



DEBRE MARKOS UNIVERSITY

INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

**A DEEP LEARNING BASED FORECASTING MONTHLY
RAINFALL FOR EAST GOJJAM ZONE**

MSc. Thesis

BY

Kerebih Belay

Program: Information Technology

Advisor Name: Direselign Addis (Ph.D)

MAY 2024 G.C

DEBRE MARKOS, ETHIOPIA

**A Deep Learning Based Forecasting Monthly Rainfall for East Gojjam
Zone**

**A Thesis Submitted to Institute of Technology, Graduate Studies
Program**

Debre Markos University

**In Partial Fulfillment of the Requirements for the Degree of Master of
Science in Information Technology**

Kerebih Belay

Advisor: Direselign Addis (Ph.D)

MAY 2024 G.C

DEBRE MARKOS, ETHIOPIA

DECLARATION

This is to certify that the thesis prepared by Kerebih Belay Hunegnaw, titled: “**A Deep Learning Based Forecasting Monthly Rainfall for East Gojjam Zone**” Submitted for partial fulfillment of the requirements of the degree Master of Science in Information Technology, Debre Markos Institute of Technology is recorded my original work and the thesis work has not been presented for the award of any higher degree in any other university. I have fully cited and referenced all material and results that are not original to this work.

Name of Student: _____

Signature: _____

Date of Submission: _____

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

Advisor Name: _____

Advisor Signature: _____

THESIS APPROVAL SHEET

DEBRE MARKOS UNIVERSITY INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTING

A Bidirectional Long Short-Term Memory (BLSTM) Based Forecasting of Monthly Rainfall in East Gojjam Zone

Submitted by:

Name of Student	Signature	Date
<u>KEREBIH BELAY HUNEGNAW</u>	_____	_____

Approved by:

Name of Major Advisor	Signature	Date
1. _____	_____	_____

Name of Co-Advisor	Signature	Date
2. _____	_____	_____

Name of External examiner	Signature	Date
3. _____	_____	_____

Name of internal examiner	Signature	Date
4. _____	_____	_____

Name of Chairman, DGC	Signature	Date
5. _____	_____	_____

Name of School Dean, SGC	Signature	Date
6. _____	_____	_____

ACKNOWLEDGEMENTS

First, I would like to thank the Almighty God and His mother, St. Mary, for their blessings and for giving me the strength, perseverance, and determination. Without their assistance, neither my thesis work nor my efforts would have been successful.

I would like to express my sincere gratitude to my thesis advisor **Direselign Addis** (Ph.D), I am deeply thankful to him for giving his insights and constructive comments, continuous guidance, ideas, and suggestions throughout the end of this research. I am also thankful to my classmate friends for always being there in case of any difficulty for giving me viable ideas and guidance in my research from beginning to end.

I would like to thank **Ghion Technology College** in Debre Markos branch for give a chance to follow master's program.

Finally, I would like to give my deepest gratitude to my lovely father, **Belay Hunegnaw**, and my mother **Enieayehu Kebede** for their unconditional love, assistance, support, and encouragement starting from my childhood up to the current level of work. I also thank all the rest family and best friends in whole my life. All of this continuous support is the reason for my spirit, which refreshed and strengthened me during the accomplishment of this intensive research work.

Name: Kerebih Belay

Signature: _____

Date: _____

ABSTRACT

Accurate rainfall prediction is a critical task with significant implications for agriculture, water resource management, and disaster preparedness. In regions like the East Gojjam zone of Ethiopia, where agriculture is a primary livelihood, reliable rainfall forecasts are essential for planning and decision-making. The dynamic and unpredictable nature of rainfall patterns poses challenges for traditional forecasting methods, necessitating the development of more sophisticated approaches to improve prediction accuracy.

In this study, we employ advanced deep learning techniques to forecast monthly rainfall using a 35-year meteorological dataset from the National Meteorology Service Agency (NMSA) of Bahir Dar. The data preprocessing steps included handling missing values through mean imputation and normalizing the dataset within a range of [0, 1]. We divided the dataset into training and testing subsets and utilized Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BLSTM), and Conventional Neural Network over Long Short-Term Memory (CNN-LSTM) networks for the predictions. The models were trained and optimized using the Adam optimization algorithm, with key parameters such as the number of epochs, batch size, and learning rate being fine-tuned to enhance performance.

The results demonstrated that the BLSTM model achieved the highest accuracy with a performance rate of 98.8% and a Mean Squared Error (MSE) of 0.012. These findings indicate that deep learning models, particularly BLSTM, can significantly improve the accuracy of rainfall predictions in the East Gojjam zone. The study's outcomes have important implications for agricultural planning and water resource management, providing a more reliable tool for forecasting in regions heavily dependent on rainfall.

Keywords: Rainfall prediction, LSTM, CNN-LSTM, BLSTM.

Table of Contents	Page
ACKNOWLEDGEMENTS	iv
ABSTRACT.....	v
List of Figure.....	ix
List of Table	x
List of Abbreviation	xi
CHAPTER ONE	1
1.Introduction	1
1.1.Background of the study	1
1.2.Statement of the Problem	2
1.3.Research Questions	3
1.4.Objective	4
1.4.1.General Objective	4
1.4.2.Specific Objective.....	4
1.5.Scope of the study	5
1.6.Limitation of the study	5
1.7.Significance of the study	5
1.8.Contribution of the study	6
1.9.Organization of the Thesis	7
CHAPTER TWO	8
2.LITERATURE REVIEW.....	8
2.1.Introduction	8
2.2.Ethiopian weather characteristics.....	10
2.3.Ethiopian Seasonal Climate Characteristics.....	10
2.4.Rainfall in Ethiopia	10
2.5.Machine learning techniques For Rainfall Prediction.....	11
2.5.1.Artificial Neural Network Based Weather Forecasting.....	11
2.6.Deep learning	13
2.6.1.Long short Term Memory (LSTM)	14
2.6.2.Bidirectional Long short Term Memory (Bi-LSTM)	14
2.6.3.Convolutional Neural Network Long-Sort Memory (CNN-LSTM)	15
2.7.Related work Summary.....	16
CHAPTER THREE.....	17
3.RESEARCH METHODOLOGY	17

3.1.Introduction	17
3.2.Study Area.....	17
3.2.1.Overview of Ethiopian National Metrology Agency	17
3.3.Data source.....	18
3.4.Dataset Set Preparation	19
3.5.Data preprocessing	20
3.5.1.Data Reduction	20
3.5.2.Data cleaning	20
3.5.3.Handling missing data	20
3.5.4.Data Normalization.....	20
3.6.Model Development.....	21
3.6.1.Proposed model Architecture.....	21
3.6.2.Long Sort Term Memory (LSTM).....	21
3.6.3.Convolutional Neural Network with Long-Sort-Memory (CNN-LSTM).....	23
3.6.3.1.Convolution Neural networks (CNN).....	23
3.6.4.Bidirectional Long Short Term Memory (Bi-LSTM).....	24
3.7.Model Implementation Tools.....	25
3.8.Experimental Setup	27
3.9.Model Evaluation Techniques.....	27
CHAPTER FOUR.....	29
4.EXPERIMENT AND RESULT DISCUSSION	29
4.1.Introduction	29
4.2.Dataset Set Preparation	29
4.3.Hyper Parameters in the Model.....	29
4.4.Experimental Result	33
4.4.1.Experimental Result of Bi-LSTM model.....	33
4.4.2.Techniques of overfitting avoidance.....	34
4.4.3.Experimental Result of LSTM and CNN-LSTM.....	36
4.4.4.Experimental Result LSTM model	36
4.4.5.Experimental Result CNN-LSTM model	38
4.4.6.Over all Comparison of Different Algorithms.....	39
CHAPTER FIVE.....	42
5.CONCLUSION AND RECOMMENDATION.....	42

5.1.Conclusion	42
5.2.Recommendation.....	43
References	44

List of Figure

Figure 3. 2: Data set Features.....	19
Figure 3. 4: Proposed model architecture	21
Figure 3. 5: LSTM architecture.....	22
Figure 3. 6: CNN architecture.....	24
Figure 3. 7: BLSTM architecture	25
Figure 4. 1: Validation and training for the BLSTM model and number of iterations	35
Figure 4. 2: Evaluation of the BLSTM model with testing dataset	36
Figure 4. 3: Validation and training for the LSTM model and number of iterations.....	37
Figure 4. 4: Evaluation of the proposed LSTM model with testing dataset.	37
Figure 4. 5: Validation and training loss CNN-LSTM model and number of iterations. .	38
Figure 4. 6: Evaluation of the proposed CNN-LSTM model with testing dataset.....	39

List of Table

Table 2. 1: Related work	16
Table 3. 1: The feature and their corresponding units	19
Table 4. 1: Hyper-Parameters and Value in the Proposed Model.....	32
Table 4. 2: BLSTM model evaluation matric	35
Table 4. 3: Evaluation of the performance of the proposed LSTM model	37
Table 4. 4: Evaluation of the performance of the proposed CNN-LSTM model	38
Table 4. 5: Evaluation of the performance of the proposed CNN-LSTM model	39

List of Abbreviation

ANFIS	adaptive neuro-fuzzy inference system
ARIMA	Regressive Integrated Moving Average
Bi-LSTM	Bidirectional Long Short Term Memory
CNN	Conventional Neural Network
DNN	Deep Neural Network
ELM	Extreme Learning Machine
GPU	Graphic Process Unit
IITM	Indian Institute of Technology Madras
LSTM	Long Short Term Memory
MAE	mean absolute error
MLP	Multi- Layer Perceptron
MSE.	Mean Squared Error
NMA	National Meteorological Agency
NMSA	National Meteorological Services Agency
NWP	Numerical Weather Prediction
RMSE	Root mean squared error
RNN	Recurrent neural networks
TPU	Tensor Process Unit

CHAPTER ONE

1. Introduction

1.1. Background of the study

Meteorological observation has started in nineteenth century mainly by European missionaries with very limited stations. The importance of meteorological information's becomes realized by different sectors and thus, the National Meteorological Services Agency (NMSA) was established by Government Proclamation Number 201/1980, now renamed as; National Meteorological Agency (NMA) which was accountable for Water Resource commission during the establishment (Currently, NMA is reporting to the Ministry of Water Irrigation and Energy)[1].

Ethiopia has had almost double-digit economic growth over the past ten years, making it one of the developing countries in Africa. But the failed rain had disastrous effects on the 96 million people's access to food [2].

“The Belg rain was much worse than the National Meteorology Agency prediction at the beginning of the year. As a result, food insecurity and malnutrition increased”. Thus, supporting meteorology experts by developing an effective model can provide timely and accurate rainfall prediction has tremendous advantage[3].

Forecasting the amount of rain has an impact on various stakeholders, from farmers in the agricultural sector to travelers making travel plans. Furthermore, early warning systems for flooding can make use of rainfall predictions that are correct[4].

Rainfall or any other type of climate prediction is quite difficult, even if it is very useful. One model designed for one place may not work as well in another since rainfall depends on a variety of dependent elements, which differ from one geographic area to another[5].

In this study, we focus on forecasting rainfall across East Gojjam. Notably, there has been no prior research on employing Bi-LSTM, CNN-LSTM, LSTM, or Deep Neural Network (DNN) techniques for this purpose in any East Gojjam zone. It is important to recognize the regional variations in weather characteristics, which result in different parameters being recorded by weather stations. Therefore, models designed for one region may not be applicable to another[5].

For our study, we utilized data from an automatic weather station located in Bahir Dar, the capital city of the Amhara region. The main objective of this research is to explore the efficacy of deep learning techniques for monthly rainfall forecasting, with potential applications for Ethiopian planners, policymakers, agriculture industry professionals, and meteorology specialists.

1.2. Statement of the Problem

This research addresses the critical issue of rainfall forecasting and the importance of early warning systems in an agricultural nation like Ethiopia. The Ethiopian Meteorology Agency regularly collects meteorological data in our nation.

Traditional machine learning methods face several challenges when applied to rainfall forecasting such as Spatial Variability, Non-linearity and Seasonality, Handling imbalanced data, including rainfall prediction using Random Forests and others. so Rainfall patterns often exhibit non-linear relationships with meteorological variables and Rainfall patterns can vary significantly across different geographical locations due to local topography, land cover, and atmospheric conditions long-term dependencies.

Those models often treat data points independently (cross-sectional), ignoring sequential dependencies. This can result in suboptimal predictions, especially in time-series forecasting where the order and timing of events are crucial.

The researchers carried out a study on artificial neural network weather forecasting, using a single algorithm, to investigated the usage of artificial neural networks for temperature prediction one day in advance using a three-year data set [6].

Another researcher conducted on Application of data mining for weather forecasting by using 15 year's data set to predict rainfall after one day ahead. He used data mining methodology and three single data mining algorithms[4].

However, the weather differs from nation to nation depending on how close they are to the equator. Therefore, the developed model from other nations is not implemented to our nation without verification. Furthermore, those earlier investigations were carried out in our nation employing machine learning algorithms. Furthermore, the prior researcher's prediction period only accounts for one day's worth of advance[7].

