

High Frequency Rainfall Prediction using Machine Learning Approach to Numerical Weather Modelling

by

Afsana Afrin Borna

17101031

Ashfak Ahmed Ani

17101460

Mahir Ashhab

16201099

Shahriar Saleh

17101455

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science & Engineering

Department of Computer Science and Engineering
Brac University
June 2021

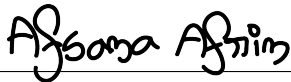
© 2021. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Afsana Afrin Borna
17101031



Ashfak Ahmed Ani
17101460



Mahir Ashhab
16201099



Shahriar Saleh
17101455

Approval

The thesis/project titled “High Frequency Rainfall Prediction using Machine Learning Approach to Numerical Weather Modelling” submitted by

1. Afsana Afrin Borna (17101031)
2. Ashfak Ahmed Ani (17101460)
3. Mahir Ashhab (16201099)
4. Shahriar Saleh (17101455)

Of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science & Engineering on June 2, 2021.

Examining Committee:

Supervisor:
(Member)



Md. Saiful Islam
Lecturer

Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Associate Professor

Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)



Dr. Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

The weather has an impact on almost every aspect of our daily lives. Life would be much easier if we could control the weather. Until then, we will have to settle for trying to predict weather but weather prediction is very unpredictable as even a small change in the surface and atmospheric properties can heavily impact the weather. General weather forecasts, as we all know, are not all that accurate as they attempt to predict the weather conditions of large areas for a large period of time as the tools or mediums used to predict these weather conditions are not accurate enough. They use meteorological and climate data from large areas and integrate those data into different machine learning algorithms. Therefore these weather forecasts fail to be accurate for smaller areas of a large city. As a result, the daily weather forecasts we get from mobile applications or broadcasts are based on larger areas that may be less accurate for a specific area of a city. To solve the less accurate weather prediction problem, this research proposal focuses on constructing a model for precipitation forecasting with the parameters such as Temperature, Wind Speed, Wind Direction, Sea level, and Humidity which are the factors that impact the outcome at the particular spot of interest. This study aims to present a research proposal that combines hyper-accurate forecasts, including hour-by-hour precipitation prediction with customisable information to the street level using supervised machine learning algorithms, Long Short Term Memory (LSTM), Gated Recurrent Units (GRU) and Linear Regression (LR) and feeding historical weather data from the past 40 years. The performance of these algorithms are assessed by comparing their results with each other to find the best algorithm suited for this research. The test results show that the Recurrent Neural Network (RNN) models excel the linear regression model in accuracy and indicate that RNN models can be an effective way for weather forecasting.

Keywords: Weather Forecasting, Precipitation, LSTM, GRU, LR, RNN, Forecasts, Neural Network , Prediction, Atmospheric, Meteorological, Machine learning.

Dedication

We would like to wholeheartedly dedicate this study to our beloved parents, who have been our source of motivation and strength and all the wonderful faculties who guided us every step of the way. Thank you for all the unconditional guidance and support.

Acknowledgement

All praises are to Allah; first of all, we would like to start by thanking almighty Allah, the most gracious and merciful who blessed us to complete this research. In this part, special gratitude to our supervisor Md. Saiful Islam for his valuable guideline, consultation, correction and advice. This thesis would never have been possible without his constructive suggestion, continual encouragement and assistance. Moreover, our profound thanks to our co-supervisor, Shehran Syed, encouraged and invested his valuable time in clarifying our doubts during this thesis work. Finally, we would like to express our gratitude to our family and friends for being a constant source of inspiration.

Table of Contents

Declaration	i
Approval	ii
Ethics Statement	iii
Abstract	iii
Dedication	iv
Acknowledgment	v
Table of Contents	vi
List of Figures	viii
List of Tables	ix
Nomenclature	x
1 Introduction	1
1.1 Overview	1
1.2 Importance/ Usefulness	3
1.3 Motivation	4
1.4 Research Objective	5
1.5 Research Outline	6
2 Literature Review	7
3 Background Analysis	11
3.1 General Supervised Algorithm	11
3.1.1 Long Short-Term Memory units (LSTMs)	11
3.1.2 Gated Recurrent Unit (GRU)	14
3.1.3 Linear Regression (LR)	16

4	Dataset Description	18
4.1	Dataset	18
4.2	Dataset Details	18
4.3	Dataset Pre-Processing	20
4.4	Feature Selection	22
4.5	Train-Test Split	23
5	Proposed Methodology	24
5.1	Work Flow Overview	24
5.2	Precipitation prediction using LSTM	26
5.3	Precipitation prediction using GRU	28
5.4	Precipitation prediction using LR	30
5.5	Testing the Models	31
6	Evaluation and Result	32
6.1	Evaluation of Models	32
6.2	Graphical Analysis	36
7	Conclusion and Future Work	39
	Bibliography	41

List of Figures

3.1	General Architecture of LSTM	12
3.2	General Architecture of GRU	14
3.3	Linear Regression intuition	16
4.1	Duplicate Rows	21
4.2	Heat-Map of Numerical Features	22
5.1	Proposed Work Flow Diagram	25
5.2	Proposed LSTM Architecture	26
5.3	Proposed GRU Architecture	28
5.4	Block Diagram of Proposed Linear Regression Model	30
6.1	Box-Plot of All Continuous Features	33
6.2	Comparison of MAE Scores after Model Implementation	35
6.3	Comparison of R^2 Scores (%) after Model Implementation	35
6.4	Real vs Predicted Rain_1h result of LSTM	36
6.5	Scatter-Plot of Predicted vs Actual Rain_1h using Linear Regression .	37
6.6	Actual vs Predicted Rain_1h using Linear Regression	37
6.7	Actual vs Predicted Rain_1h using GRU	38

List of Tables

4.1	List of Initial Columns in Dataset	19
4.2	Initial Missing Value Count	20
4.3	Final Set of Features Selected	23
5.1	Train-Test Split sample	27
6.1	The Model's MAE Values & R^2 Scores	34

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

σ Sigma Activation Node

ANN Artificial Neural Network

ARIMA Auto Regressive Integrated Moving Average

GRU Gated Recurrent Units

LR Linear Regression

LSTM Long Short-Term Memory

MAE Mean Absolute Error

MSE Mean Square Error

NN Neural Network

RNN Recurrent Neural Network

SVM Support Vector Machine

Chapter 1

Introduction

1.1 Overview

A lot can go wrong if people don't plan for the weather. In order to operate daily activities smoothly and rhythmically, a weather forecast is very important. Weather forecasting is the prediction of what the atmosphere will be like in a specific location by using technology and scientific knowledge. Since the 19th century, people have made every effort to predicate the weather informally and professionally, but not all the research has desired accuracy. Traditional weather forecasting has been built on a foundation of deterministic modelling, which starts with fundamental conditions, feeds them into a computational model, and produces a forecast for the future.

Weather forecasting is a big challenge aiming to predict something that is inherently unpredictable. Forecasting is the utilization of science and technology to estimate weather conditions for a certain geographical region and time. Atmospheric condition predictions are done by compiling quantitative statistics on the present condition of the environment in a specific location and use climatology to forecast how the climate will change. A minor change in the weather parameter in one region might have significant long-term repercussions in another. Any forecasting inaccuracy that occurs will quickly spread and result in further inaccuracies on a greater scale. That's why scientists have been trying to increase the accuracy of the machine learning methods used to predict the weather as much as possible.

Subsequently, Weather forecasting has been transformed and improved with the progress of technology. Currently, sophisticated AI-driven algorithms are used to generate accurate weather forecasts based on historical weather data. There are many traditional and modern approaches for predicting the weather. One of the contemporary approaches to predicting the weather of a particular area is using machine learning Algorithms. These predictions can be made by acquiring radar data, meteorological data, or historical data of a specific location and using different AI-driven or machine learning algorithms.

Nowadays, people are pretty familiar with weather forecasting. Usually, they get this type of weather forecast from their smartphones, as some apps provide weather

updates. People come to know about real-time weather data, including precipitation, temperature, wind speed, visibility, humidity, air pressure, and so on. The primary source of data for this type of mobile app is radar data. But data may vary from model to model.

In this research paper, different analysis on the weather data set, collected from Open Weather Map has been performed. The data set contains the past 40 years' hourly weather data with 19 variables. The data set has been split into training and testing data, and the number of variables also narrowed down to 11. They are temperature, humidity, pressure, wind speed, wind direction cloud coverage, rain_1h and rain_3h. These data are then trained using different hybrid machine learning algorithms, and the results from these algorithms are compared to find the best model with minimum errors and maximum accuracy. Moreover most large-scale weather forecasting models depend on a combination of image and quantitative data from radar and other meteorological sensors. But lightweight numerical models that make accurate weather predictions from just quantitative data of various meteorological features are not as common. That is where the focus of this study is.

This research paper is divided into seven chapters. The chapters are Introduction, Literature Review, Background Analysis, Dataset Description, Implementation of Models, Evaluation and Result, Conclusion and Future Work.

1.2 Importance/ Usefulness

Bangladesh is situated in the tropical monsoon zone, with a climate symbolized by high temperatures, frequent rainfall, occasionally considerable humidity, and discernible seasonal fluctuations. Thus, weather alerts are necessary to safeguard our lives and property. Forecasting depending on temperature and precipitation are vital in agriculture and on a daily basis since heavy rain significantly restricts outdoor activity. Further, railway or plane routes might be adjusted to accommodate for anticipated weather disruptions. Farm workers must make numerous day-to-day decisions depending on climatic conditions, so a more precise prediction might help them choose the best days for seeding or harvesting. Weather irregularity may cause physical damage to crops and soil erosion. Thus, there is no aspect of denying the impact of weather in agricultural fields. Furthermore, industries that are precipitation dependent, such as landscaping or utility firms that need to perform maintenance, can better match labor and resources to anticipated weather occurrences. This inaccurate weather information affects the day to day life of people who live in larger cities of our country. They tend to depend a lot on the daily weather forecasts. It is necessary to make accurate precipitation forecasting to save people's lives by better preparing for an upcoming weather-related event. Moreover, in the weather forecasting industry, timely and accurate information is indispensable, and only precise predictions can precisely depict these changes. Lastly, floods and wildfires do not depend on weather conditions alone, but they strongly influence them. The purpose of precipitation prediction is to deliver details to people and organisations that can help to minimize weather-related harms while increasing community advantages such as life and property security, health and safety of the general population, promoting economic well-being and high quality of life. To sum up, travel preparations, scheduling plans, saving properties, protecting crops, sports scheduling are the benefits that can be secured by having access to accurate prediction.

1.3 Motivation

Our daily life relies heavily on the surrounding weather. As a result, we are always around or with a weather forecasting medium or tool. Weather forecasting had existed since 1861, when the first-ever daily weather forecast was issued in The Times. So, we have been familiar with weather forecasting and its impact on our lives for a long time. Although weather forecasting has been prevalent in our lives for a long time, the accuracy of the forecasts is about 80% when predicting one week's weather forecast. Moreover, a five-day forecast has an accuracy of around 90%, and when predicting ten days or longer forecasts, the accuracy drops to around 50%. [31] This happens because of the sudden change in the properties of the surface. A small change in the surface and atmospheric properties can heavily impact the weather. Furthermore, the weather forecasts are done over large areas, which make predicting the correct weather even more difficult.

