Prediction in Weather Forecasting,” 2018, [Online]. Available: http://arxiv.org/abs/1811.06341.

[68] R. Chen, W. Zhang, and X. Wang, “Machine Learning in Tropical Cyclone Forecast Modeling : A Review,” *J. Atmos.*, vol. 11, no. June 2021, pp. 665–676, 2020, doi: 10.3390/atmos11070676.

[69] M. Hossain, B. Rekabdar, S. J. Louis, and S. Dascalu, “Forecasting the weather of Nevada: A deep learning approach,” *Proc. Int. Jt. Conf. Neural Networks*, vol. 2015-Septe, pp. 2–7, 2015, doi: 10.1109/IJCNN.2015.7280812.

[70] S. Y. Lin, C. C. Chiang, J. Bin Li, Z. S. Hung, and K. M. Chao, “Dynamic fine-tuning stacked auto-encoder neural network for weather forecast,” *Futur. Gener. Comput. Syst.*, vol. 89, pp. 446–454, 2018, doi: 10.1016/j.future.2018.06.052.

[71] M. Qiu *et al.*, “A short-term rainfall prediction model using multi-task convolutional neural networks,” *Proc. - IEEE Int. Conf. Data Mining, ICDM*, vol. 2017-Novem, pp. 395–404, 2017, doi: 10.1109/ICDM.2017.49.

[72] D. M. Cheng, B., & Titterington, “Neural Networks: A Review from a Statistical Perspective.,” *Stat. Sci.*, vol. 9, no. 1, pp. 2–31, 1994, doi: 10.2307/2246134.

[73] L. Shukla, *Designing Your Neural Networks*, no. Sep, 23. 2019. https://towardsdatascience.com/designing-your-neural-networks-a5e4617027ed [Accessed: 29th January 2023].

[74] S. K. Sarvepalli, S. Sarat, and K. Sarvepalli, “Deep Learning in Neural Networks: The science behind an Artificial Brain,” no. October 2015, 2015, doi: 10.13140/RG.2.2.22512.71682.

[75] J. Singh and R. Banerjee, “A study on single and multi-layer perceptron neural network,” *Proc. 3rd Int. Conf. Comput. Methodol. Commun. ICCMC 2019*, no. Iccmc, pp. 35–40, 2019, doi: 10.1109/ICCMC.2019.8819775.

[76] C. M. Bishop and Pattern, *Pattern Recognition and Machine Learning*. 2006. https://link.springer.com/book/9780387310732 [Accessed: 26th January 2023].

[77] H. Ramchoun, M. Amine, J. Idrissi, Y. Ghanou, and M. Ettaouil, “Multilayer Perceptron : Architecture Optimization and Training,” *Int. J. Interact. Multimed. Artif. Intell.*, vol. 4, no. 1, pp. 26–30, 2016, doi: 10.9781/ijimai.2016.415.

[78] T. Wu and J. Chen, “Machine Learning-based Short-term Rainfall Prediction from Sky Data,” *ACM Trans. Knowl. Discov. Data*, vol. 16, no. 6, 2022.

[79] Wikipedia The free encyclopaedia. *Multilayer perceptron*. 2018. https://en.wikipedia.org/wiki/Multilayer_perceptron [Accessed: 6th February 2023].

[80] Y. Zhao, B. Deng, and Z. Wang, “Analysis and study of perceptron to solve XOR problem,” *Proc. - 2nd Int. Work. Auton. Decentralized Syst. IWADS 2002*, no. Nov, pp. 168–173, 2002, doi: 10.1109/IWADS.2002.1194667.

[81] W. Schiffmann, M. Joost, and R. Werner, *Optimization of the Backpropagation Algorithm for Training Multilayer Perceptrons*. 1994.

[82] J. Wu, “Introduction to Convolutional Neural Networks,” *Nov. Softw. Technol.*, no. May 1, pp. 1–31, 2017.

[83] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, “Fundamental Concepts of Convolutional Neural Network,” *Neural Networks*, vol. 02, no. June, 2020, doi:

10.1007/978-3-030-32644-9.

[84] F. Sultana, A. Sufian, and P. Dutta, “Advancements in image classification using convolutional neural network,” *Proc. - 2018 4th IEEE Int. Conf. Res. Comput. Intell. Commun. Networks, ICRCICN 2018*, pp. 122–129, 2018, doi: 10.1109/ICRCICN.2018.8718718.

[85] S. Pouyanfar *et al.*, “A survey on deep learning: Algorithms, techniques, and applications,” *ACM Comput. Surv.*, vol. 51, no. 5, 2018, doi: 10.1145/3234150.

[86] I. Kaastraa and M. Boydb, “Designing a neural network for forecasting financial time series 29,” *Neurocomputing*, vol. 10, no. 3, pp. 215–236, 1996, [Online]. Available: https://doi.org/10.1016/0925-2312(95)00039-9.

[87] T. Edwards and D. Tansley, “Traffic trends analysis using neural networks,” *... Neural Networks to ...*, no. June 2000, 1997, [Online]. Available: https://uhra.herts.ac.uk/dspace/handle/2299/7171.