Additionally, extending prediction duration one month ahead of time plays a crucial role for early warning and improving quality of life. In order to take safety precautions and use deep learning technology to address a wide range of rainfall prediction challenges, extend the prediction term one month in advance and build a deep learning model with the goal of month-by-month rainfall forecasting. While some efforts have been made to create rainfall prediction models in Ethiopia, the majority of these models have been created utilizing different statistical and numerical prediction techniques. Therefore, in order to take into account, the non-linear interactions among the data, more effort needs to be put into creating rainfall prediction models employing contemporary data driven techniques[8]. The issue with this strategy, though, is that rainfall is dependent on a wide range of intricate meteorological processes that change over time and space.

Deep learning techniques, which are based on neural networks, are used to find meaningful findings for a big dataset. Using this strategy, we forecast rainfall patterns based on their attributes, ignoring the physical principles driving the rainfall process[9].

This work aims to forecast rainfall through pattern recognition. Deep learning techniques are used to create the prediction models. We offer LSTM, Bi-LSTM and CNN-LSTM model based approach for monthly rainfall prediction and comparison it's result based on stat of the art.

1.3. Research Questions

To guide this research and ensure a comprehensive understanding of the problem domain, several specific research questions have been formulated. These questions are designed to address the core aspects of the study, from data preprocessing to model performance evaluation, and ultimately to the practical implications of the findings. By systematically answering these questions, the study seeks to provide valuable insights and contribute to the development of more reliable rainfall forecasting models. This study aims to answer the following research questions:

- Data preparation and Preprocessing
 - ❖ How can missing values in the meteorological dataset be effectively handled to ensure robust model training?
 - ❖ What normalization techniques are most suitable for preparing the rainfall dataset for deep learning models?
- Model Selection and Configuration:

- ❖ What are the comparative strengths and weaknesses of LSTM, BLSTM, and CNN-LSTM networks in the context of monthly rainfall prediction?
- ❖ How do different hyper-parameters (e.g., number of epochs, batch size, learning rate) affect the performance of deep learning models in rainfall forecasting?
- Model Performance Evaluation:
 - ❖ Which deep learning model (LSTM, BLSTM, or CNN-LSTM) provides the highest accuracy in predicting monthly rainfall for the East Gojjam zone?
 - ❖ What are the key performance metrics (e.g. MSE, RMSE, MAE) that best evaluate the effectiveness of the deep learning models in this study?
- Practical Implications and Applications:
 - ❖ How can the improved rainfall predictions from the selected deep learning model benefit agricultural planning and water resource management in the East Gojjam zone?
 - ❖ What are the potential challenges and limitations of implementing deep learning-based rainfall forecasting models in practical scenarios?

By systematically answering these questions, the study seeks to provide valuable insights and contribute to the development of more reliable rainfall forecasting models.

1.4. Objective

1.4.1. General Objective

The general objective of this research is to develop deep learning-based models for forecasting monthly rainfall model in East Gojjam Zone of Ethiopia.

1.4.2. Specific Objective

In this study, the following specific objectives will be address to achieve the research main objective.

- ✓ To review and analyze the gap in the area
- ✓ To collect and organize the data required for the research work
- ✓ To design and implement a deep learning model that can effectively predict monthly rainfall patterns.
- ✓ To devise a mechanism within the model that can identify and select the most relevant features for accurate rainfall forecasting.

- ✓ To incorporate strategies that prevent overfitting, ensuring the model's generalizability and performance on unseen data.

1.5. Scope of the study

This research focuses on the application of deep learning techniques to forecast monthly rainfall in the East Gojjam Zone, Ethiopia. The study utilizes meteorological data from east gojjam spanning a period of 35 years. The prediction model employs seven key meteorological indicators, providing a comprehensive and detailed approach to rainfall forecasting. The scope of this study is confined to the East Gojjam Zone and does not extend to other regions of Ethiopia. The forecast generated by the model covers the entire month, offering a holistic view of the expected rainfall patterns.

1.6. Limitation of the study

This study is unable to include flood forecasting based on the predicted rainfall level. In this study, the rainfall predictors like cloud cover and sea level pressure are not available in ENMA database. Therefore, those attributes are not incorporated.

If those attributes are incorporated the performance of the model is increased and more robust and also Lack of sufficient reference material, particularly local research, on rainfall forecasting utilizing deep learning technology is the primary constraint of this study. Another drawback is that the researcher has only been able to anticipate rainfall because of the nature of the dataset that is nowadays available and the time constraints for finishing the study.

1.7. Significance of the study

This research is pivotal in enhancing our understanding of machine learning technologies for monthly rainfall forecasting, particularly in the East Gojjam Zone of Ethiopia. By determining the optimal deep learning methods for rainfall prediction, the study contributes to the broader field of climate science and meteorological studies. Accurate rainfall forecasting is critical for regions dependent on agriculture, as it allows for better planning and resource management. The insights gained from this study could serve as a foundation for developing more sophisticated and region-specific predictive models in the future.

Furthermore, the study's findings have significant practical implications for risk management and policy formulation. Effective rainfall forecasting enables the development of efficient risk management systems that can mitigate the adverse effects of extreme weather events, such as

floods and droughts. Policymakers can leverage these forecasts to implement proactive measures, ensuring the safety and well-being of communities. Additionally, accurate predictions support agricultural planning and water resource management, directly benefiting farmers and other stakeholders in the region.

Lastly, this research facilitates proactive, knowledge-driven decision-making by providing valuable data on future weather patterns. The ability to extract predictive information from the National Meteorological Agency's database aids in identifying key variables influencing rainfall. This information is crucial for government planners and decision-makers in crafting informed strategies to address the impacts of climate variability. The developed models not only predict monthly rainfall but also enhance our ability to anticipate and respond to environmental changes, thereby supporting sustainable development in Ethiopia.

1.8. Contribution of the study

This research makes several significant contributions to the field of meteorological forecasting, particularly for the East Gojjam Zone of Ethiopia:

Introduction of Deep Learning Models: This study proposes the use of advanced deep learning models Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (Bi-LSTM), and Conventional Neural Network over Long Short-Term Memory (CNN-LSTM) for predicting monthly rainfall in the East Gojjam Zone. These models have not previously been applied to this dataset, marking a novel application of these techniques in this context.

Benchmark for Future Research: By providing a comprehensive analysis and comparison of various deep learning models, this research establishes a benchmark for future studies. The results and methodologies outlined can serve as a reference point for researchers aiming to improve rainfall prediction models.

Comprehensive Model Comparison: The study includes a detailed comparison of several deep learning algorithms, offering insights into their relative performance and suitability for rainfall prediction. This comparative analysis helps in identifying the most effective model for accurate and reliable forecasts.

Practical Implications: The developed models are not only academically valuable but also have practical implications for agriculture, aviation, and water resource management in

Ethiopia. Accurate rainfall predictions can help in planning and decision-making processes, ultimately benefiting various sectors reliant on weather forecasts.

Enhanced Understanding of Key Variables: The research also contributes to identifying key meteorological variables that significantly influence rainfall prediction. This understanding aids in refining models and improving prediction accuracy.

Overall, this study enhances the existing body of knowledge on machine learning-based weather forecasting and provides a solid foundation for both academic research and practical applications in meteorological services.

1.9. Organization of the Thesis

The rest of the thesis is organized as follows. **Chapter Two** discusses the relevant and related literature reviews on important concepts and elements that were used in the thesis. It discusses the basic concepts of data preprocessing and feature extraction in forecasting systems, and deep learning algorithm were discussed. It includes related works done by different researchers.

Chapter Three presents the methodology we have used to accomplish the research work such as data preprocessing and general architecture of the proposed model also included in detail were discussed.

Chapter Four presents the proposed model implementation and dataset preparation used in the experimental setup described in chapter three. In addition, deals with the experimentation results we got on the proposed model and discusses in detail why and how we got those results.

Chapter Five represents the conclusion and recommendation of the study.

CHAPTER TWO

2. LITERATURE REVIEW

2.1. Introduction

This chapter evaluated the literature on the ideas that serve as the foundation for this argument. First, the researcher gives an outline of the monthly rainfall-forecasting model and talks about typical Ethiopian traits. Subsequently, various methods for executing monthly rainfall forecasting were thoroughly covered, including data preprocessing and normalization. Furthermore, talk about several deep learning algorithms. In conclusion, examine the different studies conducted on rainfall forecasting models in Ethiopian and other countries.

Various researchers have proposed different researches that use various machine-learning algorithms. ANN was used to create a late spring early summer rainfall forecasting model for the Geum River Basin in South Korea. The best ANN model with five input variables had relative root mean square errors of 25.84%, 32.72%, and 34.75% for training, validation, and testing datasets, respectively. The hit score, which is the number of hit years divided by the total number of years, was more than 60%, which indicates that the ANN model successfully predicts rainfall in the study area[10].

The authors proposed a rainfall predictive model for crop recommendation that can be used in some parts of Ethiopia. Their rainfall prediction model was created using ANN and KNN. The three basic rainfall parameters used were maximum temperature, minimum temperature, and average rainfall. They conducted experiments on summer rainfall using meteorological stations in Gojjam and Gonder. However, they did not forecast for all Ethiopian seasons, and their performance needs to be improved.[11]

The authors proposed a rainfall forecast for the Indian state of Kerala using KNN, ANN, and extreme learning algorithms. The rainfall prediction design model for Kerala presented here is crucial in addressing water shortages and preventing drought. They used time series meteorological data from the Indian Institute of Technology Madras (IITM). In terms of precision, the results show that ANN and Extreme Learning Machine (ELM) models outperform KNN models[12].

The authors present an ensemble forecast of semi-arid rainfall using large scale climate predictors. They focused on computing the correlation between climate predictors and seasonal

precipitation over a long term forecast period (1967–2009) for a semi-arid catchment in Iran. Linear regression together with two nonlinear models, the adaptive neuro-fuzzy inference system (ANFIS) and the multi-layer perceptron, were applied to forecast seasonal ensemble precipitation time series[13].

An ensemble forecast of spring precipitation modes showed a stronger correlation with the preceding season (winter predictors) in the ANFIS algorithm. An analysis suggests that seasonal precipitation is statistically aligned with the predictor's variability. Climate modeling and prediction are critical in water resource management, particularly in arid and semi-arid countries where water shortages are common[14].

The authors present a drought index modeling approach based on large-scale climate indices by using the Adaptive Neuro-Fuzzy Inference System (ANFIS), the M5P model tree, and the Multilayer Perceptron (MLP). They used factor analysis to determine the climate signal from 25 climate signals, and then used ANFIS, the M5P model tree, and MLP to forecast the Standardized Precipitation Index (SPI) one to 12 months in advance[13].

The performance of the models was assessed using error parameters and Taylor diagrams, which revealed that the MLP outperformed the other models. Interest in semi-arid climate forecasting has grown due to risks associated with above-average levels of precipitation. Longer-lead forecasts are difficult to make due to short-term extremes and data scarcity. The Classification and Regression Trees (CART) model, which is a rule-based algorithm, was used for prediction of the precipitation over a highly complex semiarid climate system using climate signals. Deep-learning methods have advanced, and research has been done to apply them to time series prediction[5].

The authors proposed a multi-stacked LSTM to forecast temperature, wind speed, and humidity for 24 and 72 hours. They used hourly meteorological data from nine Morocco cities for 15 years, from 2000 to 2015. The authors concluded that deep LSTM networks could effectively forecast weather parameters and recommended that they can be used for other weather-related problems[15].

The authors at predict monthly rainfall over Simtokha, an area in Bhutan's capital, Thimphu. Bhutan's National Center of Hydrology and Meteorology Department (NCHM) provided the rainfall dataset. Based on the parameters reported by the automatic weather station in the area,

they investigated the predictive capability of Linear Regression, Multi- Layer Perceptron (MLP), and Convolutional Neural Network (CNN)[16].

2.2. Ethiopian weather characteristics

Ethiopia is situated above the equator between 3015'N and 180N and 330E and 480E. Ethiopia's borders are as follows: Sudan and South Sudan to the west; Kenya to the south; Eritrea to the north; and Somalia and Djibouti to the east. The nation is made up of twelve river basins with different sizes and possibilities for water resources, covering an area of roughly 1.1 million km². The regions of the nation are classified as highland or lowland, with altitudes varying from 4,620 meters above sea level to 120 meters below it. While there are just two seasons in the tropics the rainy and the dry in Ethiopia, the seasons are classified into four distinct portions according to the pattern of rainfall[5].

2.3. Ethiopian Seasonal Climate Characteristics

Ethiopia is located in a tropical region lying between the equator and tropical cancer of the earth[13]. Therefore, Ethiopia generally has a tropical climate and one of the influences of Ethiopia's climate and weather is the seasonal migration of the Inter-tropical Convergence Zone (ITCZ); that is the position of the earth relative to the sun and by the complex geographical topography of the country[5].

2.4. Rainfall in Ethiopia

Ethiopia divided into four main topographical regions from west to east; the Great Rift Valley, the Somali plateau, the Ogaden plateau and the Ethiopian plateau. Over half of Ethiopia is on the Ethiopian plateau, which is bounded to the west by the savanna and forested Sudan lowlands. It has a number of high mountains that range in elevation from 1524 to 1829 meters above sea level. Ras Dashen, Ethiopia's highest point, is located at a far higher elevation of 4620 meters above sea level. The plateau is divided by several deep valleys and descends gradually from east to west. [11].