Such inaccuracy can cause many problems to the everyday life of people who live in larger cities of our country. They tend to depend a lot on the daily weather forecasts, e.g. if someone wants to travel to a certain area of the city and the weather forecast shows it will be sunny for the rest of the day but after arriving it was raining heavily there. Although few types of research were made in the past on rainfall prediction, these researches did not bring any promising result or impact on weather forecasts. Furthermore, agriculture and farming mainly depend on seasons and weather conditions, as farmers must make many necessary decisions based on weather conditions. Farmers may use weather updates to better acknowledge and trace the growth, status, and quality of their crops, allowing them to make more rational choices. As weather predictions are influenced by location, surface property changes and time, so forecasting based on a large region tends to give less accurate or false predictions, and due to that, farmers may suffer. More precise weather updates based on particular areas may guide them in making significant and potentially costly decisions. The objective is to present a weather forecasting model to overcome the previous insufficiencies.

Our research aims to find a more effective and robust weather forecasting model which will predict the weather more accurately than its previous counterpart.

We will do this by acquiring a rich dataset from a smaller region and then using the dataset to implement different hybrid machine learning algorithms, e.g. LSTM, LR, RNN etc. After that, we will compare the results of these hybrid machine learning algorithms and choose the one that has the best results overall. Considering Bangladesh has a subtropical monsoon climate identified by wide seasonal variations, e.g. rainfall, high temperatures, and high humidity, research on predicting the weather conditions in Bangladesh will be a very thoughtful initiative that has not been done prior.

1.4 Research Objective

Unpredictable weather conditions can have a drastic effect on the people of any country or region. Bangladesh is an agricultural country, relies heavily on predicting weather conditions and being able to take steps accordingly. Moreover, the weather of Bangladesh changes very frequently as it has six different seasons, people's life is highly influenced by these changes. For this reason, accurate weather forecasting has become a necessity. The focus of this research revolves around accurately predicting weather conditions and developing a multi-featured weather forecasting model. The objectives of this research are as follows:

- As changes in weather conditions depend on many small factors, using different factors, e.g. temperature, humidity, precipitation, wind speed, wind degree etc., will help predict the weather conditions more accurately.
- Using a rich and large numerical dataset of a small region. Predicting the weather in a smaller area will produce better outcomes than predicting the weather in a larger area.
- Using different hybrid machine learning algorithms, e.g. RNN, LSTM, Linear Regression etc. and comparing the results to develop the best possible algorithm to predict the weather.
- The collected data will be trained on multiple hybrid machine learning algorithms, e.g. RNN LSTM, Linear Regression etc., to improve the models' precision.
- Comparing the results of these hybrid machine learning algorithms to develop the best possible algorithm to predict the weather.
- There have been previous researches done on weather prediction. But most of their accuracy is not satisfactory. This research's main goal is to predict the weather condition accurately and in the shortest time possible.
- Using the amount of precipitation to predict whether there will be a lot of rain or just a little drizzle. Furthermore, using precipitation data to predict flood risks.
- This research aims to find the most efficient and accurate machine learning model to predict weather updates for a specific region.

1.5 Research Outline

The aim of this research is to predict rainfall for the next one hour with greater accuracy based on 40 years of historical data and present adequate reasoning behind the results that have been generated. Using the best machine learning models which are better suited for our research and provide substantial visual reasoning behind the results.

To begin with, in the first chapter (Chapter 1), overview of the research, its goal and how accurate weather prediction can influence different sectors of life. The research objectives bring into light the goals and objectives of the research in contrast to our country.

Secondly, in (Chapter 2) literature review, the comparable work by other researchers over rain prediction are discussed, outlining the significant results they achieve and the obstacles they encountered while using different methodologies and algorithms. Next in (Chapter 3) background analysis of different machine learning algorithms used in this research and their implementation details are discussed.

In (Chapter 4) the detailed analysis of the dataset is discussed. Details about data collection, preprocessing, feature selection with significant details and visualization on how the dataset is pre-processed before implementing in the algorithms are elaborated.

Furthermore, in (Chapter 5) selected models and their implementation are described. Details and visualizations about the algorithms, workflow, training-testing, and implementation are discussed.

In (Chapter 6) the research results that have been achieved are discussed, The accuracy metrics are visualized with a detailed analysis on them.

Moreover, in (Chapter 7) the results achieved and discussed in chapter 6 are compared among each other finding the best algorithm and the results. Finally, conclusion and future work were discussed in the last chapter (Chapter 8).

Chapter 2

Literature Review

There has been great success in the past and recent times, predicting weather using different machine learning algorithms and neural network architectures, which will help guide us in getting more accurate and faster results than before.

Xiangyan Qing and Yugang Niu [1] used LSTM for solar irradiance prediction and compared it to Persistence Algorithm, Linear Least Square Regression(LR), and multilayered feedforward neural networks applying Backpropagation Algorithm (BPNN). In terms of RMSE, the proposed method was found to be 18.34 percent more accurate than BPNN. A study can be performed using this data and combining it with other quantitative analysis to observe climate and weather conditions changes.

Navin Sharma, Pranshu Sharma, David Irwin, and Prashant Shenoy [2] used LR and Support Vector Model (SVM) on a training dataset of historical solar intensity observations to derive a prediction of solar intensity from a collection of forecasted weather metrics at the IEEE International Conference on Smart Grid Communications (SmartGridComm) in 2011. SVM is 27 percent more accurate than existing forecast-based models that use sky conditions to make predictions, and 51 percent more accurate than basic approaches that merely use the past to make predictions. Xingjian Shi, Zhourong Chen, Hao Wang, and Dit-Yan Yeung [3] built an end-to-end trainable model using the convolutional LSTM (ConvLSTM) to provide an accurate and timely forecast of rainfall intensity in a particular neighbourhood. On a synthetic MovingMNIST data collection, they first compare the ConvLSTM network with the FC-LSTM network. In addition, a new radar echo dataset was created to compare the ConvLSTM model against the state-of-the-art ROVER algorithm using a variety of regularly used precipitation nowcasting metrics. The findings of the studies demonstrate that the ConvLSTM model consistently outperforms both the FC-LSTM and the state-of-the-art operational ROVER method in managing spatiotemporal correlations. ROVER2 can make more precise predictions than ConvLSTM, but it also causes more false alarms and it is less accurate in general. ConvLSTM was discovered to be more accurate in predicting future rainfall patterns, particularly at the border.

Mohamed Akram Zaytar, Chaker El Amrani [4] have used recurrent neural networks (RNN). They have used LSTM a special kind of RNN, as they wanted to use older weather data for more accuracy. The findings indicate that LSTM-based neural networks are comparable with traditional approaches, and that they can accurately determine weather forecasts for the next 24 to 72 hours.

Wander S. Wadman, Aijun Deng, Gopikrishna V. Maniachari and Younghun Kim [5] proposed HyperCast, a deep learning technique combining weather forecasts, produced by a hyperlocal weather forecasting model called Nostradamus and by training a feedforward neural network on both these weather forecasts and measurements, the resulting power forecasts are highly accurate as they exploit both longer-term meteorological insights and shorter-term time series analysis. HyperCast often achieves single digit MAPEs (9.2% on average) and always outperforms the Persistence Forecast in accuracy (12.5% on average).

Aivaras Ciurlionis, Mantas Lukosevicius [6] have used nowcasting algorithms such as extrapolation algorithms (basic translation, step translation, and sequence translation) and a single machine learning algorithm based on convolutional neural networks (CNN). They have compared their used algorithms with persistence algorithms. They have also used Hanssen–Kuiper’s (HK) score. This score describes the performance of a classification model. Here CNN has a higher HK score than all other algorithms used. It has 0.5-45 minutes higher accuracy score than other algorithms. Although the work of Aivaras C. and Mantas L. has a different goal than ours their work has helped us understand which Machine learning algorithms give more accurate weather predictions.

Y.Radhika and M.Shashi [7] compared the performance of Support Vector Machines (SVMs) trained with nonlinear support vector regression and Multi-Layer Perceptron (MLP) trained with Back-Propagation algorithm, while estimating the day’s highest temperature based on the previous n days’ data Five years of weather data were used to build the models. In the case of MLP, the Mean Square Error (MSE) ranges from 8.07 to 10.2, but in the case of SVM, it ranges from 7.07 to 7.56, which concluded that by proper selection of the parameters, SVM performs better than MLP for atmospheric temperature prediction.

Mark Holmstrom, Dylan Liu, Christopher Vo [8] have used a Linear Regression (LR) model, which attempts to forecast high and low temperatures using a linear combination of features. They also implemented a variant of a functional regression method, which seeks for historical weather patterns which are the most relevant to present weather patterns and then forecasts the weather using these previous patterns. They show that linear regression is a low-bias, high-variance model that can be enhanced by increasing the sample size in their research. They further claim that functional regression is a high-bias, low-variance model, implying that the model selection was bad and that the model’s prediction cannot be refined with more data. However, the functional regression model’s bias could be the forecast based on the weather of the previous four or five days rather than the previous two.

Brandan Quinn and Eman Abdelfattah [9] proposed a model to predict rain by using classifiers which are Extra Trees, Random Forests (RF) Logistic Regression (LR), Stochastic Gradient Descent (GD), and Support Vector Machines (SVM) [1]. The three models that achieve the best accuracy among these five are the Extra Trees, The Random forest, and the SVM. Their initial data set consisted of 100,991 records and there are about 142,000 records in the second data collection. The major challenge of their work is to find a classifier that is not affected by the dataset's small number of positive levels.

Seabstain Scher and Gabriele Messori [10] employ Convolutional Neural Networks (CNNs), Persistence and Local Dimension, and Weather Type Clustering in their model. The data they utilize is the GEFS reforecast dataset's second version (Hamill et al,2013). In 60-62 percent of cases, the forecasted full uncertainty outperforms the climatology, and in 64-66 percent of cases, the expected full uncertainty outperforms the climatology.

S. Prabakaran, P. Naveen Kumar and P. Sai Mani Tarun [16] have implemented LR on the collected training data set of 70 years from 1901 to 1970 for each month to forecast rainfall based on many climatic variables such as average temperature and cloud cover. Rainfall is used as the dependent variable in their study, whereas average temperature and cloud cover are used as independent variables. After applying linear regression, to get the error percentage, subtract the desired value from the predicted value from the actual value and multiply by 100. Lastly, the average error percentage was around 7 percent. Moreover, according to researchers, this model may improve further by using multiple regression models.

The research paper written by S. Poornima and M. Pushpalatha [17] presents an Intensified LSTM based Recurrent Neural Network (RNN) to predict rainfall. Rainfall data from the Hyderabad region from 1980 to 2014 is used as a dataset in the forecast procedure. The dataset for training includes rainfall data (34 years from 1980 to 2013), and the trained model is then evaluated using the dataset for 2014. Maximum Temperature, Minimum Temperature, Maximum Relative Humidity, Minimum Relative Humidity, Wind Speed, Sunshine, and Evapotranspiration are the experimental factors employed in this study, with Rainfall as the outcome variable.. Accuracy of the Intensified LSTM model was compared with Holt-Winters, ELM, ARIMA, RNN and Long Short-Term Memory models. As a result, it was discovered that the Intensified LSTM-based precipitation prediction model has an accuracy of 87.99 percent and that the predicted values, with the exception of the peaks, nearly match the actual values.