[88] I. Roesch and T. Günther, “Visualization of Neural Network Predictions for Weather Forecasting,” *Comput. Graph. Forum*, vol. 38, no. 1, pp. 209–220, 2019, doi: 10.1111/cgf.13453.

[89] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, “Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, no. March, pp. 5457–5466, 2018, doi: 10.1109/CVPR.2018.00572.

[90] C. W. Ruiz, J. Perapoch, F. Castillo, S. Salcedo, and E. Gratacós, “Learning Long-Term Dependencies with Gradient Descent is Difficult,” *Pediatr. Catalana*, vol. 66, no. 2, pp. 53–61, 2006.

[91] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

[92] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, “TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS 1 LSTM: A Search Space Odyssey,” *arXiv:1503.04069*, pp. 1–11, 2015, [Online]. Available: https://arxiv.org/pdf/1503.04069.pdf.

[93] Hochreiter and Schmidhuber, “Long Short-Term Memory,” *IEEE Trans. NEURAL NETWORKS Learn. Syst.*, vol. 50, no. 6, pp. 2199–2207, 2018, doi: 10.17582/journal.pjz/2018.50.6.2199.2207.

[94] N. Meade and M. R. Maier, “Evidence of Long Memory in Short-term Interest Rates,” *J. Forecast.*, vol. 568, no. 22, pp. 553–568, 2003.

[95] J. S. Sepp Hochreiter, “Long short term memory,” *Neural Comput.*, vol. 9, no. 8, pp.

1–32, 1997.

[96] P. J. WERBOS, “Generalization of Backpropagation with Application to a Recurrent Gas Market Model,” *Neural Networks*, vol. 1, no. May, pp. 339–356, 1988.

[97] F. A. Gers and F. Cummins, “Learning to Forget : Continual Prediction with LSTM,” *Neural Comput.*, vol. 2471, no. 12, pp. 2451–2471, 2000.

[98] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, “Translating videos to natural language using deep recurrent neural networks,” *NAACL HLT 2015 - 2015 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Conf.*, pp. 1494–1504, 2015, doi: 10.3115/v1/n15-1173.

[99] J. Donahue *et al.*, “Long-Term Recurrent Convolutional Networks for Visual Recognition and Description,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 677–691, 2017, doi: 10.1109/TPAMI.2016.2599174.

[100] Y. Verma, “Complete Guide To Bidirectional LSTM (With Python Codes).,” *Anal. indiamag*, no. July 17, pp. 1–19, 2021, [Online]. Available: https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/#:~:text=In bidirectional%2C our input flows in two directions%2C,to preserve the future and the past information.

[101] S. Aswin, P. Geetha, and R. Vinayakumar, “Deep Learning Models for the Prediction of Rainfall,” *Proc. 2018 IEEE Int. Conf. Commun. Signal Process. ICCSP 2018*, no. 3–5, pp. 657–661, 2018, doi: 10.1109/ICCSP.2018.8523829.

[102] X. Zhang, S. N. Mohanty, A. K. Parida, S. K. Pani, B. Dong, and X. Cheng, “Annual and Non-Monsoon Rainfall Prediction Modelling Using SVR-MLP: An Empirical Study from Odisha,” *IEEE Access*, vol. 8, no. February, pp. 30223–30233, 2020, doi: 10.1109/ACCESS.2020.2972435.

[103] D. Endalie, G. Haile, and W. Taye, “Deep learning model for daily rainfall prediction: case study of Jimma, Ethiopia,” *Water Supply*, vol. 22, no. 3, pp. 3448–3461, 2022, doi: 10.2166/WS.2021.391.

[104] M. Chhetri, S. Kumar, P. P. Roy, and B. G. Kim, “Deep BLSTM-GRU model for monthly rainfall prediction: A case study of Simtokha, Bhutan,” *Remote Sens.*, vol. 12, no. 19, pp. 1–13, 2020, doi: 10.3390/rs12193174.

[105] I. Salehin, I. M. Talha, M. Mehedi Hasan, S. T. Dip, M. Saifuzzaman, and N. N. Moon, “An Artificial Intelligence Based Rainfall Prediction Using LSTM and Neural Network,” *Proc. 2020 IEEE Int. Women Eng. Conf. Electr. Comput. Eng. WIECON-ECE 2020*, vol. 978, no. 1, pp. 5–8, 2020, doi: 10.1109/WIECON-ECE52138.2020.9398022.

[106] N. S. Sani, A. H. A. Rahman, A. Adam, I. Shlash, and M. Aliff, “Ensemble Learning for Rainfall Prediction,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 11, pp. 153–162,

2020, doi: 10.14569/IJACSA.2020.0111120.

[107] C. M. Liyew and H. A. Melese, “Machine learning techniques to predict daily rainfall amount,” *J. Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00545-4.