Due to these geographic variations, Ethiopia presents a particularly difficult test for climate models. The central part of Ethiopia is controlled by highland plateaus, which split the country into two climatically [17].

The ITCZ, which runs through this more southern location in southern Ethiopia and is recognized for having two different wet seasons (March–May), The primary rainfall season, known as Bega, which lasts from October to December and brings in about 100 mm monthly,

is followed by the main Belg season, which brings in 100–200 mm monthly. The eastern most corner of Ethiopia receives very little rainfall at any time of the year. The movements of the ITCZ are sensitive to variations in Indian Ocean sea surface temperatures and differ from year to year, as the beginning and period of the rainfall seasons vary considerably inter-annually, causing repeated drought[18].

The most well documented causes of this variability is the El Niño southern oscillation. Warm phases of ENSO (El Niño) have been related with reduced rainfall in the main wet season, July, August and September, in north and central Ethiopia, causing drought, but also with greater rainfalls in the early February to April rainfall season which mainly affects southern Ethiopia[5].

So capturing the general patterns of rainfall, climate models should be able to offer a reasonable simulation of the seasonal cycle of rainfall. For the reason that the high spatial variation of rainfall over Ethiopia both in terms of the seasonal cycle and the inter-annual variability, Ethiopia is aggregated into a number of homogeneous rainfall zones [19].

2.5. Machine learning techniques For Rainfall Prediction

Machine learning is a branch of research that focuses on extracting knowledge from datasets, building models for realistic simulations, and applying techniques for pattern recognition, prediction, and classification to the input data[9].

Categorical and Numerical Weather Prediction model are the most used approaches for predicting rainfall. Rainfall data have a non-linear character. For time series rainfall, the three primary parameters are amount, concentration, and frequency. These numbers can change from one moment to the next and from one place on Earth to another position on Earth [36].

2.5.1. Artificial Neural Network Based Weather Forecasting

ANN, often commonly referred to as a "neural network" (NN), is a computational or mathematical model that emulates a biological brain system. It is based on biological neural networks. It processes information using a connectionist methodology and computing to convert a collection of inputs into a set of desired outputs. It is composed of an interconnected group of artificial neurons[4].

An ANN is often an adaptive technique that modifies its structure in response to internal or external data passing through the network while it is learning. Depending on its architecture,

neural networks give the data mining industry associations, classifications, groupings, prediction, and forecasting[8] .

When using ANN to forecast time series, various methods can be applied. One or more one-time series may be utilized as input values in these methods. The Multilayer Perceptron utilizes the McCulloch and Pitts neuron model, which is predicated on an additive aggregation function. Consequently, it is possible to think of the multilayer perceptron's output as a non-linear transformation of some of its inputs. Here, non-linearity is provided via the activation function. A multilayer perceptron consists of one or more neurons in the output layer and many neurons in the hidden layer. The weighted sum of the inputs multiplied by itself yields a non-linear function as the output. In particular, multilayer perceptron neural networks' performance is strongly impacted by the quantity of neurons in the hidden layer[20].

In addition to time-series data, RNNs are intended to capture temporal contextual information. The construction of RNNs includes feedback loops, which are utilized to provide the results of previous time steps as input to the current time step. In addition to the time-series dataset, this construction allows RNNs to generate complex temporal contextual figures. The LSTM, on the other hand, includes a basic structure known as a memory cell that allows it to recall temporal contextual information and propagate unit outputs between multiple time steps[21].

Input layer: A layer's neuron count usually corresponds to the input feature of the network. One or more hidden layers come after the input layers[9].

Hidden layer: A feed-back neural network has one or more hidden layers. Neural networks use the weight values on the connections between the layers to encode the knowledge they have learnt from the row training data. Neural networks can model nonlinear functions thanks to hidden layers[9].

Output layer: The output layer provides the model's output (classification or prediction). Based on the input from the input layer, the output layer provides us with an output. The neural network's ultimate output could be a list of probabilities (classification) or a real-valued output (regression), depending on how it was configured. The kind of activation function we apply to the neurons in the output layer regulates this[9].

Connections between layers: In a fully linked feed back network, links between layers are the connections that flow out from every neuron in one layer to every other neuron in the following layer. As the backpropagation learning algorithm discovers the optimal solution, these weights are gradually adjusted[9].

In contrast to traditional weather forecasting, which uses three weather parameters such as effective temperature, wind, and effective illumination, the researchers proposed "Neural Network Load Forecasting with Weather Ensemble Predictions" to investigate the use of weather ensemble predictions in the application of neural networks to load forecasting for lead times from 1 to 10 days ahead. Ultimately, they draw the conclusion that neural network load forecasting has a great deal of potential for using weather ensemble predictions[14].

Using only the average maximum and minimum temperature, the researcher presented "A Time Series ANN Approach for Weather Forecasting" to predict the atmospheric condition. They train multilayer perceptron on 60 years' worth of historical real data, and the results they obtain are validated using Mean Square Error (MSE) algorithms. Ultimately, the researchers draw the conclusion that multilayered neural networks can be useful instruments for predicting the weather. Nevertheless, they did not contrast the newly suggested model outcome with the output of previous weather predicting techniques[8].

2.6. Deep learning

Deep learning can perform better on forecasting rather than Machine learning. Even it provides higher accuracy than machine learning. We can tune different parameters according to our data and model. Just one requirement is that we need good GPU power for training. Moreover, another thing is that it takes a longer time to train but it is worth it [29].

Deep learning performs significantly better when it comes to tackling actual problems. Deep learning is needed to evaluate and learn in a vast amount of comfortable data in the event of more and more big data, which is to integrate deep learning and big data technologies. Deep learning's levels of complexity and layers are designed for this type of data, but as more data is collected, the more layers of a neural network get simpler, which is one of the reasons why deep learning has become a popular streaming computing technology. Deep learning, on the other hand, has far more applications than machine learning [2].

Deferent researchers suggested that in this research, we employed deep learning algorithms to forecast monthly rainfall. We suggested a rainfall forecast model based on deep learning for the selected area, East Gojjam zone. Layer of Input is Artificial input neurons comprise the input layer of a deep learning system. They feed the system's first preprocessed meteorological data so that further layers of neural neurons can process it. One particular type of recurrent neural network (RNN) that can learn long-term dependencies is called an LSTM[9].

2.6.1. Long short Term Memory (LSTM)

An artificial recurrent neural network (RNN) architecture called long short term memory (LSTM) is employed in the field of deep learning[9]. In order to address the issue of disappearing/vanishing gradients that arises during the training of conventional RNNs, LSTMs were created. In many cases, LSTM has an advantage over RNNs, hidden Markov models, and other sequence learning techniques due to its relative insensitivity to gap length. LSTM features feedback connections, in contrast to conventional feedforward neural networks like RNN. It can process whole data sequences (like audio or video) in addition to individual data for time series data prediction[5].

For example, LSTM is applicable to tasks such as handwriting recognition, speech recognition, and Grammar learning and anomaly detection in network traffic or intrusion detection systems, Sign language translation and weather forecasting etc. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. With the advancement in deep learning techniques, deferent researcher to be done to implement it in time series prediction, Recurrent neural networks (RNNs), in particular LSTM and CNN have found their niche in time series prediction[22] [23].

The authors used multi-stacked LSTM to forecast 24 h and 72 h of weather data, i.e., temperature, wind speed, and humidity. They used 15 years of hourly meteorological data from 2000–2015 of nine cities of Morocco. The authors deduced deep LSTM networks could forecast the weather parameters effectively and suggested it for other weather-related problems[5].

2.6.2. Bidirectional Long short Term Memory (Bi-LSTM)

This model's Bi-LSTM layer receives input from preprocessed meteorological parameters. By reading data both forward and backward, this layer creates an appropriate embedding. The output of the Bi-LSTM layer is batch normalized prior to transferring the hidden embedding to the next layer. The data are batch normalized one more before being submitted to the final dense layer. The single neuron in the last layer, where T is the current month, uses a linear activation function to forecast the monthly rainfall for T + 1 (next month)[9].

The Many-to-One (many input and one output) variant of LSTM, which was trained using the weather parameters from the previous 12 months, was used to estimate the amount of rainfall

for the future month. The activation function in Bi-LSTM and CNN-LSTM was the default tanh function on the output layer and sigmoid on the hidden layer and Adam served as the optimizer. The design was repaired the following a significant over-adjustment of the parameters. Hyper-parameter was accomplished using a randomized grid search and the programmer's heuristic expertise[24].

2.6.3. Convolutional Neural Network Long-Sort Memory (CNN-LSTM)

CNN-LSTM Hybrid Classifier Model In the CNN-LSTM hybrid algorithm, the convolutional layers in the CNN algorithm are used to extract features through the input data whereas the LSTM algorithm is used to analyze short-term and long-term dependencies[16].

A, Pre-processing: the raw dataset is acquired on a per-day basis for the previous 10 years. The analysis of features is carried out to obtain essential attributes from the data. The Pre-processing is carried out on the raw data collected to remove unwanted values, remove redundancy in the data collection, and clean the data thoroughly to be fed into the model for training[16].

B. CNN-LSTM hybrid classifier Model: A common approach is to adopt a parameter optimization of a neural network to work in different combinations to estimate the performance of a network. This classifier involves a high computational analysis of a huge number of training parameters[2].

Generally, there is no existing monthly rainfall forecasting system due to East gojjam zone. This method is based on Deep Learning, which has seen tremendous success in forecasting tasks. As a result, we propose LSTM, Bi-LSTM and CNN-LSTM model rainfall prediction to improve the performance of model and compare its result based on stat of the art for rainfall forecasting for east gojjam zone, Ethiopia.

This research focuses on monthly rainfall using various parameters such as temperature, rainfall, sunshine, wind speed and humidity data obtained from NMSA in East Gojjam Zone. The following are some of a summary of the literature review is shown below.

2.7. Related work Summary

Table 2. 1: Related work

Author /Year	Title	Daily- Monthly- Yearly	Types of NN	Rainfall Predicting Variables	Error Measure	Gaps of Study
Endalie, Demeke Haile, Getamesay Taye,Wondmagegn	Deep learning model for daily rainfall prediction: case study of Jimma, Ethiopia	15 years daily recorded data	LSTM	Minimum temperature Maximum temperature Solar radiation Wind speed Relative humidity Precipitation	RMSE	they did not forecast for all month
Lee, Jeongwoo	Application of artificial neural networks to rainfall forecasting in the Geum River Basin, Korea	5 years daily data	ANN	humidity and wind speed Min and max temperature	RMSE	Forecasting only spring early summer rainfall
Abhishek et al,[4] 2012	A rainfall prediction model using artificial neural network	8 months of data from 1960 to 2010	BPFNN, BPA, LRN and CBP	Average humidity and average wind speed	MSE	Use single station data features.
Khajure et al,[25] 2016	Future weather forecasting using soft computing techniques	Daily records for 5 years	ANN and a fuzzy inference system	Temperature, humidity, dew point, visibility, pressure and wind speed.	MSE	did not contrast the newly suggested model

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1. Introduction

This chapter provided theoretical information about the methods and materials applied to implement monthly rainfall forecasting system. In this study, the researcher discusses the design and system architecture of the proposed work. The model has two components dataset preparation and deep learning implementation.

Dataset preparation performs data collection, data preprocessing, and train a model using LSTM, Bi-LSTM and CNN-LSTM deep learning algorithms and compare them using model evaluation metric. In addition to dataset preparation, the researcher discusses the implementation software tool and model evaluation technique used to measure the performance of the proposed system. This study's primary goal is to use deep learning to forecast monthly rainfall, with a focus on the east Gojjam zone.

The methodologies used in this thesis are given below.

- ✓ **Literature survey:**-reviewing various literatures that are relevant to my work and theoretical ideas that provide support.
- ✓ **Select algorithm:** - choose the best algorithms from deep learning neural networks that are quite effective at predicting atmospheric conditions.
- ✓ **Propose the system:**-Examine the current system and suggest a new one with a forecasting model that works well in Ethiopia.
- ✓ **Design and Implement:**- utilizing simulation software for the suggested system's design and implementation
- ✓ **Test and Validate:**-Validating and testing the developed system as well as the outcome of the suggested system, respectively

3.2. Study Area

3.2.1. Overview of Ethiopian National Metrology Agency

According to the Ethiopian Meteorological Agency, missionaries began to observe the weather in Addis Ababa toward the close of the 19th century. Furthermore, meteorological stations were founded in Adamitulu and Gambela in 1890 and 1986, respectively. However, a department of civil aviation was also established in the early 1950s to manage meteorological activity due to the increased demand for meteorological information for safe air transport

operations and daily activities. When the economic and social sectors finally realized how important meteorological services were, the metrological station's role was changed to include supporting non-aviation related operations[26].

According to the Ethiopian National Meteorological Agency, the Ethiopian government on December 31, 1980, formally created the National Meteorological Services Agency by proclamation no. 201 of 1980. The Ethiopian National Meteorological Agency now uses ammunition from the United States and the United Kingdom[26].