A.H.M. Rahmatullah Imon, Manos C Roy, S. K. Bhattacharjee [18] used logistic regression for predicting rainfall. Initially clustering and brushing data screening algorithms to discover erroneous data in the data collection and select a segment of data where irregularity is readily obvious. In their research, the spurious observations are about 0.55 percent. They used a logistic regression approach, in which just two values are used: 0 for no rainfall and 1 for rainfall. It means that max-

imum and minimum temperatures, as well as afternoon humidity, have a positive impact on rainfall, but evaporation and morning humidity has a negative impact. Maximum temperature, according to meteorologists, should have a negative impact on rainfall. The research's deviation statistic implies that the data fit the model effectively. Even yet, at the 10percent levels of significance, the Home-Lemeshow statistic indicates that the fit is poor. Then they evaluate for outliers in the data using the Generalized Standardized Pearson Residuals (GSPR) diagnostic. After that, they remove the outliers from the data and re-fit the logistic regression model by using the remaining data. The revised outcome indicates that minimum temperatures and afternoon humidity have a positive impact on rainfall, whereas maximum temperatures, evaporation, and morning humidity have a negative impact. They also feel that eliminating the outlying items resulted in a considerable improvement in model fitting (p-value increased from 0.09 to 0.67) as shown by the Hosmer-Lemeshow statistic. Finally, they investigate the validity of their outfitted model using cross-validation analysis. Their fitted logistic model can accurately predict rain on 562 days out of 590 (95.25 percent) and did not rain on 49 days out of 58 (84.48 percent). As a result, the chance of a misclassification error is only 0.0571. Cohen's Kappa's value is 0.6942, which is a high number for forecasting a climatic variable such as rainfall.

Atik Mahabub, Al-Zadid Sultan Bin Habib [22] have used several regression algorithms, e.g. Support Vector Regression (SVR), Linear Regression, Bayesian Ridge, Gradient Boosting (GB), Extreme Gradient Boosting (XGBoost), Category Boosting (CatBoost), Adaptive Boosting (AdaBoost), k-Nearest Neighbors (KNN) and Decision Tree Regressor (DTR) to predict the weather. They used data from 2012 to 2018 to anticipate the weather in Bangladesh because they wanted to get the best possible outcome for Bangladesh for the most recent weather data. Wind speed, rainfall, humidity, and temperature are the variables used to forecast weather (low and high). After implementing the regression models, they have observed that Decision Tree Regressor (DTR) performed better than other models with a score of 0.37 MAE, 2.09 MSE and 3.78 percent MAP. According to the researchers, with fewer weather parameters, regression-based ML algorithms can forecast weather events with a small margin of error.

Chapter 3

Background Analysis

3.1 General Supervised Algorithm

3.1.1 Long Short-Term Memory units (LSTMs)

Sequence prediction problems are one of the most challenging problems in data science to overcome, and they cover everything from estimating sales to identifying patterns in stock market data, from translating one language to another to anticipating the next word being typed on your phone's keyboard. With modern data science discoveries, it has been determined that LSTM is one of the most powerful and well-known subsets, and it is a specific type of RNN built to recognize patterns in data sequences. LSTM networks are a type of RNN that extends the memory of the original network. RNN is a type of neural network which is developed to interpret any hidden patterns in data while taking into account the sequential nature of the input. Because of its chained like instructions created by loops in the network, it has no trouble connecting past information to the current task. Even so, there's a chance that the gap between useful information from the past and the point in the present when it's needed will widen significantly. Due to the Vanishing Gradient Problem, it may be difficult for RNN to learn to connect the information and detect patterns in the series of data in this situation.

In backpropagation, each iteration of the training procedure updates the weight of the neural network according to the partial derivative of the error function for the current weights. However, the problem emerges when the weight value does not change at all since the gradients will be exceptionally small, and the neural network will be unable to continue training the data. The LSTM was created to tackle this condition and fix the problem.

Sepp Hochreiter and Juergen Schmidhuber introduced LSTM to overcome the vanishing gradient problem in the late twentieth century. Other researchers then refined and popularized it. It's optimal for categorizing, processing, and forecasting time data with variable time lags. The long-term dependency problem is explicitly overcome using LSTM. The default behaviour of the LSTM is to remember information

for a lengthy duration of time. This is because LSTMs store their data in a memory, similar to a computer's memory system, from which it may read, write, and remove data.

There are three types of gates in an LSTM, input gates, which determine whether or not new input is permitted in, the forget gate, which deletes unnecessary information, and the output gate selects whether or not to let it affect the output at the current time step.

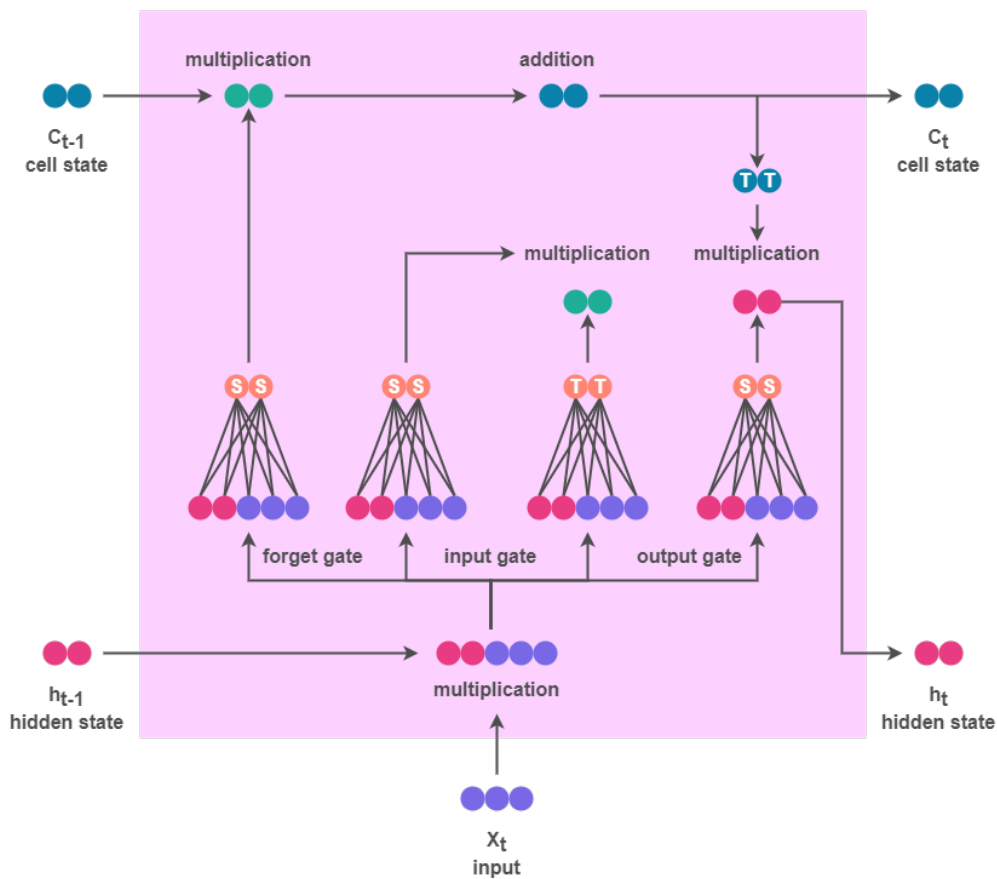


Figure 3.1: General Architecture of LSTM

3.1.1.1 The Input Gate

The input is squashed between -1 and 1 using a tanh activation function. The input is then multiplied element-by-element by the input gate's output. When multiplied element-wise by the input, the input gate is essentially a hidden layer of σ with weighted input values, x_t and h_{t-1} , which outputs values between 0 and 1 and selects which inputs are turned on and off.

3.1.1.2 The Forget Gate

This gate is also a set of σ that is multiplied by C_{t-1} element-by-element to decide which prior states should be remembered, i.e. output near to 1, and which should be deleted, i.e. output near to 0. From this, the LSTM cell learns the relevant context.

3.1.1.3 The Output Gate

The output gate, which has two components, a tanh squashing function and an output σ , is the final stage of the LSTM cell. The output σ is multiplied by the squashed state's C_t , similar to the other gating functions in the cell, to determine which state values are cell output values. As the gating functions regulate what is input, what is remembered in the internal state variable, and finally what is output from the LSTM cell, the LSTM cell is exceedingly adaptable.

$$i = \sigma (w_i [h_{t-1}, x_t] + b_i) \quad (3.1)$$

$$f = \sigma (w_f [h_{t-1}, x_t] + b_f) \quad (3.2)$$

$$o = \sigma (w_o [h_{t-1}, x_t] + b_o) \quad (3.3)$$

i = input gate.

f = forget gate.

o = output gate.

σ = represent sigmoid functions.

w_x = weight for the respective gate(x) neurons.

h_{t-1} = output of the previous LSTM block (at timestamp $t - 1$).

x_t = input at current timestamp.

b_x = biases for the respective gates (x)

Equation 3.1 represents the Input Gate which tells us whether to allow a new input or not. Equation 3.2 represents the Forget Gate which tells us whether the information is redundant or not. And equation 3.3 represents the Output Gate which tells us whether we let the redundant information affect our output at the current time step.

3.1.2 Gated Recurrent Unit (GRU)

The GRU is an improvement over the conventional RNN. Kyunghyun Cho et al. proposed it for the first time in 2014. GRU, like LSTM, uses gates to regulate the flow of information. However, it outperforms LSTM and has a simpler design. GRU also has a unique property in that, unlike LSTM, it does not have a discrete cell state (C_t). There is just one hidden state (H_t). As a result of their compact and streamlined architecture, GRUs are easier to train.

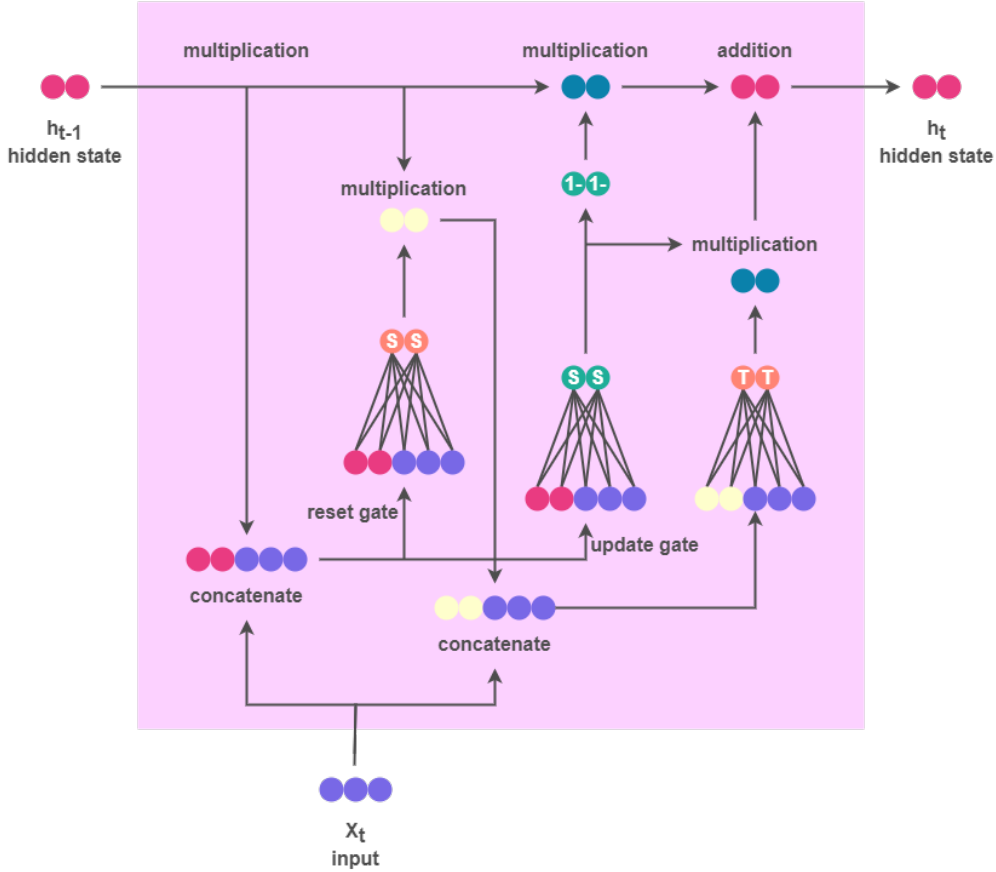


Figure 3.2: General Architecture of GRU

At each timestamp t , it accepts an input x_t and the hidden state h_{t-1} from the previous timestamp $t - 1$. It subsequently generates a new hidden state h_t , which is once again passed to the following timestamp. In a GRU, there are essentially two gates, one for reset and another for update.