[108] Y. O. Ouma, R. Cheruyot, and A. N. Wachera, “Rainfall and runoff time-series trend analysis using LSTM recurrent neural network and wavelet neural network with satellite-based meteorological data: case study of Nzoia hydrologic basin,” *Complex Intell. Syst.*, vol. 8, no. 1, pp. 213–236, 2022, doi: 10.1007/s40747-021-00365-2.

[109] Vikrant Singh, “Study of Various Rainfall Estimation Prediction Techniques using Data Mining,” *Int. J. Eng. Res.*, vol. V9, no. 07, pp. 1227–1229, 2020, doi: 10.17577/ijertv9is070464.

[110] A. Alturki, G. G. Gable, and W. Bandara, “A design science research roadmap,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6629 LNCS, no. 4, pp. 107–123, 2011, doi: 10.1007/978-3-642-20633-7_8.

[111] A. Ayanso, K. Lertwachara, and F. Vachon, “Design and behavioral science research in premier IS journals: Evidence from database management research,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6629 LNCS, pp. 138–152, 2011, doi: 10.1007/978-3-642-20633-7_10.

[112] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design Science in Information systems research,” *Manag. Inf. Syst. Res. Cent.*, vol. 28, no. 1, pp. 75–105, 2004.

[113] O. A. E. Joseoh g. Walls, George R. Widmeyer, “Building an information system design theory for vigilant EIS,” *Inf. Sci. Res.*, vol. 3, no. 1, pp. 36–59, 1992.

[114] J. F. Nunamaker and M. Chen, “Systems development in information systems research,” *Proc. Hawaii Int. Conf. Syst. Sci.*, vol. 3, pp. 631–640, 1990, doi: 10.1016/b978-1-876938-42-0.50016-2.

[115] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design Science in Information Research 1,” *Des. Sci. IS Res. MIS Q.*, vol. 28, no. 1, pp. 75–105, 2004.

[116] J. F. N. Jr, M. Chen, T. D. M. Purdin, J. A. Y. F. Nunamaker, and M. Chen, “Systems Development in Information Systems Systems Development in Information Systems Research,” *J. Manag. Inf. Syst. ISSN*, vol. 1222, no. May, pp. 89–106, 2016, doi: 10.1080/07421222.1990.11517898.

[117] M. K. Rossi, M., and Sein, “Design science research and the core of information systems,” *26th Inf. Syst. Res. Semin. Scand. IRIS Assoc. Haikko Finl.*, vol. 7286 LNCS, pp. 309–327, 2003, doi: 10.1007/978-3-642-29863-9_23.

[118] J. B. Ken Peffers, Tuure Tuunanen, Charles E. Gengler, Matti Rossi, Wendy Hui, “THE DESIGN SCIENCE RESEARCH PROCESS: AMODEL FOR PRODUCING

AND PRESENTING INFORMATION SYSTEMS RESEARCH," *Comuter Sci. Res.*, vol. 24, no. 57–77, p. 24, 2006, [Online]. Available: h.

[119] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, 2007, doi: 10.2753/MIS0742-1222240302.

[120] A. P. Pandian, "Performance Evaluation and Comparison using Deep Learning Techniques in Sentiment Analysis," *J. Soft Comput. Paradig.*, vol. 3, no. 2, pp. 123–134, 2021, doi: 10.36548/jscp.2021.2.006.

[121] M. Wieland and M. Pittore, "Performance evaluation of machine learning algorithms for urban pattern recognition from multi-spectral satellite images," *Remote Sens.*, vol. 6, no. 4, pp. 2912–2939, 2014, doi: 10.3390/rs6042912.

[122] M. Rout, J. K. Rout, and H. Das, *Correction to: Nature Inspired Computing for Data Science*, no. January. 2020.

[123] J. S. Clark, *Model Assessment and Selection*. 2020. The Elements of Statistical Learning (pp.219-259)

[124] V. R. Joseph and A. Vakayil, "SPlit: An Optimal Method for Data Splitting," *Technometrics*, vol. 64, no. 2, pp. 166–176, 2022, doi: 10.1080/00401706.2021.1921037.

[125] D. Z. Abidin, S. Nurmaini, R. Firsandava Malik, Erwin, E. Rasywir, and Y. Pratama, "RSSI Data Preparation for Machine Learning," *Proc. - 2nd Int. Conf. Informatics, Multimedia, Cyber, Inf. Syst. ICIMCIS 2020*, pp. 284–289, 2020, doi: 10.1109/ICIMCIS51567.2020.9354273.

[126] N. S. Sani, I. I. S. Shamsuddin, S. Sahran, A. H. A. Rahman, and E. N. Muzaffar, "Redefining selection of features and classification algorithms for room occupancy detection," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 8, no. 4–2, pp. 1486–1493, 2018, doi: 10.18517/ijaseit.8.4-2.6826.

[127] N. S. Sani, M. A. Rahman, A. A. Bakar, S. Sahran, and H. M. Sarim, "Machine learning approach for Bottom 40 Percent Households (B40) poverty classification," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 8, no. 4–2, pp. 1698–1705, 2018, doi: 10.18517/ijaseit.8.4-2.6829.