Additionally, NMA forecasts weather using a conventional statistical procedure that is both dynamic. The Wolf model, employed by the National Meteorological Agency, is based on a mathematical formula created by a meteorologist. They did not, however, make use of sophisticated machine learning methods. Thus, data mining and intelligent machine learning techniques are quite advantageous for enhancing prediction performance[24][27].

3.3. Data source

We used historical data records from Bahir Dare Station spanning thirty-five years (1988–2023) to conduct this analysis. The record that was obtained includes observations of seven features those are date when created in experiment, minimum temperature, maximum temperature, rainfall, sunshine, wend speed and humidity, data column is created through experiments. These features are described as follows:

- ❖ **Relative Humidity:** This is the ratio of water vapor contained in the air compared to the maximum amount of moisture that the air can hold at that specific temperature and pressure.
- ❖ **Sunshine:** Sunshine (Daylight) is a portion of the electromagnetic radiation specified off by the Sun, in certain infrared, visible, and ultraviolet light. On Earth, sunshine is scattered and filtered through Earth's atmosphere, and is obvious as daylight when the Sun is above the horizon.
- ❖ **Temperature:** This is amount of the air's hotness or coldness and is the most measured quantity of the air. Temperature is measured in degrees on the Fahrenheit, Celsius, and Kelvin scales.
- ❖ **Rainfall:** This is a type of precipitation that occurs when water vapor in the atmosphere condenses into droplets that can no longer be suspended in the air. The occurrence of rainfall is dependent upon several factors. Things such as prevailing wind directions, ground elevation, and location with respect to mountain ranges.

❖ **Wend speed:**-the movement of air in ratio is called wend speed.

Table 3. 1: Data set features

	date	Rainfall	Weend sped	Sunshine	Rhumid	Tmaximum	Tminimum
0	1988-01-02	14.4	1.5	8.2	54.1	94.00	9.9
1	1988-01-03	6.5	0.8	7.8	40.5	83.50	5.9
2	1988-01-04	4.8	0.3	7.8	47.7	87.81	5.9
3	1988-01-05	8.8	0.3	8.9	40.8	93.62	5.9
4	1988-01-06	28.6	0.3	7.6	46.9	94.81	5.9

3.4. Dataset Set Preparation

For this study, the researcher takes ENMA data set dividing the available sample dataset into a training dataset and the testing dataset is the most common approach used in machine learning to build and evaluate prediction models. Those are training dataset, and testing dataset (80%, 20%) respectively depend on the data set, in this study 12960 data is small the concept of big data so the training data is more trained so we select this scenario. In addition, monthly dataset for testing the developed LSTM, Bi-LSTM and CNN-LSTM model and compare the result using evaluation matric. All dataset variables shows in the following table.

Table 3.2: Features and corresponding units

Number	Field Name	Data Type	Description	Unit
1	date	Date time	Month considered	
3	Rainfall	Numeric	Rain fall of the Month	Millimeter(mm)
4	Relative humidity	Numeric	Relative humidity of the Month	Percentage (%)
6	Sunshine	Numeric	Sunshine of the Month	Hours (h)
7	Maximum Temperature (<i>t max</i>)	Numeric	Max temp of the Month	°C
8	Minimum Temperature (<i>t min</i>)	Numeric	Min temp of the Month	°C
9	Wind speed	Numeric	Wind speed of the Month	Meters per second (m/s)

3.5. Data preprocessing

Data preprocessing is a fundamental step in data analysis and machine learning. It involves transforming raw data into a format that is more suitable for modeling and analysis. The goal of data preprocessing is to improve the quality of the data and enhance the accuracy and efficiency of any subsequent analysis or modeling processes.

The NMSA provided the monthly weather feature records spanning the years 1988 through 2023. Seven features were present in the raw data at first; however, several of them had a high number of noisy and missing values. So on this work the following techniques are used.

3.5.1. Data Reduction

Data reduction refers to techniques and processes used to reduce the volume or complexity of data while preserving its informational content. In this work to use Feature Selection techniques such as domain knowledge understanding the domain and problem context is crucial for selecting appropriate data reduction techniques and ensuring that the reduced dataset retains its relevance and usefulness by talking expert's idea.

3.5.2. Data cleaning

Data cleaning, also known as data cleansing or data preprocessing, refers to the process of identifying and correcting errors, inconsistencies, and missing values in a dataset to improve its quality and prepare it for analysis. so under this on this work to use handling missing data.

3.5.3. Handling missing data

Handling missing values is a crucial step in data preprocessing to ensure that the data used for analysis or modeling is as complete and accurate as possible. In this study to use mean imputation techniques.

3.5.4. Data Normalization

Data normalization is a technique used to standardize the range of numerical data to a common scale, without distorting differences in the ranges of values. It is a crucial preprocessing step in many machine learning and data analytics tasks. in this study to use Min-Max Normalization. MinMax Scaler to get the new scaled value Z[24].

$$Z = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

Where $\min(x)$ and $\max(x)$ are the minimum and maximum value, respectively. Z is the value to be scaled.

Data transformation should be done to fit and converge the model. The performance of a neural network is improved through data transformations (normalization). There are three existing data transformation methods.

- ❖ Linear transformation
- ❖ Statistical standardization and
- ❖ Mathematical function

Among the data transformation methods, in this study to be used one is scaling which is a type of linear transformation. Scaling is done to have a specific range, such as the range between -1 and 1 or 0 and 1, for the entire data set of a given station[28].

3.6. Model Development

3.6.1. Proposed model Architecture

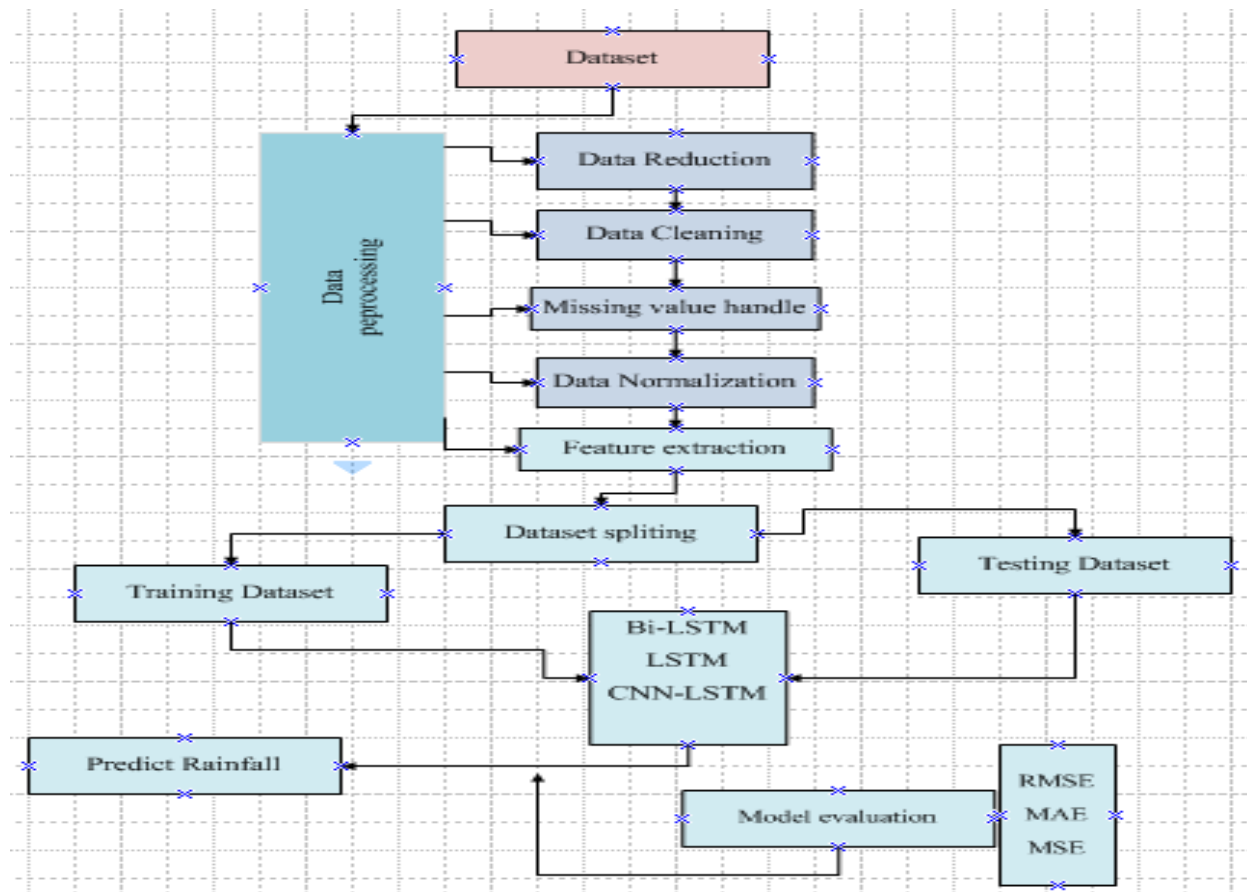


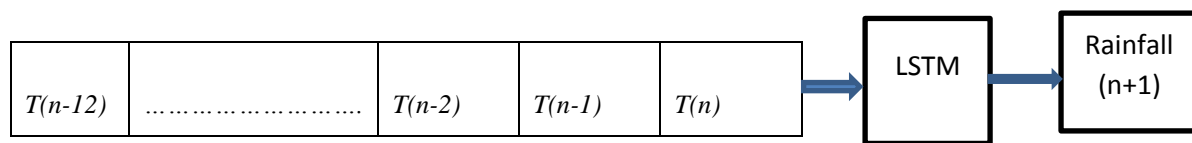
Figure 3. 1: Proposed model architecture

3.6.2. Long Sort Term Memory (LSTM)

In time series analysis, LSTM is the most widely used model. It comes in a variety of forms, including unidirectional LSTM and Bi-LSTM. As seen in Figure 3.4, the rainfall for the

upcoming month was predicted using the Many-to-One (many input and one output) form of LSTM, which was trained using the weather parameters from the previous 12 months.

Data from unidirectional LSTM processes are solely reliant on historical data. Bidirectional processing makes the greatest use of the data by iterating through time-steps both forward and backward. To get two layers next to each other, it replicates the architecture's first recurrent network. It gives the first layer the input exactly as it is and gives the second layer a copy that has been flipped. Even if it was customary Its application has expanded beyond rainfall forecasting to improve LSTM performance across a variety of domains[24].



Each sample of data contains 12 time-steps of previous data. We used 12 months of previous data to predict the rainfall of the next month (n + 1).

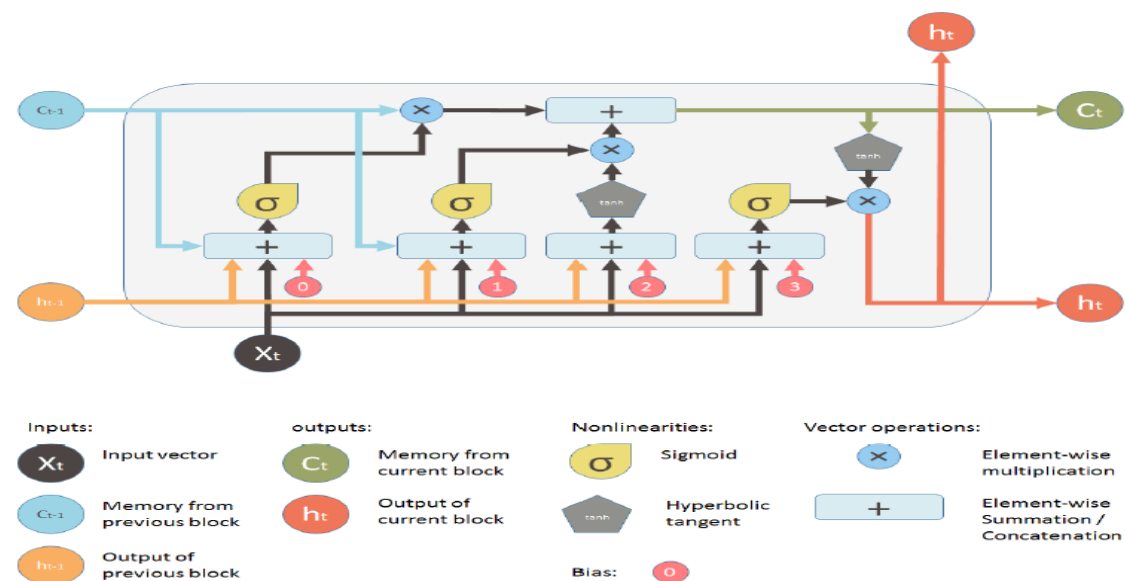


Figure 3. 1: LSTM architecture

On the above figure, the input gate accepts new data from the output layer the sigmoid function is used as an activation function for each hidden layer node and the output layer node. The sigmoid activation function formula performs on the hidden and output layer is shown below.

$$Y = \frac{1}{1+e^{-x}} \quad (2)$$

Where Y is the output of the hidden or output layer, X is the input from the input layer or hidden layer.

On the above architecture, LSTM cell used to remember the previous output feed as input for the current output. The contextual information store on the LSTM cell has three gates input gate, forget gate, and output gate. In this cell, the input gate holds the value comes from the output layer and gives it to the output gate, the output gets to accept the data from the input gate then feed to the hidden layer and the forget gate used to forget the previous input gate data.[29]

✓ LSTM

A. Input get:-

$$i_t = \sigma(W_{ix} \cdot x_t + W_{ih} \cdot h_{t-1} + b_i) \quad (3)$$

i_t represents the input gate activation.