3.1.2.1 Reset Gate

The Reset Gate is in charge of the network's short-term memory. This gate is used to decide the amount of knowledge that may be forgotten in the past. The Reset gate's equation is as follows.

$$r_t = \sigma(x_t(u_r + h_{t-1})w_t) \quad (3.4)$$

Because of the σ , the value of r_t will vary from 0 to 1. The reset gate's weight matrices are u_r and w_r .

3.1.2.2 Update Gate

The formula is used to determine the update gate u_t for time step t is,

$$u_t = \sigma(x_t(u_u + h_{t-1})w_u) \quad (3.5)$$

The weight of x_t is multiplied by its own weight u_u when it is inserted into the network unit. h_{t-1} , which stores information from earlier $t - 1$ units and is multiplied by its own weight w_u , in the same way. Both results are combined together, and the result is squashed between 0 and 1 using the σ .

The vanishing gradient problem of a conventional RNN is solved by GRU using an update gate and a reset gate. These two vectors basically decide what data should be transmitted to the output. They are special in that they can be trained to remember data from the past without needing to let it fade away with time or erase irrelevant data. They're one of a kind in that they can be taught to remember information from the past without having to let it fade away with time or erase information that has nothing to do with the prediction.

The Hidden state h_t may be found using a two-step procedure in GRU. The first phase in the procedure is the candidate hidden state, and the second is the hidden state.

3.1.2.3 Candidate Hidden State

$$\hat{h}_t = \tanh(x_t(u_g + (r_t(h_{t-1}))w_g) \quad (3.6)$$

The input and hidden state preceding timestamp $t - 1$ are multiplied by the reset gate output r_t . The candidate's hidden state is the outcome of passing all of this information to the tanh function. When the value of r_t equals 1, the complete information from the preceding hidden state h_{t-1} is taken into account. Similarly, if the value of r_t is 0, then the preceding hidden state's information is completely disregarded.

3.1.2.4 Hidden State

The candidate state is utilised to produce the current hidden state h_t once we know it. GRU controls both the historical information, h_{t-1} , and the fresh information from the candidate state through a single update gate.

$$h_t = u_t(h_{t-1}) + (1 - u_t)\hat{h}_t \quad (3.7)$$

The value of u_t is really important here, as it can vary from 0 to 1.

3.1.3 Linear Regression (LR)

Linear regression assesses the relationship between two different quantitative variables to observe better and understand the data. One of the variables is identified as the independent variable, while the other is identified as the dependent variable. The MSE and MAE are calculated using LR. In order to do that, firstly, line regression has to be calculated. The line regression is derived while the data is being trained. The line regression formula can be best expressed through the following expressions,

$$Y = a + bX \quad (3.8)$$

$$b = \frac{N \sum XY - (\sum X)(\sum Y)}{N \sum x^2 - (\sum x)^2} \quad (3.9)$$

$$a = \frac{\sum Y - b \sum X}{N} \quad (3.10)$$

Here, the dependent variable is Y, the independent variable is X, the intercept is a, and the slope is b.

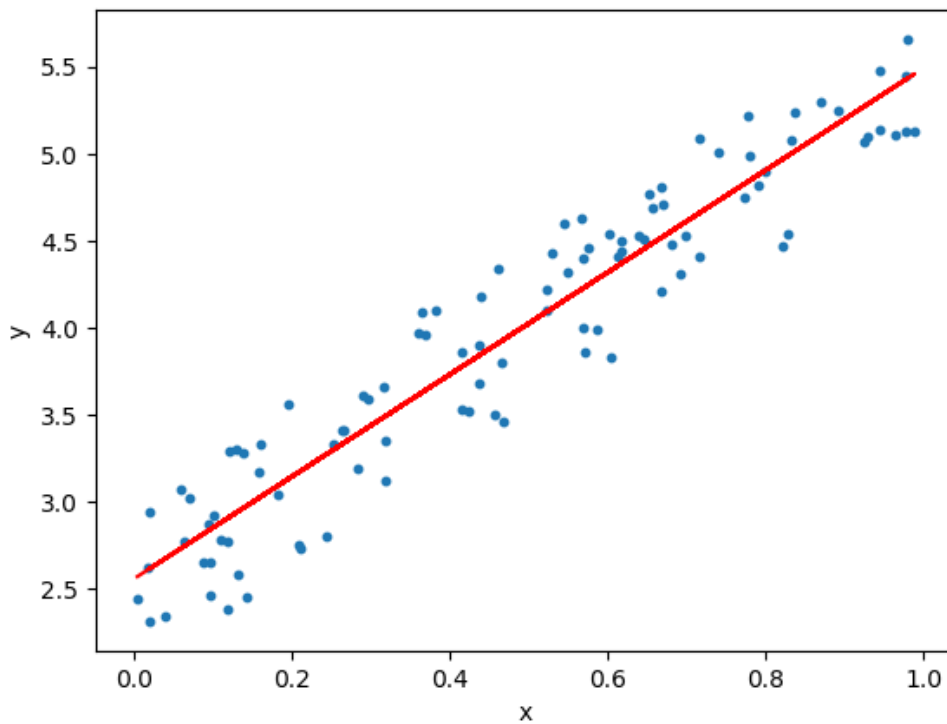


Figure 3.3: Linear Regression intuition

3.1.3.1 Mean Absolute Error (MAE)

The MAE is the simplest straightforward metric for calculating regression error. The absolute value of the distances between the predicted value and the regression line are used to accomplish this. The MAE is calculated using the following equation,

$$MAE = \frac{1}{N} \sum_{i=0}^N |y_i - x_i| \quad (3.11)$$

Here, the number of observations is N, the predicted value is y, and the actual value is x.

3.1.3.2 Mean Squared Error (MSE)

The MSE indicates how near the regression line is to the predicted value. This is accomplished by squaring the distances between the predicted value and the regression line, which represents the error. The goal is that the error is as minimum as possible. The MSE is calculated using the following equation,

$$MSE = \frac{1}{N} \sum_{i=0}^N (y_i - x_i)^2 \quad (3.12)$$

Here, the number of observations is N, the predicted value is y, and the actual value is x.

Chapter 4

Dataset Description

4.1 Dataset

Originally the data was to be collected from an app. The app would be used by a small group of people in a small area. It would've ensured the data is more detailed. Unfortunately, due to the Covid-19 pandemic, the original procedure was halted. Therefore, to collect highly detailed weather data, we acquired 40 years of historical weather data of Tejgaon, Bangladesh, from the open weather map website.

4.2 Dataset Details

The historical weather dataset consists of a total of 25 columns and 369497 entries. Among 25 columns, 19 columns are directly or indirectly related to weather.

Column Name	Count	Data Type
Dt	369497	Int64
Dt_iso	369497	Object
Timezone	369497	Int64
City_name	369497	Object
Lat	369497	Float64
Lon	369497	Float64
Lat	369497	Float64
Temp	369497	Float64
Feels_like	369497	Float64
Temp_min	369497	Float64
Temp_max	369497	Float64
Pressure	369497	Int64
Sea_level	0	Float64
Ground_level	0	Float64
Humidity	369497	Int64
Wind_speed	369497	Float64
Wind_deg	369497	Int64
Rain_1h	39978	Float64
Rain_3h	5297	Float64
Snow_1h	0	Float64
Snow_3h	0	Float64
Clouds_all	369497	Int64
Weather_id	369497	Int64
Weather_main	369497	Object
Weather_description	369497	Object
Weather_icon	369497	Object

Table 4.1: List of Initial Columns in Dataset

4.3 Dataset Pre-Processing

From table 4.1, it is visible that there are many missing values in the dataset. The following table 4.2 represents the number of missing values by each column.

Column Name	Missing Values Count
Dt	0
Dt_iso	0
Timezone	0
City_name	0
Lat	0
Lon	0
Lat	0
Temp	0
Feels_like	0
Temp_min	0
Temp_max	0
Pressure	0
Sea_level	369497
Ground_level	369497
Humidity	0
Wind_speed	0
Wind_deg	0
Rain_1h	329519
Rain_3h	364200
Snow_1h	369497
Snow_3h	369497
Clouds_all	0
Weather_id	0
Weather_main	0
Weather_description	0
Weather_icon	0

Table 4.2: Initial Missing Value Count

‘Sea_level’, ‘Ground_Level’, ‘Sonw_1h’ and ‘Snow_3h’ columns do not contain any values at all. These columns were dropped because they would not affect the forecasting results at all. ‘Dt_iso’ here describes the date and time. ‘Dt’, ‘Tiemzone’, these two columns were also dropped as we kept dates and times with respect to UTC. Moreover, ‘City_Name’, ‘Lat’ and ‘Lon’ these columns also don’t contribute to weather forecasting because these values do not change. These columns were also dropped.

This dataset has 369497 rows. However, there should only be 367776 hours between January 1, 1979, and December 14, 2020. This means that there are duplicate values in the dataset. The following figure represents the duplicate rows in the dataset.

index	dt_iso	temp	feels_like	temp_min	temp_max	pressure	humidity	wind_speed	wind_deg	rain_1h	rain_3h	clouds_all	weather_id	weather_main	weather_description	weather_icon
28,142	1982-03-18 14:00:00 +0000 UTC	21.9	24.25	20.6	24.27	1,008	92	2.29	217	NaN	20	40	201	Thunderstorm	thunderstorm with rain	11n
28,143	1982-03-18 14:00:00 +0000 UTC	21.9	24.25	20.6	24.27	1,008	92	2.29	217	NaN	20	40	502	Rain	heavy intensity rain	10n
28,144	1982-03-18 15:00:00 +0000 UTC	21.35	23.96	20.6	23.0	1,008	92	1.55	216	NaN	20	40	201	Thunderstorm	thunderstorm with rain	11n
28,145	1982-03-18 15:00:00 +0000 UTC	21.35	23.96	20.6	23.0	1,008	92	1.55	216	NaN	20	40	502	Rain	heavy intensity rain	10n
28,146	1982-03-18 16:00:00 +0000 UTC	21.39	23.89	20.6	22.98	1,009	92	1.73	235	NaN	20	40	201	Thunderstorm	thunderstorm with rain	11n
...
361,001	2019-12-27 02:00:00 +0000 UTC	15.41	16.13	14.5	16.2	1,015	94	1	50	NaN	0.4	90	741	Fog	fog	50d
361,002	2019-12-27 03:00:00 +0000 UTC	16.14	17.12	14.5	17	1,016	94	1	50	NaN	0.4	90	500	Rain	light rain	10d
361,003	2019-12-27 03:00:00 +0000 UTC	16.14	17.12	14.5	17	1,016	94	1	50	NaN	0.4	90	741	Fog	fog	50d
361,004	2019-12-27 04:00:00 +0000 UTC	16.17	17.16	14.5	17	1,016	94	1	50	NaN	0.4	90	500	Rain	light rain	10d
361,005	2019-12-27 04:00:00 +0000 UTC	16.17	17.16	14.5	17	1,016	94	1	50	NaN	0.4	90	741	Fog	fog	50d

Figure 4.1: Duplicate Rows

From figure 4.1, it is visible that there are 3438 duplicate rows. There may be two rows of information for a given hour. The only difference comes in the 'weather_id', 'weather_main', 'weather_description' and 'weather_icon'. These are categorical features, so these features cannot have an average. The duplicate rows were dropped except one of the duplicate rows at random; otherwise, it would be hard to resampling data per row. 'Weather_main', 'Weather_description' and 'Weather_icon' these columns are also dropped as these are categorical values and do not contribute to the overall forecast.