[128] I. Bae and U. Ji, "Outlier detection and smoothing process for water level data measured by ultrasonic sensor in stream flows," *Water (Switzerland)*, vol. 11, no. 5, 2019, doi: 10.3390/w11050951.

[129] K. S. Kannan, K. Manoj, and S. Arumugam, "Labeling Methods for Identifying Outliers," *Int. J. Stat. Syst.*, no. October, 2015.

[130] M. M. Mukaka, "Statistics corner: A guide to appropriate use of correlation coefficient

in medical research," *Malawi Med. J.*, vol. 24, no. 3, pp. 69–71, 2012.

[131] N. Method, R. Variability, and D. To, "Normalizing Genetic Reporter Assays : Approaches and Considerations for Increasing Consistency and Statistical Significance," *Cell Notes Issue*, no. 17, pp. 9–12, 2007.

[132] D. Kim, "Normalization methods for input and output vectors in Backpropagation neural networks," *Int. J. Comput. Math.*, vol. 71, no. 1–2, pp. 161–171, 2007, doi: 10.1080/00207169908804800.

[133] L. Al Shalabi, Z. Shaaban, and B. Kasasbeh, "Data Mining: A Preprocessing Engine," *J. Comput. Sci.*, vol. 2, no. 9, pp. 735–739, 2006, doi: 10.3844/jcssp.2006.735.739.

[134] N. Mishra, H. K. Soni, S. Sharma, and A. K. Upadhyay, "Development and analysis of Artificial Neural Network models for rainfall prediction by using time-series data," *Int. J. Intell. Syst. Appl.*, vol. 10, no. 1, pp. 16–23, 2018, doi: 10.5815/ijisa.2018.01.03.

[135] M. Fathi, M. Haghi, K. Seyed, M. Jameii, and E. Mahdipour, "Big Data Analytics in Weather Forecasting : A Systematic Review," *Arch. Comput. Methods Eng.*, vol. 29, no. 2, pp. 1247–1275, 2022, doi: 10.1007/s11831-021-09616-4.

[136] L. Latifoğlu, "A novel combined model for prediction of daily precipitation data using instantaneous frequency feature and bidirectional long short time memory networks," *Environ. Sci. Pollut. Res.*, vol. 29, no. 28, pp. 42899–42912, 2022, doi: 10.1007/s11356-022-18874-z.

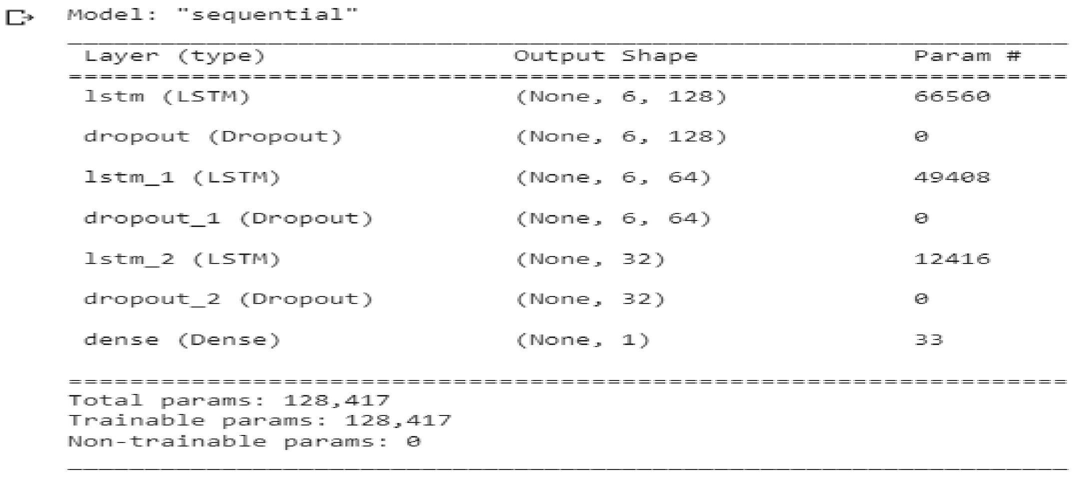
[137] X. Zhang, H. Chen, Y. Wen, J. Shi, and Y. Xiao, "A new rainfall prediction model based on ICEEMDAN-WSD-BiLSTM and ESN," *Environ. Sci. Pollut. Res.*, pp. 53381–53396, 2023, doi: 10.1007/s11356-023-25906-9.

[138] J. J. Helmus and S. M. Collis, "The Python ARM Radar Toolkit (Py-ART), a Library for Working with Weather Radar Data in the Python Programming Language," *J. Open Res. Softw.*, vol. 4, no. 1, p. 25, 2016, doi: 10.5334/jors.119.

[139] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *White Pap.*, vol. 16, no. Mar, 2016, [Online]. Available: http://arxiv.org/abs/1603.04467.