σ is the sigmoid activation function.

W_{ix} and W_{ih} are the weight matrices for the input gate applied to the input x_t and the previous hidden state h_{t-1} , respectively.

b_i is the bias term for the input gate.

B. Forget Gate f_t .

$$f_t = \sigma(W_{fx} \cdot x_t + W_{fh} \cdot h_{t-1} + b_f) \quad (4)$$

C. Output Gate o_t .

$$o_t = \sigma(W_{ox} \cdot x_t + W_{oh} \cdot h_{t-1} + b_o) \quad (5)$$

The ReLU activation function is defined as:-

$$ReLU(x) = \max(0, x) \quad (6)$$

For example, if h_t represents the output of the LSTM cell at time step t , then the output after applying ReLU activation can be computed as:

$$ReLU(h_t) = \max(0, h_t) \quad (7)$$

3.6.3. Convolutional Neural Network with Long-Sort-Memory (CNN-LSTM)

3.6.3.1. Convolution Neural networks (CNN)

In the CNN-LSTM hybrid algorithm, the convolutional layers in the CNN algorithm are used to extract features through the input data whereas the LSTM algorithm is used to analyze short-term and long-term dependencies. In addition, every problem has a unique set of data and

strategies that are applied are different to different problems. And each challenge has its own set of facts, as well as distinct techniques for different difficulties.

To solve this, CNN is utilized to generate features automatically and mix them with the classifier. The list of layers that translate input volume to output volume is the simplest of all the classifiers in this method, which is one of the advantages of the CNN classifier. There are only a few separate layers, and each layer uses a differentiable function to translate the input to output. Used multi-task CNN to predict short-term precipitation using weather parameters collected from multiple rain gauges in China. The authors concluded that the multi-site features gave better results than single-site features. A convolution is a significantly slower operation than, say max pool, both forward and backward. If the network is deep, each training step is going to take much longer[9].

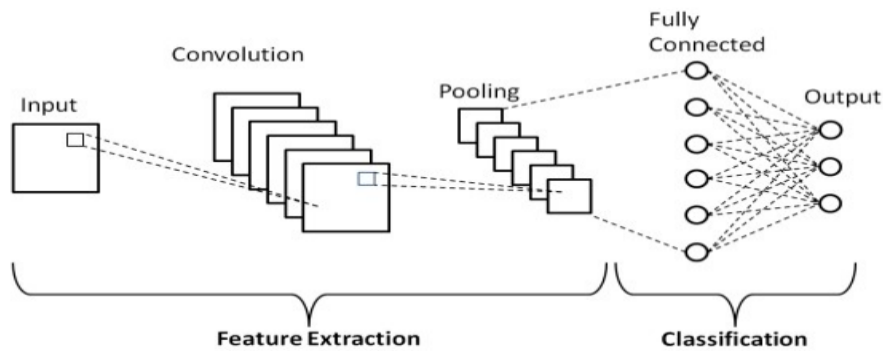


Figure 3. 2: CNN architecture [28]

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

Preprocessed meteorological parameters are input into BLSTM layer. This layer generates a suitable embedding by reading data both forward and backward. Before sending the concealed embedding to the following layer, batch normalization is applied to the Bi-LSTM layer's output. Prior to being sent to the last dense layer, the data are batch normalized once again. With a linear activation function, the single neuron in the final layer predicts the monthly rainfall for $T + 1$ (next month), where T is the current month.

The rainfall for the upcoming month was predicted using the Many-to-One (many input and one output) form of LSTM, which was trained using the weather parameters from the previous 12 months. Then, after extensively over adjusting the parameters, the architecture was fixed. In this work to use a grid search were used to do hyper parameter tuning[24].

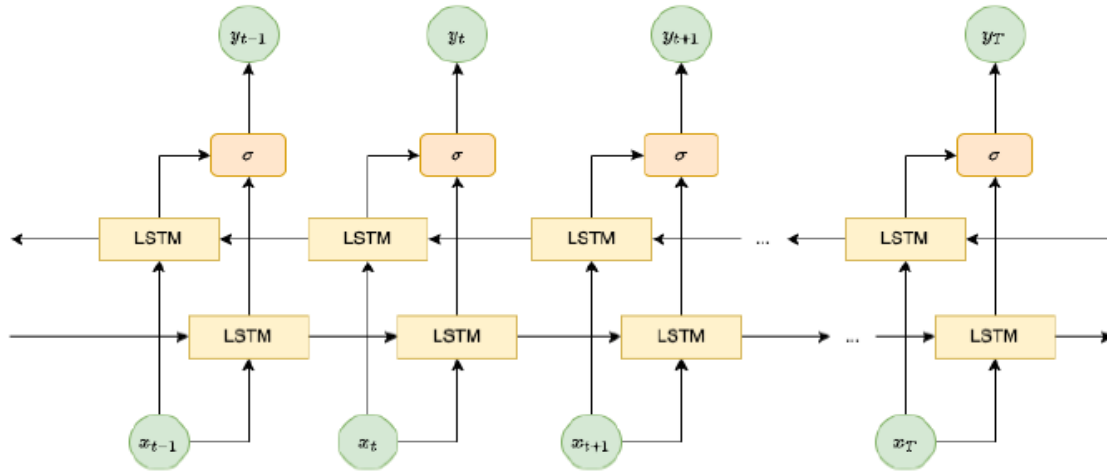


Figure 3. 3: Bi-LSTM architecture [29]

The formula for applying ReLU activation to the output of the BLSTM cell at time step t can be expressed as:

$$ReLU(ht) = [\max(0, ht^{(f)}); \max(0, ht^{(b)})] \quad (8)$$

On this research, to use the above algorithms such as LSTM, BLSTM and CNN-LSTM based on state of the art to compare its performance.

3.7. Model Implementation Tools

As implementation tool for a proposed model in this thesis, different machine learning (TensorFlow, Keras), data science libraries (Pandas, NumPy), visualization libraries (Matplotlib) and Google Colab were used. In addition to data preprocessing, data augmentation, and data splitting, some small programs were written and applied on Anaconda Jupyter notebook. The entire work of the thesis is coded using Python programming language. The details of these basic software tools applied in this study are depicted below.

Python Programming Language: is an interpreted high-level programming language that is object-oriented and it is an integrated dynamic semantic primarily for app and web development developed by Guido Van Rossum in 1991. For this study, python software was used as the main development tool, to perform data preprocessing, data analysis and data understanding. Python is chosen because it is a dynamically typed language and it has garbage collection so we don't feel to worry about unnecessary garbage in programming, It supported procedural programming, functional programming, and also object-oriented programming.

Pandas is a high-level data manipulation tool. It is built on the numpy package and its key data structure is called the Data Frame. Data Frames allow you to store and manipulate tabular data in rows of observations and columns of variables. It is popular Python-based data analysis toolkit which can be imported. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a Numpy matrix array. This makes pandas a trusted in data science and machine learning.

TensorFlow, It is an end-to-end open-source platform for machine learning and numerical computation, the tactic of acquiring data, training models, serving predictions, and refining future results. TensorFlow brings together models and algorithms for Machine Learning and Deep Learning. It's originally based on C++ and uses Python as the front end. TensorFlow is at the present the foremost popular software library. Multiple real-world applications of deep learning and machine learning make tensor flow popular [51].

NumPy: It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Google Colaboratory or Colab: used to allow you to write and execute Python in your browser, with zero configuration required, free access to GPUs, and Easy sharing.

Jupyter Notebook: - It is an open-source web-based application that consists of two categories such as the executable document or the computer program (code) written by programming languages like python and can be executed or run to perform a specific action and easily human-readable documents such as texts, paragraphs, headings, equations, graphs and links.

Keras is the most popular and powerful library running on top of TensorFlow. It has its graph data structure for handling computational graphs and communicating with TensorFlow. Keras has two main types of models to work with: sequential and functional. The sequential model is designed for simple architectures. We wrote the implementation on python using Jupyter notebook editor.

All the implementation codes are written using Jupyter notebook editor and Google Colab. The main procedures for the implementation of the model on Jupyter notebook is presented below.

Step1. Import all necessary packages on the editor

Step2. Load the dataset

Step3. Split the loaded data into training, validating and testing dataset

Step4. Design the model with three layers (input layer, hidden layer, and output layer)

Step5. Compile the model it consists of error metrics and error optimizer

Step6. Fit the model with training dataset and validating dataset with iterations (epochs)

Step7. Evaluate the model with the testing dataset

Step8. Predict the future using the testing input dataset

3.8. Experimental Setup

All the training, preprocessing, and experimental tasks were done on Toshiba laptop with Intel (R) Core (TM) i5-2520M CPU @ 2.50GHz processor speed and 4 GB RAM having window 8 operating system. Google Colab (Google Collaboratory) is a free Integrated Development Environment (IDE) from Google to support research and learning about Artificial Intelligent (AI). Collaboratory provides a code environment as Jupyter Notebook, and it is free to use Graphic Process Unit (GPU) and Tensor Process Unit (TPU).

Google Colab has pre-installed libraries that are very popular in Deep Learning research such as TensorFlow, Keras . Due to machine learning/deep learning algorithms require the system to have high speed and processing power (usually based on GPU), normal computers are not equipped with GPU. Therefore, Colab supplies GPU (Tesla V100) and TPU (TPUv2) on the cloud, one of the highest performing GPUs at the moment, to assist AI researchers [31].

Due to the proposed design based on a deep learning approach, it is difficult to run on this CPU speed so, the design Bi-LSTM, LSTM and CNN-LSTM model runs Google Colab. This tool is one of the most famous cloud services for seasoned data scientists, researchers, and software engineers. It is a cloud service based on Jupyter Notebooks for disseminating machine learning education and research. It provides a runtime fully configured for deep learning and free-of-charge access to a robust GPU.

3.9. Model Evaluation Techniques

In this research work, various performance evaluation methods are used to evaluate the trained model. The most common and widely used metric is RMSE, MAE and MSE.

Mean Squared Error: Mean-squared error is one of the most commonly used measures of success for numeric prediction[7]. This value is computed by taking the average of the squared differences between each computed value and its corresponding actual value. MSE measure using the following formula:

$$MSE = \frac{1}{n} \sum_{k=0}^n (\text{Actual} - \text{Predicted})^2 \quad (9)$$

Where:- y_i is the actual value of the i^{th} observation

n is the number of observations or data points. \bar{y}_i is the predicted value of the i^{th} observation.

Mean Absolute Error (MAE):- is a metric used to measure the average absolute differences between the predicted values and the actual values. It provides a measure of the average magnitude of errors without considering their direction, making it a useful metric for understanding the average error magnitude of a model.

The formula for MAE is:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \bar{y}_i| \quad (10)$$

Where:

n is the number of observations or data points.

y_i is the actual value of the i^{th} observation

\bar{y}_i is the predicted value of the i^{th} observation.

Root Mean Square Error (RMSE) is a commonly used metric to measure the difference between predicted values and actual values. It's especially popular in fields like statistics, machine learning, and signal processing.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (11)$$

The model error or performance is calculated using the above formula. To improve the performance of the model, an error minimization technique needed.

$$\text{Accuracy} = 1 - \text{performance metric} \quad (12)$$

CHAPTER FOUR

4. EXPERIMENT AND RESULT DISCUSSION

4.1. Introduction

In this chapter, the researcher discussed experimental evaluation of the proposed model for forecasting monthly rainfall east gojjam zone, which was specified in detail in the previous chapter. The dataset used, developed model and experimental result of the proposed models are described in detail. Experimental evaluation approves the realization of the proposed model.

4.2. Dataset Set Preparation

For this study, the researcher takes metrology dataset from the Debre Markos station and Bahir dare station respectively. As much as possible, the collected data contain different file format. To forecast monthly rainfall model the total dataset is divided into two training and testing sets, using the percentage split technique. After collecting all data used for the proposed model, then we label data set in excel. dataset labeling is an essential task for supervised machine learning and deep learning techniques because the output result of the model is determined by the labels and feed the model in the training stage.

For each dataset, the labeled information was saved in `datasetnew.csv` format compatible with Bi-LSTM, LSTM, CNN-LSTM models and compare based their performance. The researcher has used 12960 data to be used. From this, the researcher used 80% for training and 20% for testing purposes. This means that 80% of the dataset, which is 10367 data's, are used to train the model, 20% of the dataset which is 2074 and validating dataset is 518 are used to train, test and validate the model.

4.3. Hyper Parameters in the Model

In the context of deep learning, Grid Search, Manual Search, and Random Search refer to techniques for hyper-parameter tuning, which involves finding the optimal set of hyper-parameters for a deep learning model. Here is how these techniques are applied specifically in deep learning:

Grid Search: In Grid Search for deep learning, to define a grid of hyper-parameters with specific values or ranges. These hyper-parameters may include learning rate, batch size, number of layers, number of neurons per layer, dropout rate, etc. Grid Search then systematically evaluates all possible combinations of hyper-parameters using cross-validation

or a validation set. For each combination, the model is trained and evaluated, and the best performing set of hyper-parameters is selected based on a chosen evaluation metric. In this study the researchers use Grid Search because it to define hyper-parameter with specific values.