4.4 Feature Selection

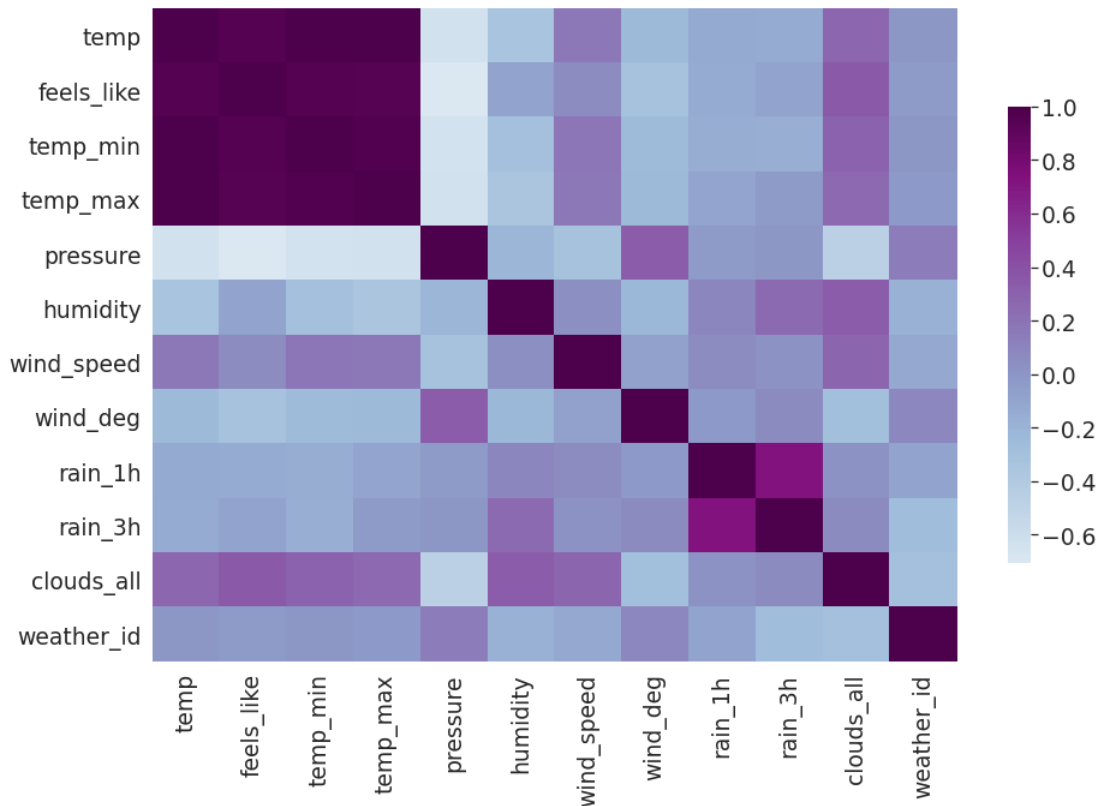


Figure 4.2: Heat-Map of Numerical Features

Figure 4.2 demonstrates a heat-map of all the numerical features of the dataset. From the heat map, it is evident that ‘weather_id’ is actually a categorical column and does not have any noteworthy relation with other variables. So, ‘weather_id’ is also dropped from the dataset. Unsurprisingly, there is also a high correlation between ‘temp’, ‘feels_like’, ‘temp_min’ and ‘temp_max’. As these four variables have a very high correlation between them, thus ‘feels_like’, ‘temp_min’ and ‘temp_max’ were dropped as these variables do not provide new information. Moreover, there were ‘NaN’ values in both ‘rain_1h’ and ‘rain_3h’. It is a normal behaviour in a dataset as there were hours when it did not rain at all. So, these rows which contain ‘NaN’ values were converted to 0.

The following table shows all the features that were selected after conducting feature selection:

Features	Types
Temp	Float64
Pressure	Int64
Humidity	Int64
Wind_speed	Float64
Wind_deg	Int64
Rain_1h	Float64
Rain_3h	Float64
Clouds_all	Int64

Table 4.3: Final Set of Features Selected

4.5 Train-Test Split

The entire dataset was split into two halves. Training and testing took up 80 percent and 20 percent of the time, respectively.

Chapter 5

Proposed Methodology

5.1 Work Flow Overview

In order to build the best possible model, it is essential to choose the correct model space given the training data. Figure 5.1 shows an overview of the workflow. The following are the specifics:

- The raw weather dataset is cleaned and preprocessed. Missing value assignment, outlier detection, and feature scaling are all part of this process.
- Train-Test split is carried out. The ratio is 80:20.
- Three regression models are built and implemented on the training dataset.
- After the training part is done, the models are tested.
- All three models were compared with each other based on their MAE values and R square scores.

In this chapter, details of training all three models are discussed.

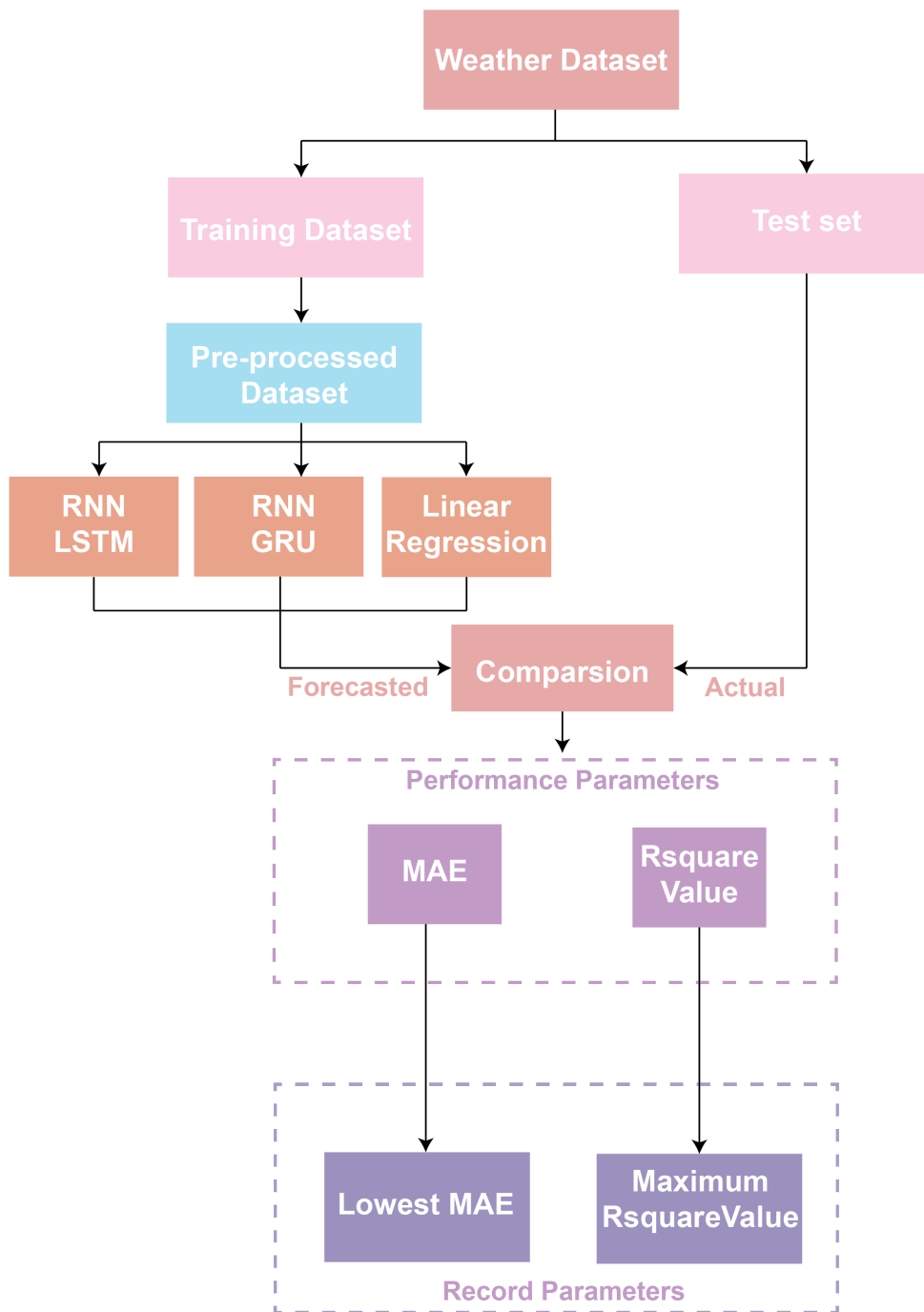


Figure 5.1: Proposed Work Flow Diagram

5.2 Precipitation prediction using LSTM

The dataset contains two variables of rainfall. For this research paper, 'rain_1h' has been used as a target variable. 'Rain_01h' column denotes precipitation volume for one hour in mm. Keras library was used to create the Neural Network Model. RNNs and especially LSTM mainly is very good while making time-series predictions. The goal is to predict the next 24 hours forecast of the precipitation using 96 hours of 'rain_1h' data. In this section, the whole process of implementing the LSTM model is discussed.