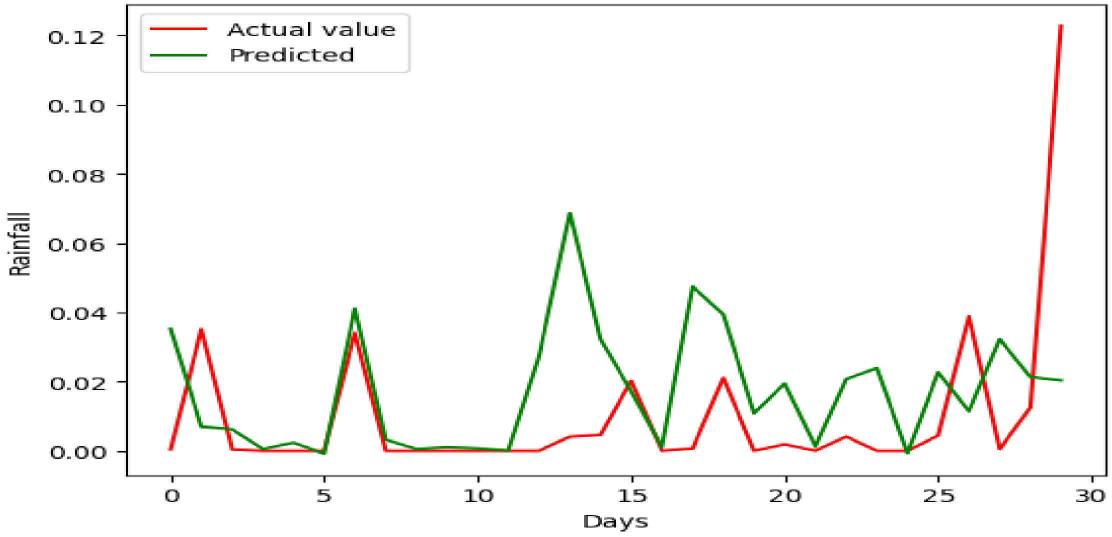
[140] and A. M. Ferdin Joe John Joseph, Sarayut Nonsiri, "Keras and TensorFlow: A Hands-On Experience," vol. 22, no. August, 2021, doi: 10.1007/978-3-030-66519-7.