Manual Search: Manual Search in deep learning involves the manual selection of hyper-parameters based on the practitioner's intuition, experience, and domain knowledge. This process typically involves iteratively training the model with different hyper-parameter configurations and evaluating its performance. The practitioner adjusts the hyper-parameters based on the observed performance until satisfactory results are achieved. Manual Search requires a deep understanding of the model architecture, dataset characteristics, and the impact of hyper-parameters on model performance.

Random search: Random Search for deep learning randomly samples hyper-parameters from predefined distributions. Instead of exhaustively searching through all possible combinations like Grid Search, Random Search selects a fixed number of hyper-parameter configurations randomly. Each sampled configuration is then evaluated using cross-validation or a validation set, and the best-performing set of hyper-parameters is selected based on the evaluation metric. Random Search is more efficient than Grid Search in exploring the hyper-parameter space, especially in high-dimensional spaces, as it covers a broader range of hyper-parameter combinations.

In deep learning, hyper-parameter tuning is crucial for achieving optimal model performance. While Grid Search ensures a thorough exploration of the hyper-parameter space, it can be computationally expensive. Manual Search leverages human expertise but may not efficiently explore the space. While training the model, some hyper-parameters i.e. variables determine the network structure for the optimized result of training. After the model has been developed and the dataset is ready to be feed to the model there are some parameters that we need to train the model. Those are:

Batch: parameter indicates the **batch size** used during training. The training process involves iteratively updating the weights of the neural network based on how many mistakes it is making on the training dataset. It is unnecessary to use all data in the training set at once to update the weights. So, a small subset of data is used in one iteration, and this subset is called the **batch size**. For this work, batch size is set to 64; it means 64 data are used in one iteration to update the parameters of the neural network.

Learning Rate: a positive fraction determining the step of learning of the network neurons. It controls how aggressively we should learn based on the current batch of data. Typically, this is a number between 0.01 and 0.0001.

Epoch:- In the context of training a neural network, an epoch refers to one complete pass of the entire training dataset through the neural network. During each epoch, the neural network iteratively updates its weights and biases based on the training data to minimize the chosen loss function so on this study compare the model using deferent values of epoch 10, 50 and 100 number of iterations.

Optimizer: - an optimizer is an algorithm that adjusts the parameters (weights and biases) of the model in order to minimize the error or loss function during the training process. The goal of optimization is to find the set of parameters that results in the best possible performance of the neural network on the given task, such as classification or regression.

Optimizers play a crucial role in the training of neural networks by efficiently updating the model parameters in the direction that reduces the loss. There are various optimization algorithms available, each with its own characteristics and advantages. Some of the commonly used optimizers include:

- ✓ **Stochastic Gradient Descent (SGD):** This is one of the simplest optimization algorithms. It updates the parameters in the direction of the negative gradient of the loss function with respect to the parameters. SGD can be used with or without momentum.
- ✓ **Adam (Adaptive Moment Estimation):** Adam is an adaptive learning rate optimization algorithm that combines the advantages of both AdaGrad and RMSProp. It maintains separate adaptive learning rates for each parameter and computes individual adaptive learning rates for each parameter.

Activation function:- an activation function is a mathematical operation applied to the output of each neuron in a neural network layer. It introduces non-linearity into the network, allowing it to learn complex patterns and relationships in the data. Activation functions are a fundamental component of neural network architectures and play a crucial role in determining the output of the neurons.

A. Sigmoid: The sigmoid activation function, also known as the logistic function, squashes the input into a range between 0 and 1. Sigmoid functions are often used in the output layer of binary classification models where the goal is to predict probabilities.

B. Rectified Linear Unit (ReLU): The ReLU activation function returns zero for negative inputs and the input value for positive inputs. It is expressed as $\text{ReLU}(x) = \max(0, x)$. ReLU functions are widely used in hidden layers due to their simplicity and effective. ReLU performed on regression tasks but sigmoid perform classification tasks.

Based on our dataset the researcher have set the value for the above parameters by making a comparison according to training time and performance. The hyper-parameters and the value assigned for each are described below.

Table 4. 1: Hyper-Parameters and Values

Parameter	Value	Remark
Batch Size	64	Selected using Grid search
Activation function in hidden layer	sigmoid	Selected using Grid search
Number of hidden layers	2	Obtained by different experiment
Learning rate	0.001	Selected using Grid search
Number of Epoch	100	Obtained by different experiment
Optimizer	Adam	Selected using Grid search
Loss function	MSE	Commonly used in forecasting tasks & Obtained by different experiment
Activation function in Output layer	tanh	Best in Regression task
Early Stopping	Patience =5	Obtained by different experiment
Dropout layer	0.2	Obtained by different experiment

4.4. Experiential Result

Different experiments are conducted to evaluate the prediction performance using origin LSTM, BLSTM, and CNN-LSTM models. Then data set split is performed by creating datasetnew.csv file that contains the path to dataset. Therefore, two csv files (train.csv and test.csv) were created for training and testing the model.

To start BLSTM model training the prepared data set is compressed and uploaded to Google Drive. Then the data set is extracted into a directory called data.

4.4.1. Experiential Result of Bi-LSTM model

The proposed Bi-LSTM model is evaluated with the three basic scoring metrics, i.e., RMSE, MAE and MSE. The results of the experiments are presented based on a neuron in Bi-LSTM model. In addition to using RMSE, MAE and MSE to assess the proposed deep learning based monthly rainfall prediction model.

We also assess the model's prediction accuracy using data that was not used during the training phase. Bi-LSTM model in terms of RMSE, MAE and MSE. On the other hand, we evaluate the predictive performance of the proposed model using a testing dataset. The outcome demonstrates that the proposed model performs well because it mitigates some types of errors. The results of the proposed (Bi-LSTM) based rainfall prediction model for forecasting rainfall. The results show that the proposed methodology is 98.8% accurate in forecasting average rainfall (in mm).

The red line on the graph represents the amount of average rainfall predicted by the model, while the blue line represents the actual amount of average rainfall measured by the rain gauge. As a result, the proposed model can be used to predict rainfall of a specific month. The plot shows the actual monthly rainfall values over east gojjam zone collected from NMSA and predicted rainfall values for 35 years, where the x-axis and y-axis represent month and rainfall values respectively.

When you observe a loss graph and notice overfitting, it typically means that the training loss continues to decrease while the validation loss starts to increase or stagnate.

Training Loss: The training loss steadily decreases over epochs as the model learns to fit the training data better. This is expected behavior during the training process.

Validation Loss: Initially, the validation loss decreases as well, indicating that the model is improving its generalization performance on unseen data. However, at some point, the validation loss either starts to increase or stops decreasing significantly, even though the training loss continues to decrease.

When you observe an actual versus predicted graph and notice overfitting, it typically means that the model performs well on the training data but poorly on unseen data.

Training Data Points: The actual versus predicted graph may show the actual target values plotted against the corresponding predicted values for the training dataset. Since the model has been trained on this data, it often fits the training data well. As a result, the points might cluster closely around the diagonal line ($y = x$), indicating a strong correlation between the actual and predicted values.

Validation or Test Data Points: The graph may also include data points from a separate validation or test dataset, which the model hasn't seen during training. In the case of overfitting, the model's predictions on this unseen data may deviate significantly from the actual values. Instead of clustering around the diagonal line, the points may be more scattered, indicating poor performance on unseen data.

On the above two graph, When you observe a loss graph and notice overfitting, it typically means that the training loss continues to decrease while the validation loss starts to increase or stagnate. When you observe an actual versus predicted graph and notice overfitting, it typically means that the model performs well on the training data but poorly on unseen data.

4.4.2. Techniques of overfitting avoidance

In deep learning, overfitting can be a significant challenge due to the complexity of neural network architectures and the large number of parameters involved. Here are some specific techniques to avoid overfitting in deep learning models:

A, Batch Normalization: Batch normalization normalizes the activations of each layer, which can help prevent the activations from becoming too large or too small. This can have a regularizing effect and reduce overfitting.

B, Early Stopping: Monitor the performance of your model on a separate validation dataset during training. Stop training when the performance on the validation set starts to degrade,

indicating that the model is starting to over fit. The patience parameter determines how many epochs the training process is allowed to continue without improvement in the validation loss. If the validation loss does not decrease for a certain number of consecutive epochs, the training process is stopped early.

C, Cross-Validation: Use techniques like k-fold cross-validation to evaluate your model's performance on different subsets of the data. This can help you assess how well your model generalizes to unseen data.

D, Dropout: Dropout is a regularization technique where randomly selected neurons are ignored during training. This helps prevent the model from relying too heavily on specific neurons and encourages it to learn features that are more robust.

On this study the researcher, minimize model overfitting to use two methods such as dropout 0.2 means that 20% of layers was dropped and Early Stopping techniques, The dropout size and Early Stopping patience to use in this experiment is define in the above Table 4.1.

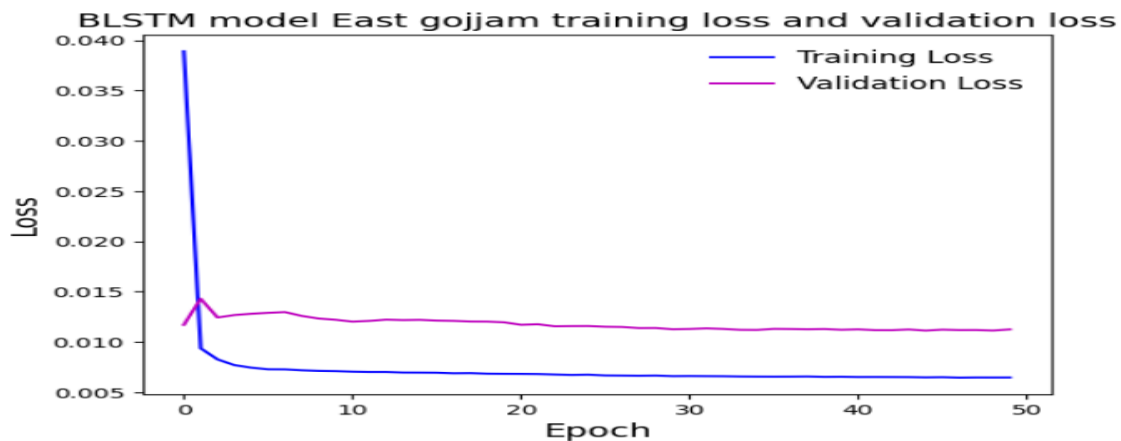


Figure 4. 1: Validation and training for the Bi-LSTM model and number of iterations

Table 4. 2: Bi-LSTM model evaluation matrix

Proposed model	RMSE	MAE	MSE
Bi-LSTM	0.108	0.064	0.012

On the beginning, Bi-LSTM has better performance in prediction. The RMSE value reaches 0.108, MAE 0.064, MSE 0.012. This proposed BLSTM model compare into other state of art in the next section.

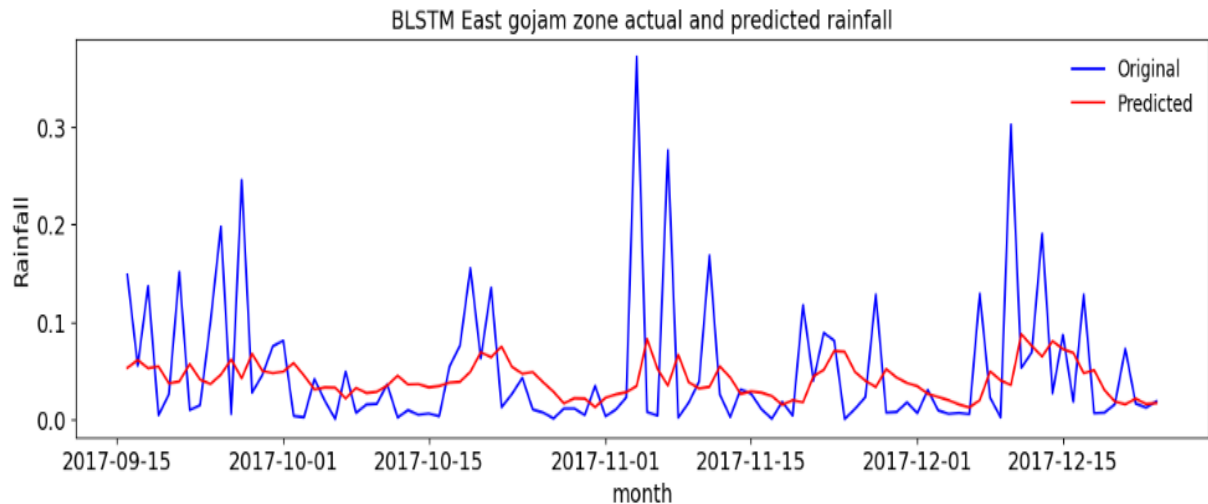


Figure 4. 2: Evaluation of the Bi-LSTM model with testing dataset

4.4.3. Experimental Result of LSTM and CNN-LSTM

After training and testing the east gojjam zone metrological dataset using the BLSTM model, the configuration information of the training platform remains unchanged, and the configuration information of the training platform is used LSTM and CNN-LSTM neural network learning model is trained and analyzed on this dataset.