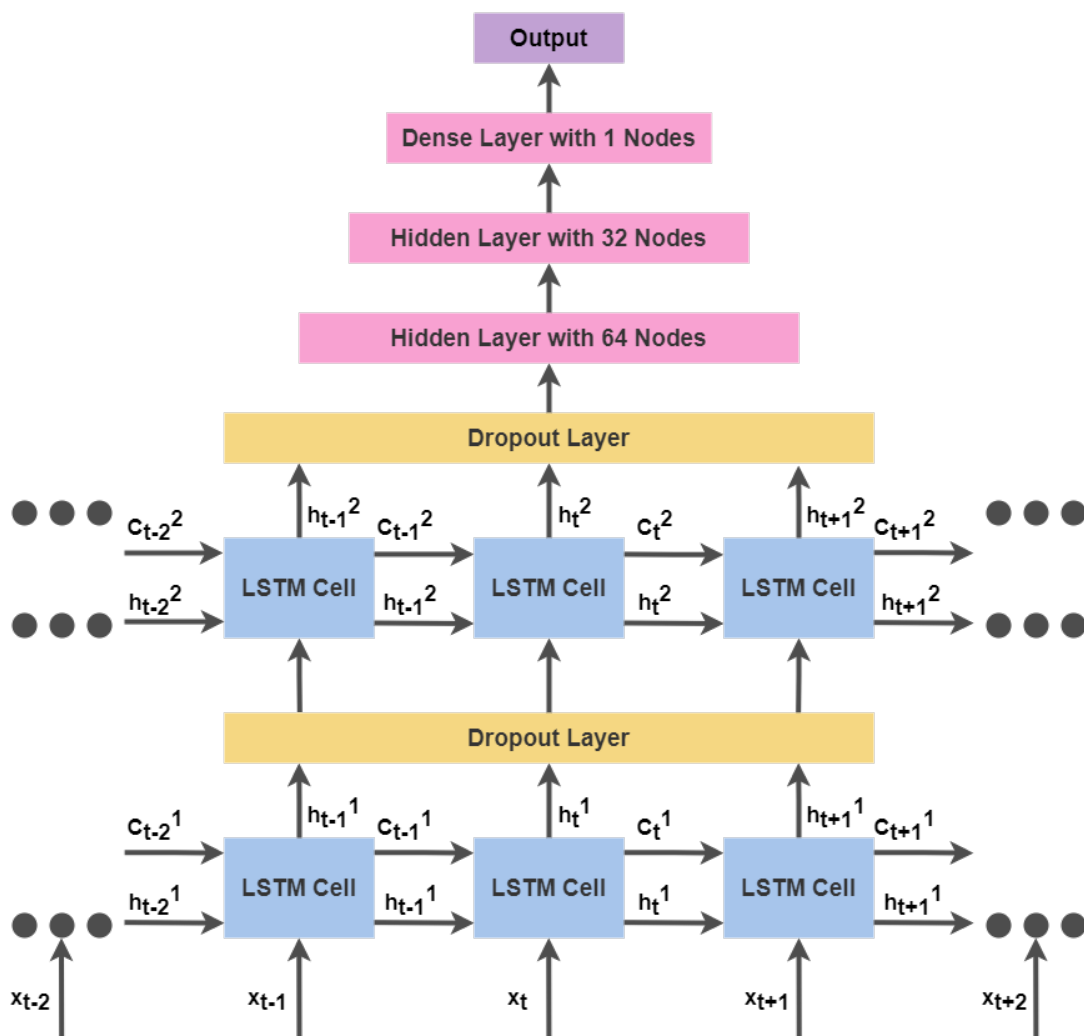


Figure 5.2: Proposed LSTM Architecture

The data was first separated into two groups: training and testing. The test-to-train ratio is 80-20. Then the data were normalized using the famous normalization method, the MinMaxScaler from the scikit-learn library. MinMaxScaler transforms each feature in the dataset and gives them a range of 0 to 1. This is important while implementing the LSTM model because normalization generates new values that preserve the source data’s general distribution and ratios while holding values within a scale applied to all numeric columns throughout the model. Following the normalisation of the data, train and test data set is created.

X (input)	Y (output)
timesteps 0-95	timesteps 96
timesteps 96-191	timesteps 192

Table 5.1: Train-Test Split sample

The above table shows a sample of how the input and output are determined before building the LSTM model.

Over fitting refers to modelling error when data is fitted too well. In other words, the model recognizes noise or random fluctuations in the training data and learns them as notions. The problem is that these notions don’t apply to fresh data, limiting the models’ generalizability. To solving this overfitting issue, a dropout layer is generally added [23]. The output layer consists of 1 Dense layer, and the activation function was set to relu. The optimisation algorithm was set to ADAM during compilation. The word ”optimization” refers to a process for finding which input parameters or arguments to a function provide the function’s minimal or maximal output. For training deep learning models, ADAM is a stochastic gradient descent substitute optimisation method [24]. ADAM builds an optimisation technique for noisy issues with sparse gradients by combining the best elements of the AdaGrad and RMSProp techniques. Moreover, Adam is simple to set up, and the default configuration parameters work well for the majority of problems. The loss function was set to Mean Squared Error, and the metrics were set to MSE and MAE. The number of epochs was set to 20. The term ”epoch” refers to a single cycle of training the neural network with all of the training data. All of the data is used exactly once in an epoch. A forward and backward pass are combined to make one pass. [25] Each epoch consists of one or more batches in which the neural network is trained using a portion of the dataset. The batch size was set to 256, considering the dataset is fairly large containing 40 years of hourly data. The larger batch size generally leads to good overall accuracy. As two dropout layers already included to remove noise, a smaller batch size would not help improve the model’s accuracy.

5.3 Precipitation prediction using GRU

At first, the Keras was library was used to import the GRU model. GRU is one kind of special RNN. GRU and LSTM both are very similar models. Like LSTM, GRU is capable of learning long term dependencies. GRU also uses a gating mechanism similar to LSTM to control the flow of input, such as preserving context across many time intervals. “Rain.1h” is the target variable. The goal is to predict precipitation for the next 24 hours.

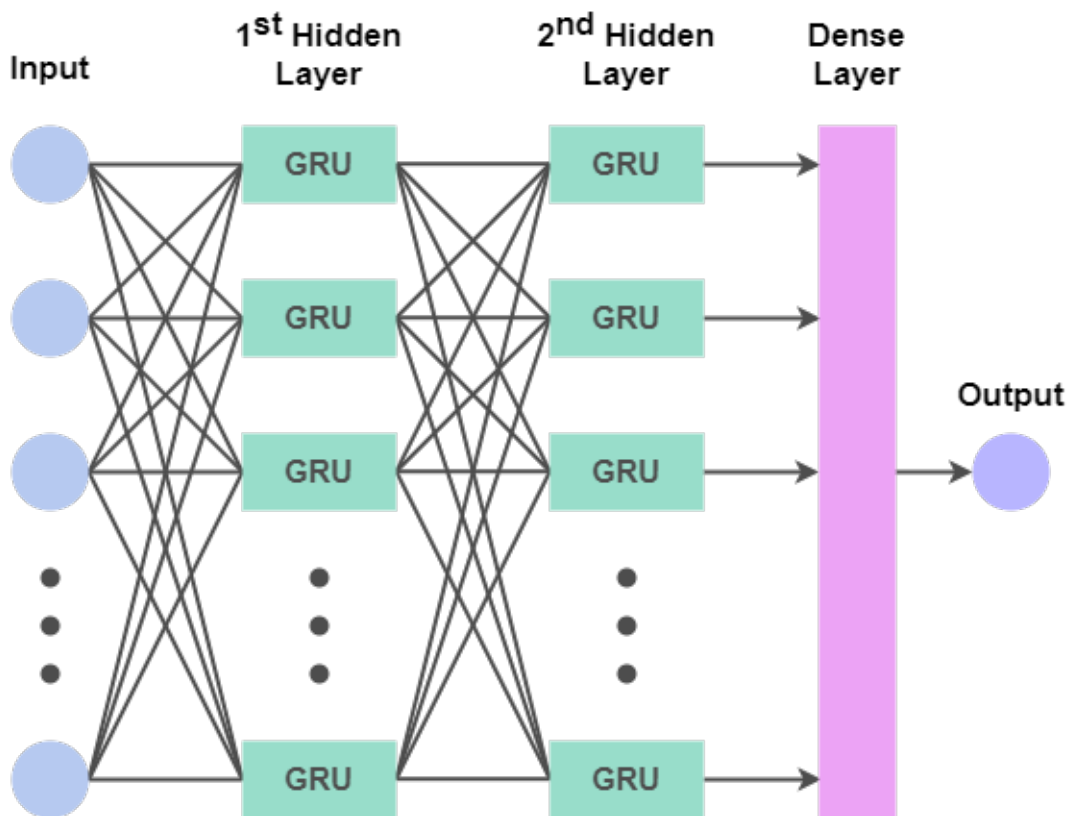


Figure 5.3: Proposed GRU Architecture

The dataset contains 24 weather observations for 24 hours. To forecast precipitation over the next 24hrs, the data was shifted to 24 time steps. After the data shifting, the Input signal and Output signal were feed to neural networks. The input signal is the data frame up to the shifted rows. The output signal is the data frame from the shifted rows. The train-test split was 80% and 20% respectively. As there is a wide range of values in the dataset, the data scaled down to the range of -1 to 1. Neural networks work best for values in the range -1 to 1. The data were scaled using a MinMax scaler from the scikit-learn library. The batch size was set to 256 to utilise GPU to its fullest. A validation set was created to prevent overfitting. In case, if there was no improvement after an epoch, the model's weight would not be saved. The model was initialized with Keras API and two layers of GRU were added to the model. A linear activation function was used on the output to allow the output to take on arbitrary values. The loss function was set to MSE. During the initial few timesteps, the output predicted won't be accurate as it has seen only few timesteps. Hence the accuracy of loss function for few timesteps are not included. The accuracy for the first 50 timesteps was ignored. The model was compiled using RMSProp as the optimiser. RMSprop is an adaptive learning rate method that keeps a moving average of the squared gradient for each weight. Moreover, in order to save checkpoints during training, TensorBoard was used to create callbacks.

5.4 Precipitation prediction using LR

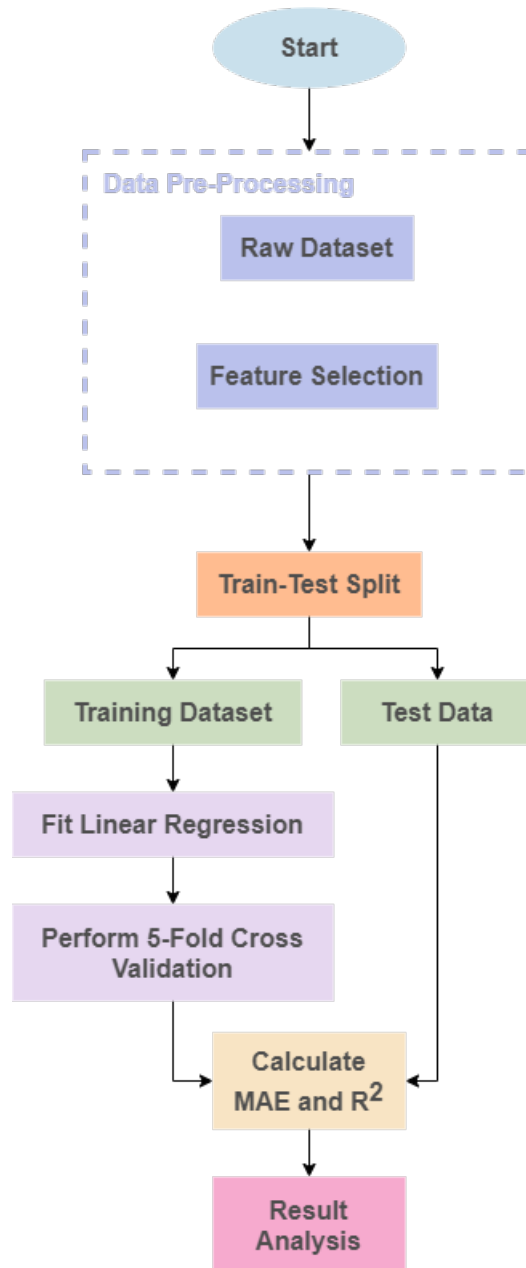


Figure 5.4: Block Diagram of Proposed Linear Regression Model

At first, the dataset was prepared for the Linear Regression Model. The Goal is to predict precipitation on a given hour using data from the previous hour. In this section, the implementation of the Linear Regression model is discussed.