Appendix: I Experimental Results of Separate datasets



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 6, 128)            66560

 dropout (Dropout)           (None, 6, 128)            0

 lstm_1 (LSTM)               (None, 6, 64)             49408

 dropout_1 (Dropout)         (None, 6, 64)             0

 lstm_2 (LSTM)               (None, 32)                12416

 dropout_2 (Dropout)         (None, 32)                0

 dense (Dense)               (None, 1)                 33

=================================================================
Total params: 128,417
Trainable params: 128,417
Non-trainable params: 0
_________________________________________________________________
```

- **The Actual and Predicted values for CNN on Station data.**



- **The Actual and Predicted values for CNN on satellite data.**

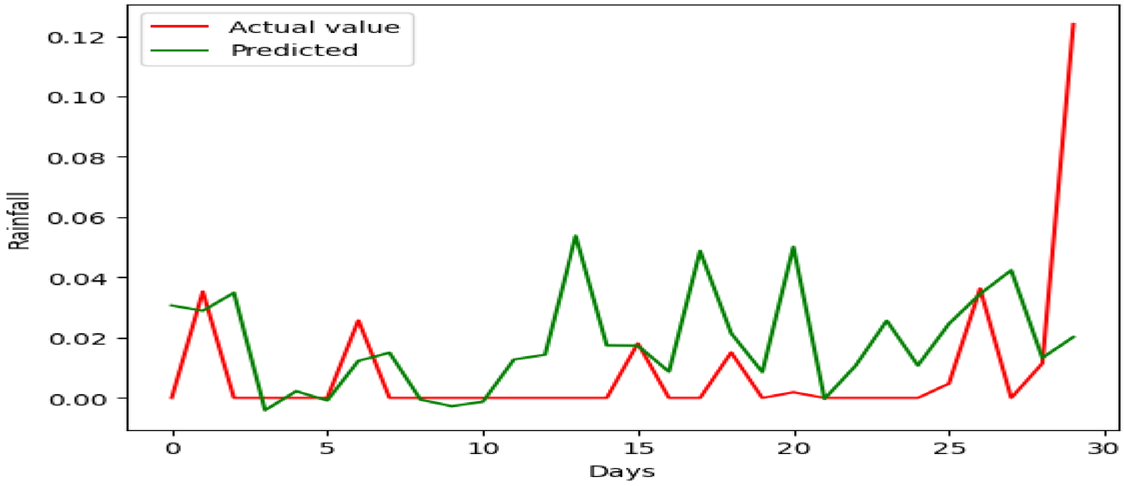
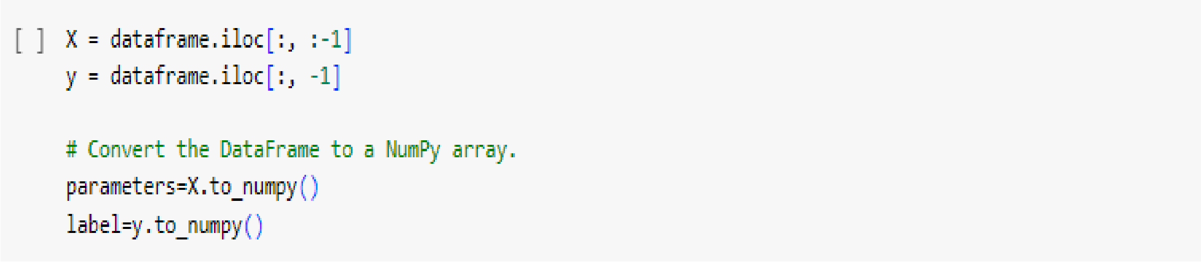


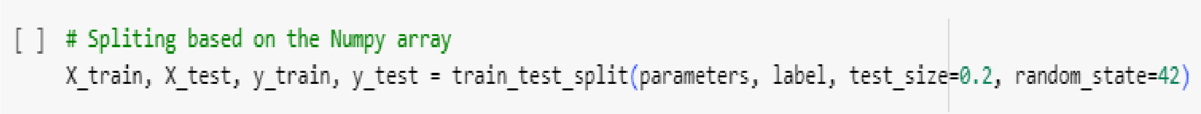
Table | Shows the Evaluation results for separate datasets

	Station			Satellite		
	RMSE	MSE	MAE	RMSE	MSE	MAE
MLP	0.0572	0.0033	0.0279	0.0625	0.0039	0.03003
CNN	0.058	0.003382	0.0243	0.060	0.0037	0.0274
LSTM	0.0582	0.00338	0.0246	0.065	0.0042	0.0359
BiLSTM	0.057	0.0032	0.0278	0.061	0.00372	0.0304

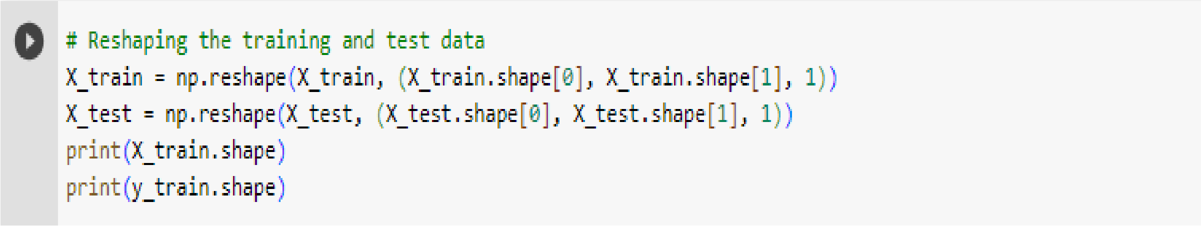
Appendix II: Bidirectional Long Short Term Memory (BiLSTM) Model Python Code



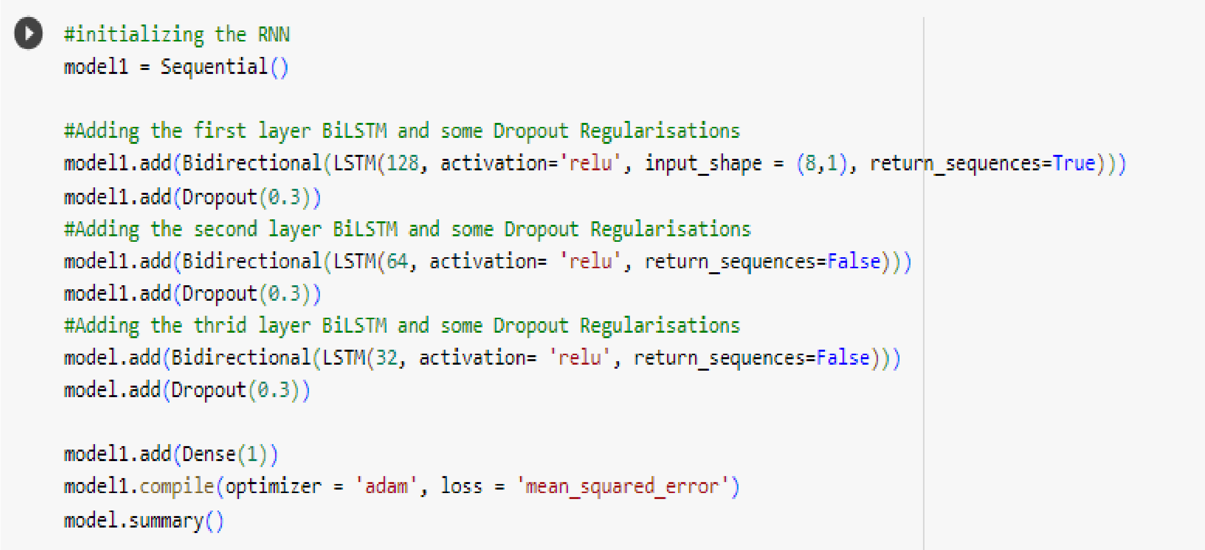
```
[ ] X = dataframe.iloc[:, :-1]
    y = dataframe.iloc[:, -1]