4.4.4. Experimental Result LSTM model

In addition to using RMSE, MAE and MSE to assess the proposed deep learning based monthly rainfall prediction model, we also assess the model's prediction using data that was not used during the training phase. The training and validation performance of the proposed LSTM model in terms of RMSE, MAE and MSE. On the other hand, we evaluate the predictive performance of the proposed model using a testing dataset. The outcome demonstrates that the proposed model performs well because it mitigates all types of errors.

The results of the proposed (LSTM) based rainfall prediction model for estimating rainfall. The red line on the graph represents the amount of average rainfall predicted by the proposed model, while the blue line represents the actual amount of average rainfall measured by the rain gauge. As a result, the proposed model can be used to predict rainfall of a specific m. The plot shows the actual monthly rainfall values over east gojjam zone collected from NMSA and predicted

rainfall values for 35 years, where the x-axis and y-axis represent month and rainfall values respectively.

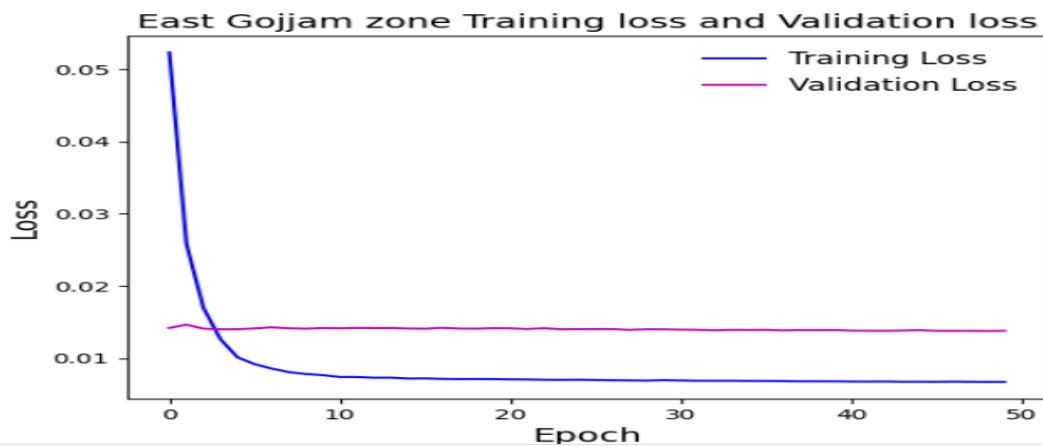


Figure 4. 3: Validation and training for the LSTM model and number of iterations

Table 4. 3: Evaluation of the performance of the proposed LSTM model

Proposed model	RMSE	MAE	MSE
LSTM	0.169	0.083	0.028

As shown in Table 4.2 in LSTM the RMSE value reaches 0.169 and MAE value is 0.083 and MSE value 0.028. In this model good prediction time while poor performance prediction when compared in Bi-LSTM model depend on loss.

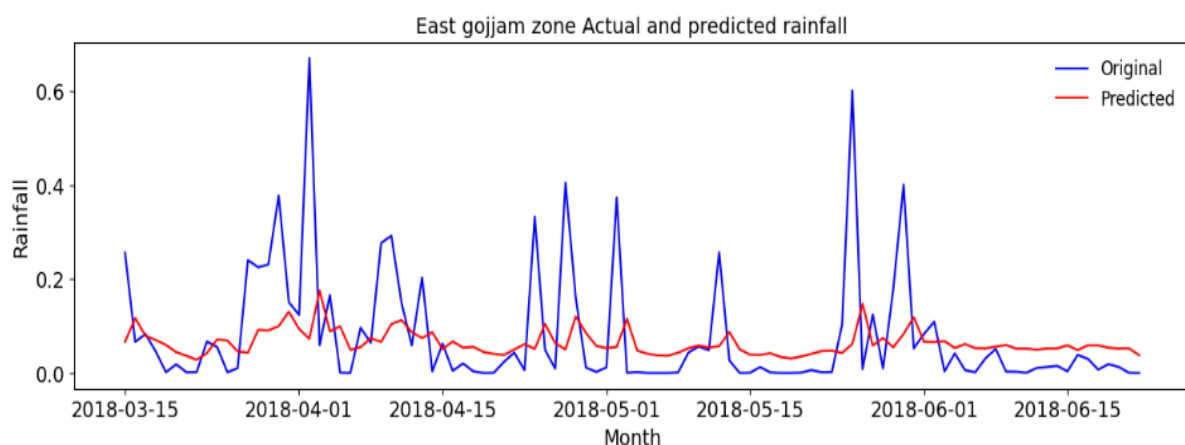


Figure 4. 4: Evaluation of the proposed LSTM model with testing dataset.

4.4.5. Experimental Result CNN-LSTM model

In addition to using RMSE, MAE and MSE to assess the proposed deep learning based monthly rainfall prediction model, we also assess the model's prediction accuracy using data that was not used during the training phase. The training and validation performance of the proposed CNN-LSTM model in terms of RMSE, MAE and MSE.

On the other hand, we evaluate the predictive performance of the proposed model using a testing dataset, and the results are presented in Table 4.4 below. The outcome demonstrates that the proposed model performs well because it mitigates all types of errors, the results of the proposed (CNN-LSTM) based rainfall prediction model for estimating rainfall.

The red line on the graph represents the amount of average rainfall predicted by the proposed model, while the green line represents the actual amount of average rainfall measured by the rain gauge. As a result, the proposed model can be used to predict rainfall of a specific m. The plot shows the actual monthly rainfall values over east gojjam zone collected from NMSA and predicted rainfall values for 35 years, where the x-axis and y-axis represent month and rainfall values respectively.

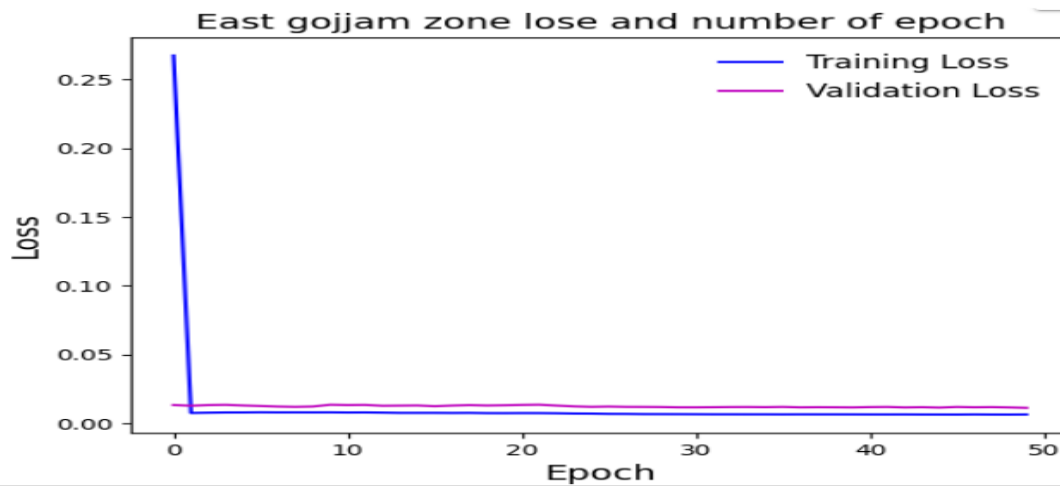


Figure 4. 5: Validation and training loss for the CNN-LSTM model and number of iterations.

Table 4. 4: Evaluation of the performance of the proposed CNN-LSTM model

Proposed model	RMSE	MAE	MSE
CNN-LSTM	0.17	0.085	0.029

The performance evaluation results of CNN-LSTM are shown in Table 4.4. The results reached RMSE with 0.17, MAE is 0.085 and MSE is 0.029. As compared in BLSTM proposed RMSE is greater.

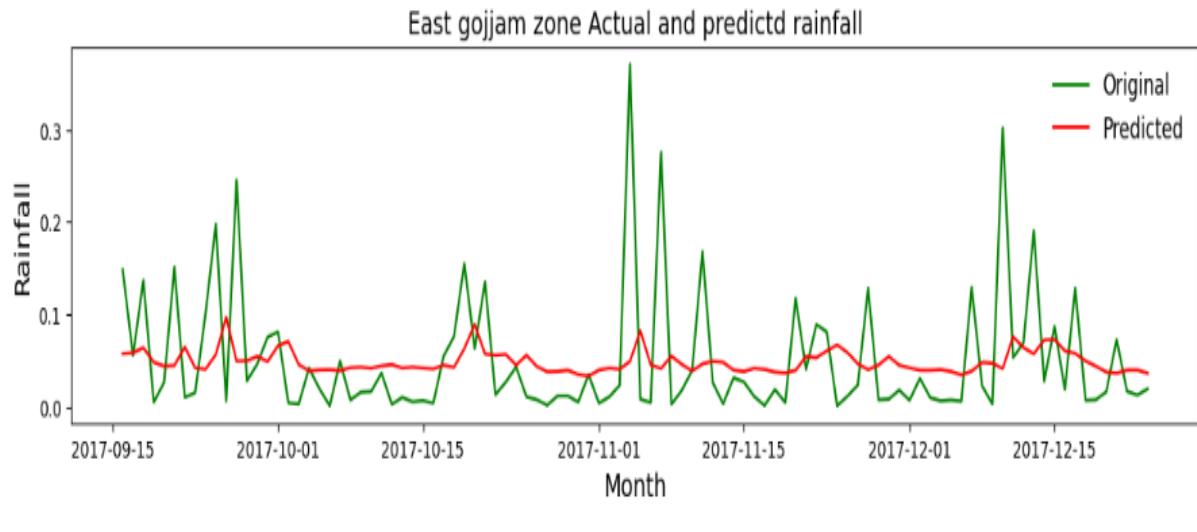


Figure 4. 6: Evaluation of the proposed CNN-LSTM model with testing dataset.

4.4.6. Over all Comparison of Different Algorithms

To validate the performance of the proposed Bi-LSTM model, other state of the art forecasting methods evaluated for comparison LSTM, and CNN-LSTM.

We compared our model with LSTM and CNN-LSTM [12] and other methods on the NMSA dataset, as shown in above table in terms of MSE. The proposed model BLSTM produced lower MSE than LSTM and CNN-LSTM deep learning algorithms. The total amount of rainfall (in mm) for the next month ($t + 1$) is the output of the designed model. Each month weather features are included in each time step. The time step $T(n)$, for example, contains weather parameters for the n^{th} month.

Table 4. 5: Evaluation of the performance of the proposed CNN-LSTM model

Method	MSE	RMSE	MAE	Accuracy
LSTM	0.028	0.169	0.083	97.2%
BLSTM	0.012	0.108	0.064	98.8%
CNN-LSTM	0.029	0.170	0.085	97.1%

On the above table 4.6, Bi-LSTM, LSTM and CNN-LSTM have minimum MSE, and the performance metric is MSE, the best model is Bi-LSTM on this study. After completing the training and test model, the next step is doing real time testing. The test involves different weather dataset. The testing results of the proposed Bi-LSTM model for forecasting monthly rainfall.

4.5. Discussion

The main goal of this research was to perform the complex task of monthly rainfall forecasting system. The developing model is in this study Bi-LSTM. The proposed model forecast monthly rainfall. The test dataset achieving good forecasting results.

In order to scientifically show the forecasting performance of our method, we train and test our proposed method with other predictor on the same dataset, and compare the test results. The RMSE, MAE and MSE of the forecast rainfall were calculate and compared LSTM, Bi-LSTM and CNN_LSTM models. The performance of the developing on the same 12960 metrological dataset is used.

On this study we preprocess the dataset and normalize using MinMax scaling techniques in his method scales the numeric feature values to a fixed range, between 0 and 1 and categorical feature One-Hot Encoding In this method, categorical variables are converted into a binary format where each category becomes a separate feature. For each category, a new binary feature is created where the presence of the category is represented as 1 and absence as 0.

Then split the dataset 0.8% for training and 0.2% for testing. Then the developing model is Bi-LSTM, LSTM and CNN-LSTM they have 30 steps, 2 number of hidden layer and 100 neurons on the first hidden layer and 64 neurons in the second layer.

In addition to use a dropout layer 0.2 on the hidden layer and on the output layer use sigmoid activation function on the hidden layer, tanh activation function on the output layer, learning rate is 0.001, optimizer also Adam, number of epoch (iteration) is 100, batch size also 64, verbose is 2 to use on this experiment depend on grid search and obtained deferent experimental result.

The experimental results as shown that on the Bi-LSTM improved good performance for prediction and forecasting. The MSE value reaches 0.012, MAE value reaches 0.064, and RMSE value reaches 0.108 with prediction time 4 sec 32ms/step.

CNN-LSTM model has a MSE value of 0.029, MAE 0.085 and RMSE of 0.170, with prediction 1 sec 12ms/step. In addition, the LSTM model has a MSE value of 0.028, MAE 0.083 and RMSE of 0.169 with 0 sec 4ms/step prediction time. The experimental results revealed that BLSTM was able to achieve the smallest MSE values, when compared with CNN-LSTM and LSTM.

In general, our proposed method in this study Bi-LSTM achieves 98.8% model accuracy in forecasting monthly rainfall, and its performance is slightly better others in state of art such as CNN-LSTM and LSTM model.