The dataset was divided into train and test set. “Rain_1h” is set to test set as it is the target variable. Train set consists of 'temp', 'pressure', 'wind_speed', 'wind_deg', 'clouds_all', 'rain_1h_1' 6 variable. “Rain_1h_1” column denotes the precipitation data for the previous hour. The train-test ratio is 80-20. The random state is set to 42 in this implementation. Since train test split divides arrays or matrices into random train and test subsets, it is required to utilize a random state number. That is, the model will yield a different result each time it is run without providing a random state; this is anticipated behavior. If a random state Equals some integer, on the other hand, the result of Run 1 will always be the same as the outcome of Run 2, i.e. your split will never be different. It makes no difference if the random state number is 42, 0, 21,... The crucial point to remember is that whenever 42 is used as a random state number, the first-time split result will be comparable. This is useful in the case of reproducible results, so that the same numbers can be consistently viewed whenever some number is assigned to the random state.

5.5 Testing the Models

Now that training of all three models is done, it is time to test all the models and see how accurately the models can predict the precipitation for 1h. Result Analysis and Comparisons are discussed in later sections.

Chapter 6

Evaluation and Result

6.1 Evaluation of Models

Now that training of the models are done and it is time to test the models if it can predict the precipitation of 1h. To predict the precipitation for the next 24 hours, test data was fed to different models. In this section, the test output of different models is compared to check which model performed better while predicting precipitation for the next 24 hours.

The criteria for evaluating and comparing the models will be based on the MAE value and R Square score of the respective models. Mean Absolute Error (MAE) is the sum of the absolute difference between actual and predicted values. The reason to choose MAE as a model evaluation metric is MAE treats all errors equally while MSE gives a larger penalisation to big prediction errors by squaring the errors. [26] Moreover, MSE is very sensitive to outliers while MAE is not sensitive to outliers at all [27]. Figure 6.1 shows the box plot of all continuous features to identify the outliers. It is evident from the figure that there are few outliers in the dataset but these are not abnormal observations. The target variable for this research ‘Rain_1h’ has obvious outliers but as the dataset has records after an hour interval most of the values in “Rain_1h” is 0. So, no value of rain is more prevalent than some value of rain in the box plot. The values above 0 are pushed after the third quartile because 0 is the maximum frequency in the box plot. So, a lot of these values where it has recordings of volume of rain is treated as outliers. Hence, the observations were not dropped from the dataset as valuable data will be lost. As MAE is not sensitive to outliers, it is chosen as the model evaluation measure over MSE. R Square score is calculated for all three models. R Square is a useful metric for evaluating how well a model fits the dependent variables.

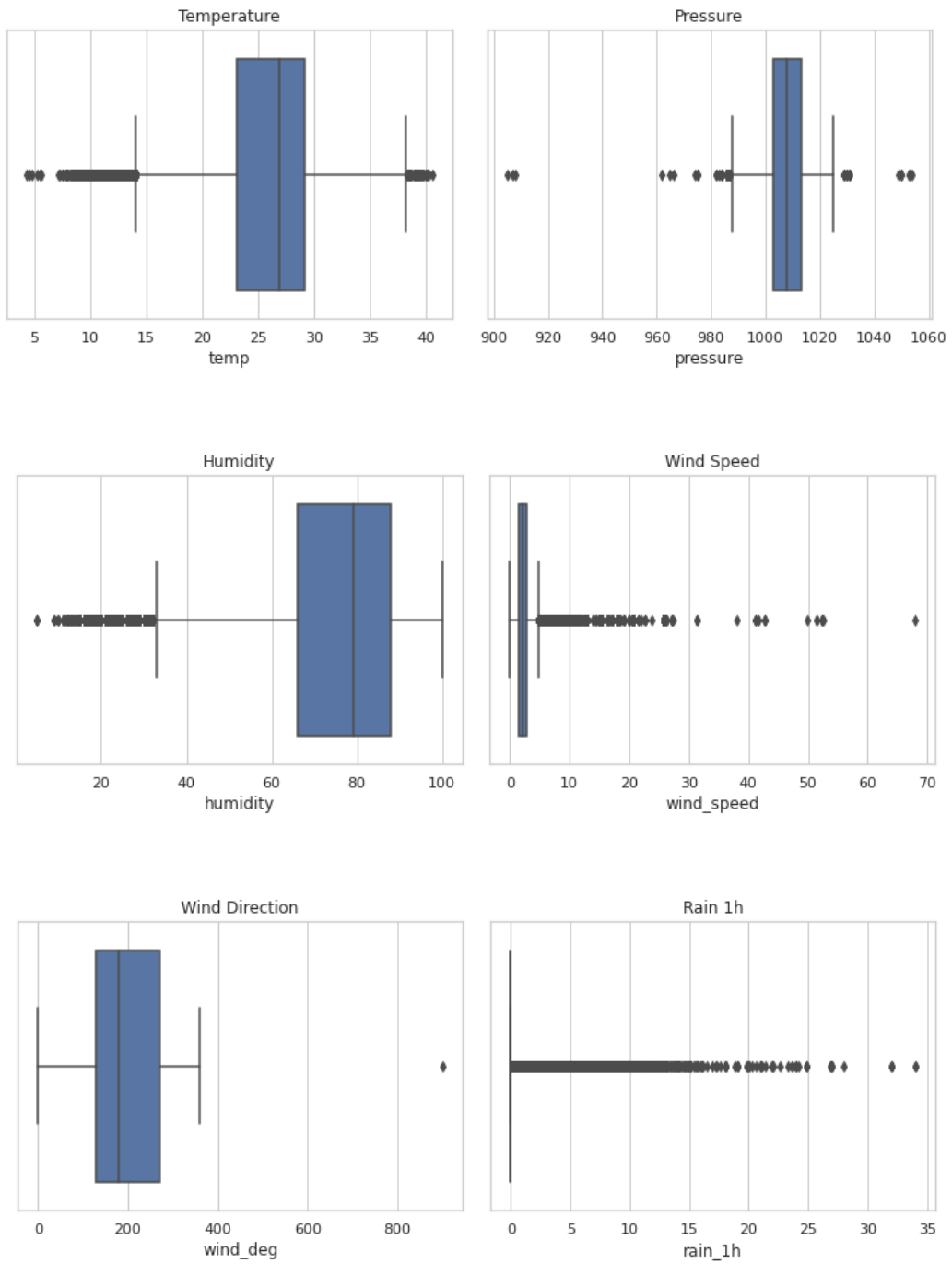


Figure 6.1: Box-Plot of All Continuous Features

The MAE values and R^2 scores of the three models are shown in Table 6.1. From Table 6.1 it is evident that RNN LSTM has performed better than the other two models with an MAE value of 0.02064 on the test set and an R Square score of 0.80303. From the below comparison it is quite clear that both RNN models have similar scores and the range of R square score for these two models are 0.76-0.80. The R square value of RNN LSTM and RNN GRU explains that around 76%-80% of data fit the model. Generally, anything above 50% is a good fit. However, LR performed poorly compared to both RNN models. Linear Regression model achieves a 44% R square score meaning only 44% of the data fit the regression model which is not a good fit. MAE value of Linear Regression model is also higher compared to LSTM and GRU. The higher MAE value tells that the distance between the actual value and the predicted value is higher. Fig 6.2 and Fig 6.3 gives a better visual understanding about the difference between the MAE and R^2 scores of the three models.

Model	MAE	R^2
RNN LSTM	0.02064	0.80303
RNN GRU	0.02135	0.76034
Linear Regression	0.14291	0.44539

Table 6.1: The Model's MAE Values & R^2 Scores

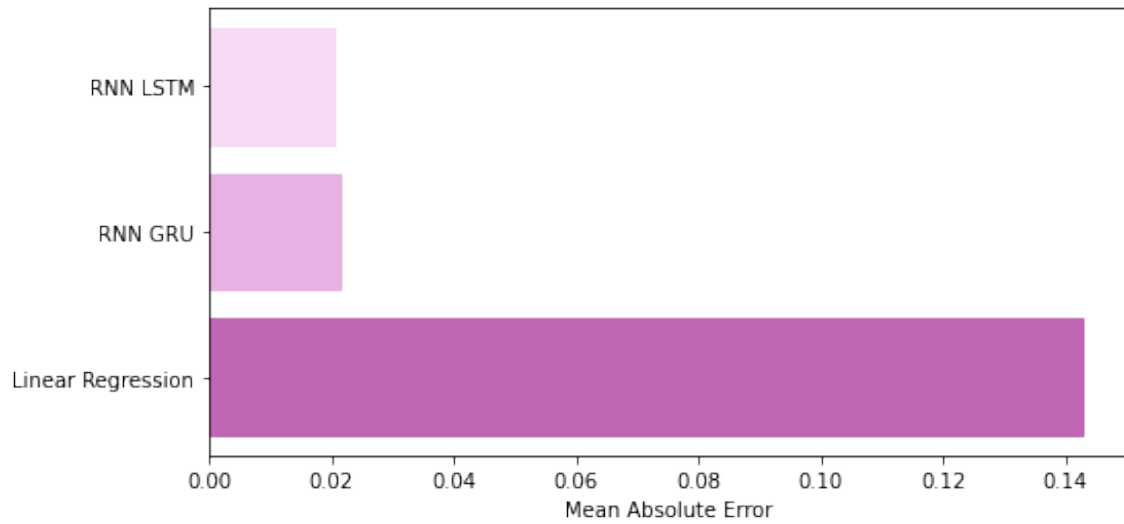


Figure 6.2: Comparison of MAE Scores after Model Implementation

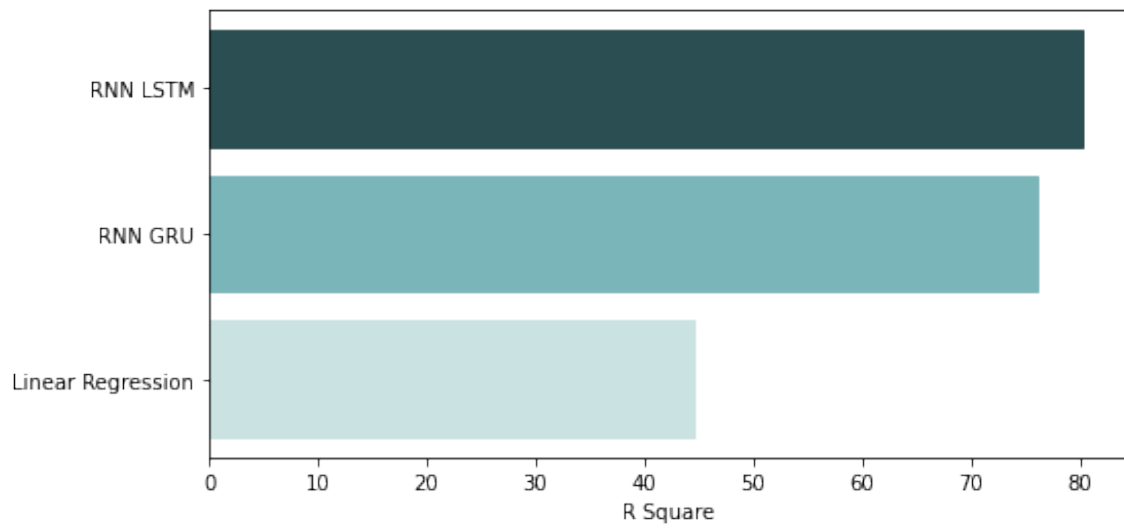


Figure 6.3: Comparison of R^2 Scores (%) after Model Implementation

6.2 Graphical Analysis

Figure 6.4 shows the plot of the prediction of the LSTM model and comparison between “Real Rain_1h” and “Predicted Rain_1h”. From the figure, it is quite clear that the LSTM model performed well enough. It tried to predict the hourly precipitation very accurately. The distance between actual and predicted lines are minimal. It can be explained by the low MAE value from Table 6.1

Figure 6.5 plots the Actual Rain_1h values vs Predicted Rain_1h values. From the figure, it is noted that the model performs better while predicting the low volume of rainfall for 1h. Figure 6.6 shows the visualization of the prediction of precipitation for 1h. It is visible that the Linear Regression model predicts the low volume of rainfall more accurately than the high volume of rainfall. The distance between actual and predicted lines are inconsistent here which explains the higher MAE value from Table 6.1

Figure 6.7 shows the visualization of predicted Rain_1h against actual Rain_1h. It is evident from the plot that the prediction of GRU is very similar to LSTM. It is expected as both models are a special kind of RNN. MAE value and R2 score from table 6.1 also reflects both LSTM and GRU have performed very similarly.