    # Convert the DataFrame to a NumPy array.
    parameters=X.to_numpy()
    label=y.to_numpy()
```



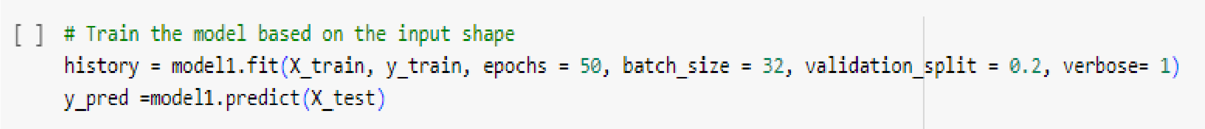
```
[ ] # Spliting based on the Numpy array
    X_train, X_test, y_train, y_test = train_test_split(parameters, label, test_size=0.2, random_state=42)
```



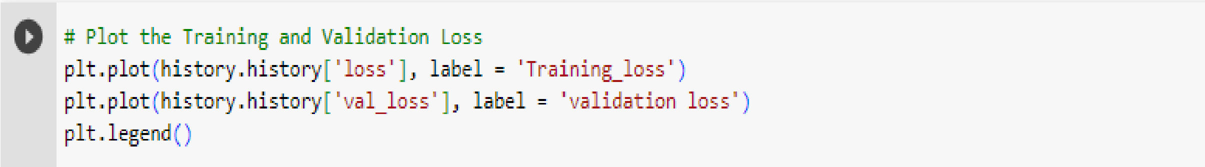
```
# Reshaping the training and test data
X_train = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
print(X_train.shape)
print(y_train.shape)
```



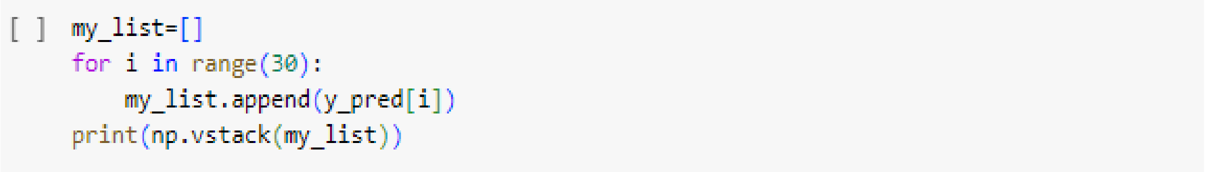
```
#initializing the RNN
model1 = Sequential()

#Adding the first layer BiLSTM and some Dropout Regularisations
model1.add(Bidirectional(LSTM(128, activation='relu', input_shape = (8,1), return_sequences=True)))
model1.add(Dropout(0.3))
#Adding the second layer BiLSTM and some Dropout Regularisations
model1.add(Bidirectional(LSTM(64, activation= 'relu', return_sequences=False)))
model1.add(Dropout(0.3))
#Adding the thrid layer BiLSTM and some Dropout Regularisations
model.add(Bidirectional(LSTM(32, activation= 'relu', return_sequences=False)))
model.add(Dropout(0.3))