CHAPTER FIVE

5. CONCLUSION AND RECOMMENDATION

5.1. Conclusion

Variations in the global climate and their effects on economic growth have raised concerns throughout the world in recent decades. The issue is particularly serious in developing nations in Africa, Asia, and Latin America, where catastrophic droughts and floods that alternate have been a recurring source of extreme economic misery.

We develop deep learning model to forecast monthly rainfall for east gojjam and used 12960 metrological dataset from Bahir dare stations one-hot label encoding to b used and use Grid search approach to select the best parameters for designing a model.

In our experiment, we expected that CNN-LSTM have best performance because of it is hybrid, However, BLSTM models have high prediction performance in compeer with CNN-LSTM.

After identifying the best hyper-parameters, **Bi-LSTM** with **80:20** train test split ratio achieved MSE 0.012 and 98.8% accuracy for forecasting monthly rainfall. Finally, we feed unseen data to the proposed Bi-LSTM model and forecast the monthly rainfall.

5.2. Recommendation

The primary aim of the study is accomplished by the creation of a rainfall prediction model for the East Gojjam Zone utilizing Bi-LSTM. These upcoming research are recommended for additional research in the domain area based on the study's findings.

- ❖ The developed model has used meteorological data with only seven features. However, future researches should expand the scope by including other features such as location, wind direction etc.
- ❖ The data used in this study is monthly weather data using a single zone. Using two or more zone or region daily data collected at large scale, to predict of both daily, monthly and annual rainfall may provide better results.

References

- [1] M. F. Bekele, M. Y. Tesfaye, M. L. Bekele, and A. T. (PhD), “National Meteorological Agency REVISED METEOROLOGICAL STATION NETWORK MASTER PLAN FOR 2021-2030,” pp. 1–118, 2020.
- [2] S. Salma and A. Ashwitha, “Hybrid Cnn-Lstm Model: Rainfall Analysis and Prediction for Karnataka Region,” *J. Theor. Appl. Inf. Technol.*, vol. 100, no. 22, pp. 6715–6716, 2022.
- [3] P. Kanchan, “Rainfall Analysis and Forecasting Using Deep Learning Technique,” *J. Informatics Electr. Electron. Eng.*, vol. 2, no. 2, pp. 1–11, 2021, doi: 10.54060/jieee/002.02.015.
- [4] K. Abhishek, A. Kumar, R. Ranjan, and S. Kumar, “A rainfall prediction model using artificial neural network,” *Proc. - 2012 IEEE Control Syst. Grad. Res. Colloquium, ICSGRC 2012*, no. February 2016, pp. 82–87, 2012, doi: 10.1109/ICSGRC.2012.6287140.
- [5] 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.
- [6] R. Nagalakshmi, M. Usha, and R. Ramanathan, “Application of Data Mining Techniques in Maximum Temperature Forecasting: A Comprehensive Literature Review,” *Int. J. Adv. Res. Comput. Manag. Stud.*, vol. 7782, no. December, pp. 1–9, 2013, [Online]. Available: <http://ijarcsms.com/docs/paper/volume1/SplDecember2013/V1I7-0002.pdf>
- [7] M. A. Mandale and B. A. Jadhawar, “Weather forecast prediction : a Data Mining application,” *Int. J. Eng. Res. Gen. Sci.*, vol. 3, no. 2, pp. 1279–1284, 2015.
- [8] A. Saxena, N. Verma, and D. K. C. Tripathi, “A Review Study of Weather Forecasting Using Artificial Neural Network Approach.” [Online]. Available: www.ijert.org
- [9] 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.

- [10] J. Lee, C. G. Kim, J. E. Lee, N. W. Kim, and H. Kim, "Application of artificial neural networks to rainfall forecasting in the Geum River Basin, Korea," *Water (Switzerland)*, vol. 10, no. 10, 2018, doi: 10.3390/w10101448.
- [11] S. A. Legesse, "The outlook of Ethiopian long rain season from the global circulation model," *Environ. Syst. Res.*, vol. 5, no. 1, pp. 1–17, 2016, doi: 10.1186/s40068-016-0066-1.
- [12] Y. Dash, S. K. Mishra, and B. K. Panigrahi, "Rainfall prediction for the Kerala state of India using artificial intelligence approaches," *Comput. Electr. Eng.*, vol. 70, no. May, pp. 66–73, 2018, doi: 10.1016/j.compeleceng.2018.06.004.
- [13] F. Fauzi, R. Erlinda, and P. R. Arum, "Rainfall Forecasting Using an Adaptive Neuro-Fuzzy Inference System with a Grid Partitioning Approach to Mitigating Flood Disasters," *JTAM (Jurnal Teor. dan Apl. Mat.)*, vol. 8, no. 2, p. 520, 2024, doi: 10.31764/jtam.v8i2.20385.
- [14] J. W. Taylor and R. Buizza, "Neural Network Load Forecasting with Weather Ensemble Predictions," vol. 17, pp. 626–632, 2002.
- [15] D. Endalie, "Deep learning model for daily rainfall prediction : case study of Jimma , Ethiopia," vol. 22, no. 3, pp. 3448–3461, 2022, doi: 10.2166/ws.2021.391.
- [16] A. Chavan, M. Honnavarkar, M. Kolekar, and S. Dhavane, "Rainfall Accuracy Prediction Using Deep Convolutional Neural Network and Artificial Neural Network," *IJISSET-International J. Innov. Sci. Eng. Technol.*, vol. 8, no. 6, pp. 282–286, 2021, [Online]. Available: www.ijiset.com
- [17] S. Mehrkanoon, "Deep shared representation learning for weather elements forecasting 1," vol. 179, no. September, pp. 1–2, 2019.
- [18] N. Sundaravalli and A. Geetha, "A Study & Survey on Rainfall Prediction And Production of Crops Using Data Mining Techniques," ... *Res. J. Eng. ...*, pp. 1269–1274, 2016, [Online]. Available: <https://www.academia.edu/download/54341943/IRJET-V3I12284.pdf>
- [19] C. McSweeney, M. New, G. Lizcano, and X. Lu, "The UNDP climate change country profiles," *Bull. Am. Meteorol. Soc.*, vol. 91, no. 2, pp. 157–166, 2010, doi:

10.1175/2009BAMS2826.1.

- [20] E. Egrioglu, U. Yolcu, C. H. Aladag, and E. Bas, “Recurrent Multiplicative Neuron Model Artificial Neural Network for Non-linear Time Series Forecasting,” *Procedia - Soc. Behav. Sci.*, vol. 109, pp. 1094–1100, 2014, doi: 10.1016/j.sbspro.2013.12.593.
- [21] I. M. Hayder *et al.*, “An Intelligent Early Flood Forecasting and Prediction Leveraging Machine and Deep Learning Algorithms with Advanced Alert System,” *Processes*, vol. 11, no. 2, 2023, doi: 10.3390/pr11020481.
- [22] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [23] K. Cho *et al.*, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” *EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*, pp. 1724–1734, 2014, doi: 10.3115/v1/d14-1179.
- [24] “1 , 1 1,” vol. 9, no. 9, pp. 107–117, 2021.
- [25] S. Khajure and S. W. Mohod, “Future Weather Forecasting Using Soft Computing Techniques,” *Phys. Procedia*, vol. 78, pp. 402–407, 2016, doi: 10.1016/j.procs.2016.02.081.
- [26] “ወርሃዊ የአየር ጠባይ ግምገማና እና ትንበያ መጽሔት,” 2023.
- [27] E. Toth, A. Brath, and A. Montanari, “Comparison of short-term rainfall prediction models for real-time flood forecasting,” *J. Hydrol.*, vol. 239, no. 1–4, pp. 132–147, 2000, doi: 10.1016/S0022-1694(00)00344-9.
- [28] A. I. O. Yussuff, M. O. Ayeni, and O. E. Oreoluwa, “Adoption of Deep Neural Network Model for the Prediction of Rain Attenuation,” *Eur. J. Electr. Eng. Comput. Sci.*, vol. 7, no. 1, pp. 70–73, 2023, doi: 10.24018/ejece.2023.7.1.498.
- [29] Z. Fan, W. Li, and K. Chang, “A Bidirectional Long Short-Term Memory Autoencoder Transformer for Remaining Useful Life Estimation,” 2023.

Appendix A: Normalization using MinMax Scaler

```
# scale the dataset with minmaxscaler within the range 0 and 1
scaler = MinMaxScaler(feature_range=(0,1))
df_for_training_scaled_en = scaler.fit_transform(df_for_training_en)
df_for_validation_scaled_en = scaler.fit_transform(df_for_validation_en)
df_for_testing_scaled_en=scaler.transform(df_for_testing_en)
```

```
#split test and train
train_split=round(len(df_en)*0.8)
df_for_training_en=df_en[:train_split]
df_temp_en=df_en[train_split:]
test_split=round(len(df_temp_en)*0.2)
df_for_validation_en=df_temp_en[:test_split]
df_for_testing_en=df_temp_en[test_split:]
```

```
print("df_for_training Shape-- ",df_for_training_en.shape)
print("df_for_validation Shape-- ",df_for_validation_en.shape)
print("df_for_testing Shape-- ",df_for_testing_en.shape)
```

```
df_for_training Shape-- (10367, 7)
df_for_validation Shape-- (518, 7)
df_for_testing Shape-- (2074, 7)
```

✓
0s



```
print("trainX Shape-- ",trainX_en.shape)
print("trainY Shape-- ",testX_en.shape)
```



```
trainX Shape-- (10337, 30, 7)
trainY Shape-- (2044, 30, 7)
```

Appendix B: BLSTM performance metric and prediction time

```
# Predict the values
y_pred_en = lstm_model.predict(testX_en)

80/80 [=====] - 4s 32ms/step

# calculate the accuracy metrics
#loss = lstm_model.evaluate(testY_en, y_pred_en)
mae = mean_absolute_error(testY_en, y_pred_en)
mse = mean_squared_error(testY_en, y_pred_en)
rmse = np.sqrt(mse) # or mse**(0.5)

print("Results of accuracy metrics:")
print("MSE:", np.round(mse, 3))
print("MAE:", np.round(mae, 3))
print("RMSE:", np.round(rmse, 3))

Results of accuracy metrics:
MSE: 0.012
MAE: 0.064
RMSE: 0.108
```

LSTM performance metric and prediction time

```
# Predict the values
y_pred_en = lstm_model.predict(testX_en)

65/65 [=====] - 0s 4ms/step

# calculate the accuracy metrics
#loss = lstm_model.evaluate(testY_en, y_pred_en)
mae = mean_absolute_error(testY_en, y_pred_en)
mse = mean_squared_error(testY_en, y_pred_en)
rmse = np.sqrt(mse) # or mse**(0.5)

print("Results of accuracy metrics:")
print("MAE:", np.round(mae, 3))
print("MSE:", np.round(mse, 3))
print("RMSE:", np.round(rmse, 3))

Results of accuracy metrics:
MAE: 0.083
MSE: 0.028
RMSE: 0.169
```

CNN-LSTM performance metric and prediction time

```
# Predict the values
y_pred_en = lstm_model.predict(testX_en)

64/64 [=====] - 1s 12ms/step

# calculate the accuracy metrics
#loss = lstm_model.evaluate(testY_en, y_pred_en)
mae = mean_absolute_error(testY_en, y_pred_en)
mse = mean_squared_error(testY_en, y_pred_en)
rmse = np.sqrt(mse) # or mse**(0.5)

print("Results of accuracy metrics:")
print("MSE:", np.round(mse, 3))
print("MAE:", np.round(mae, 3))
print("RMSE:", np.round(rmse, 3))

Results of accuracy metrics:
MSE: 0.029
MAE: 0.085
RMSE: 0.17
```

Appendix C: CNN-LSTM model development

```
# Build the CNN-LSTM model
lstm_model = Sequential()
lstm_model.add(Conv1D(filters=10, kernel_size=3, activation='tanh', input_shape=(n_steps, n_features)))
lstm_model.add(MaxPooling1D(pool_size=2))
lstm_model.add(LSTM(128, activation='sigmoid', return_sequences=True))
lstm_model.add(Dropout(0.2))
lstm_model.add(LSTM(64))
lstm_model.add(Dropout(0.2))
lstm_model.add(Dense(1, activation='tanh')) # Output layer with 1 neuron (for regression)
opt = optimizers.Adam(learning_rate=0.01)
lstm_model.compile(loss='mse', optimizer=opt)

# fit network
history = lstm_model.fit(trainX_en, trainY_en, epochs=100, batch_size=64, validation_data=(valX_en, valY_en), verbose=2, shuffle=False)

Epoch 1/100
162/162 - 13s - loss: 0.0203 - val_loss: 0.0273 - 13s/epoch - 83ms/step
Epoch 2/100
162/162 - 8s - loss: 8.2852e-04 - val_loss: 0.0272 - 8s/epoch - 52ms/step
Epoch 3/100
162/162 - 7s - loss: 7.9775e-04 - val_loss: 0.0272 - 7s/epoch - 42ms/step
Epoch 4/100
162/162 - 9s - loss: 7.8978e-04 - val loss: 0.0272 - 9s/epoch - 54ms/step
```