model1.add(Dense(1))
model1.compile(optimizer = 'adam', loss = 'mean_squared_error')
model.summary()
```



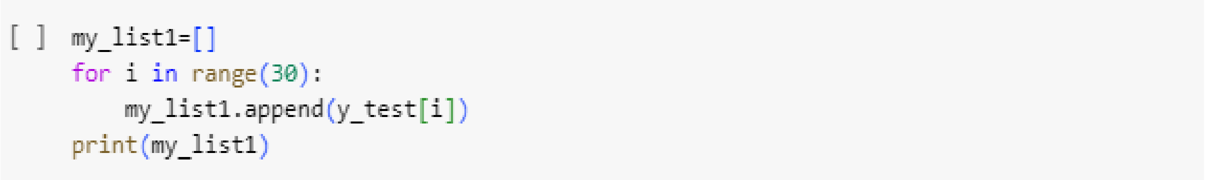
```
# Train the model based on the input shape
history = model1.fit(X_train, y_train, epochs = 50, batch_size = 32, validation_split = 0.2, verbose= 1)
y_pred =model1.predict(X_test)
```



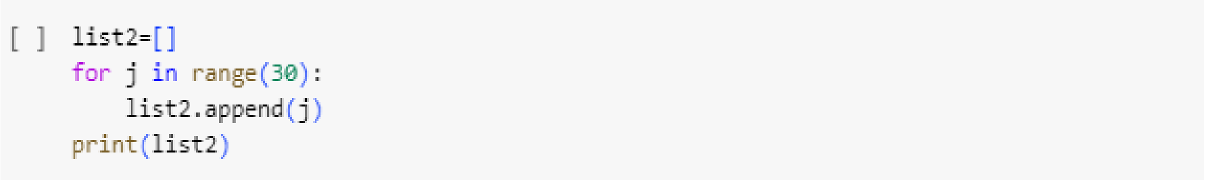
```
# Plot the Training and Validation Loss
plt.plot(history.history['loss'], label = 'Training_loss')
plt.plot(history.history['val_loss'], label = 'validation loss')
plt.legend()
```



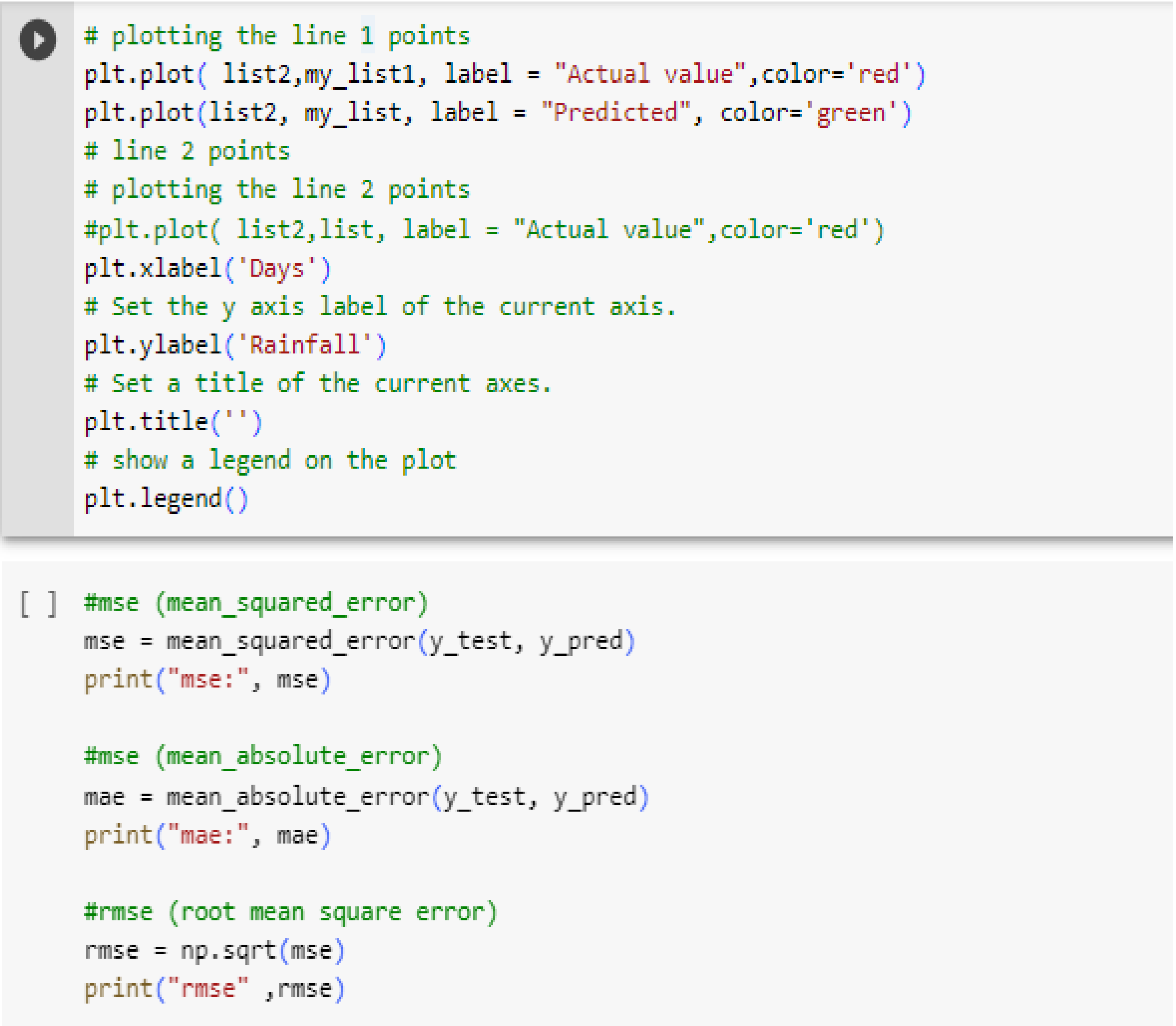
```
my_list=[]
for i in range(30):
    my_list.append(y_pred[i])
print(np.vstack(my_list))
```



```
my_list1=[]
for i in range(30):
    my_list1.append(y_test[i])
print(my_list1)
```



```
list2=[]
for j in range(30):
    list2.append(j)
print(list2)
```



```
# plotting the line 1 points
plt.plot( list2,my_list1, label = "Actual value",color='red')
plt.plot(list2, my_list, label = "Predicted", color='green')
# line 2 points
# plotting the line 2 points
#plt.plot( list2,list, label = "Actual value",color='red')
plt.xlabel('Days')
# Set the y axis label of the current axis.
plt.ylabel('Rainfall')
# Set a title of the current axes.
plt.title('')
# show a legend on the plot
plt.legend()
```

```
#mse (mean_squared_error)
mse = mean_squared_error(y_test, y_pred)
print("mse:", mse)

#mse (mean_absolute_error)
mae = mean_absolute_error(y_test, y_pred)
print("mae:", mae)

#rmse (root mean square error)
rmse = np.sqrt(mse)
print("rmse" ,rmse)
```