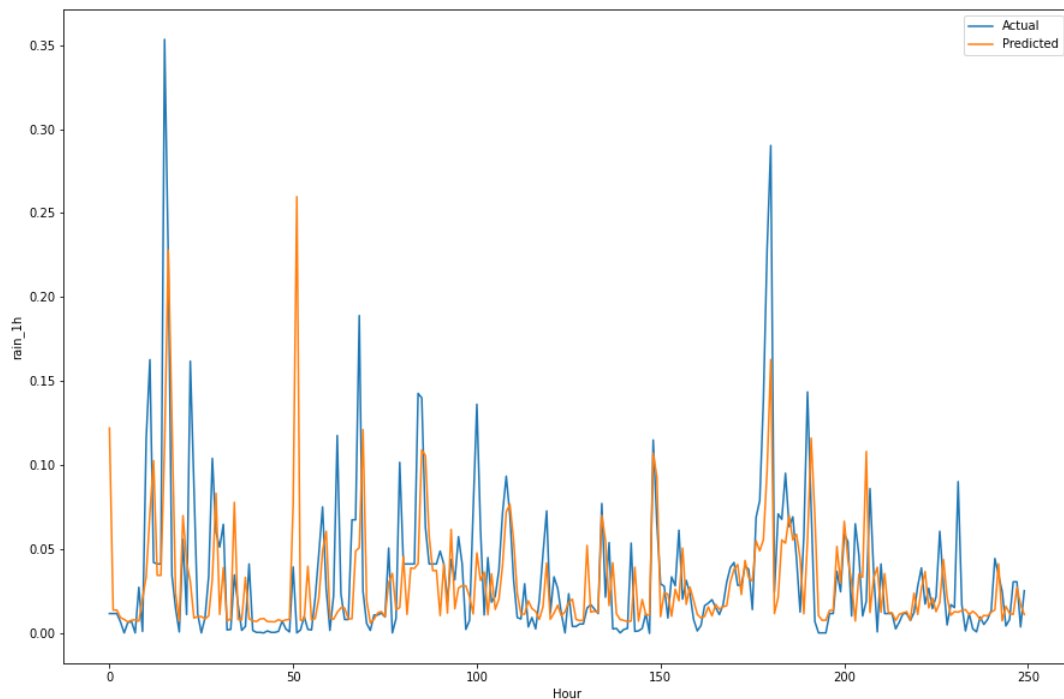


Figure 6.4: Real vs Predicted Rain_1h result of LSTM

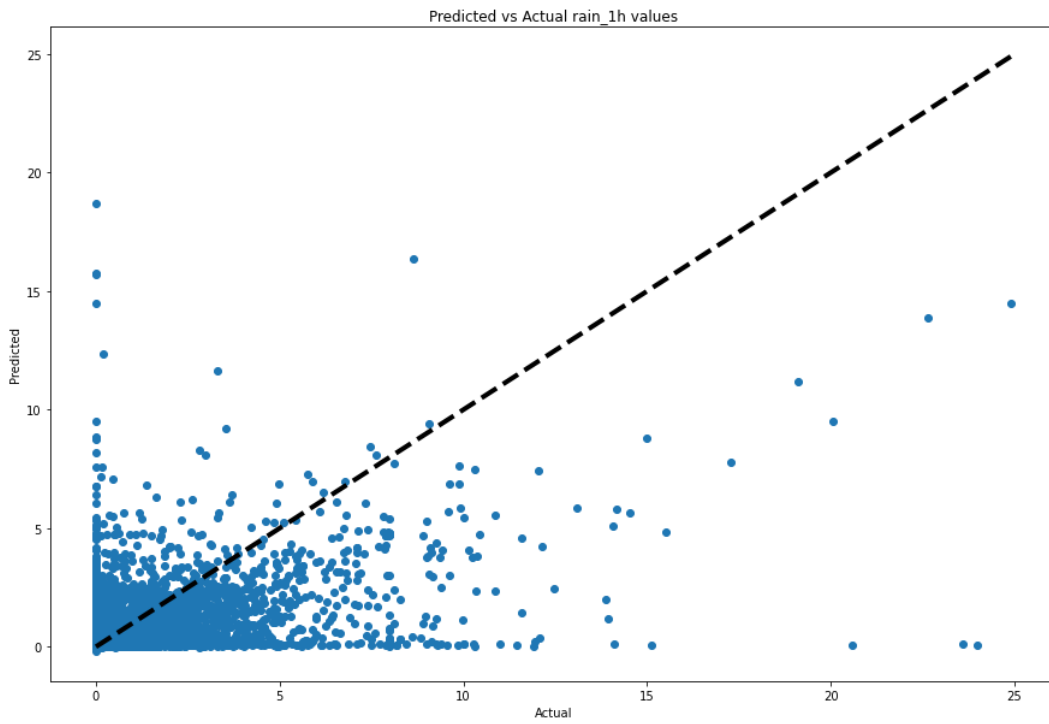


Figure 6.5: Scatter-Plot of Predicted vs Actual Rain_1h using Linear Regression

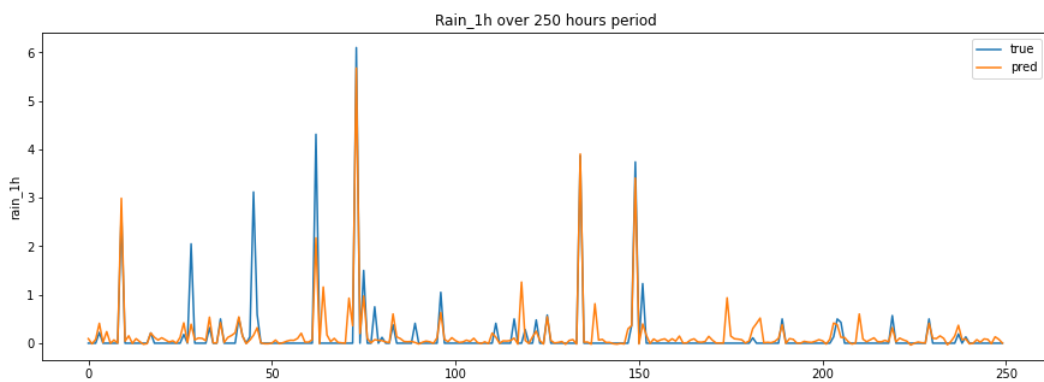


Figure 6.6: Actual vs Predicted Rain_1h using Linear Regression

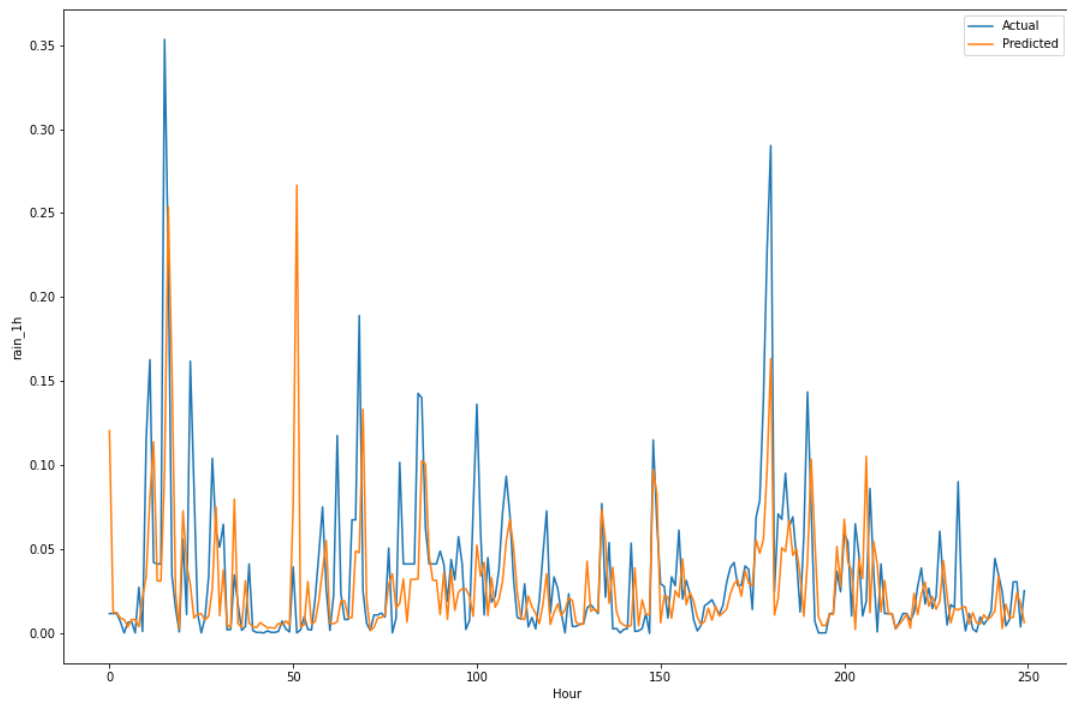


Figure 6.7: Actual vs Predicted Rain_1h using GRU

Chapter 7

Conclusion and Future Work

Bangladesh being an agricultural country and having a tropical monsoon climate, weather plays a crucial part in our everyday life. Accurate weather can help us make important decisions e.g. farmers can pick an optimal day for harvesting, train or plane schedules can be modified according to weather interruptions, businesses that are weather dependent can more accurately match labour and resources to expected weather events and planning our daily activities. In the past, there have been many types of research related to weather forecasting, but most of them were focused on either larger areas or specific weather condition. In our research, we initiate forecasting focusing on different weather parameters that allows precipitation prediction to be much more accurate on the most micro level with down to minute prediction. This research will be beneficial for people in every sector whose life is directly or indirectly influenced by weather effects.

However, there were some limitations to our study. Initially, we aimed to collect the most granular level data therefore we planned to make a mobile app to collect weather data. Our idea was to distribute that mobile app to a small group of people so that from their phone we could collect micro-level weather data of a particular area. However, this plan could not be executed because of the sudden Covid-19 outbreak in Bangladesh as the task needed a lot of fieldwork. Also in the dataset that we received from the Open Weather Map, the classification column was missing therefore we could not predict the probability of the rainfall in the next hour.

For future work, we plan to make a weather app, capable of collecting information about the atmosphere, i.e. humidity, wind speed, precipitation, and temperature at the most micro level. Additionally, as it is observed that the performance of time-series models were incredible thereupon more time-series models such as the ARIMA model will be implemented to find better accuracy in precipitation prediction. Furthermore, we only worked on Tejgaon, Dhaka weather data in this paper but in future, we aim to work with different areas of weather data as well.

In this research paper, our goal was to find how different machine learning algorithms perform to achieve more accuracy with faster and efficient precipitation prediction results. In this study three widely used machine learning algorithms were compared

on test sets to predict the next 1 hour's precipitation. In closing the RNN models performed better than the Linear Regression (LR) model to achieve finer precision of rainfall prediction for the next 1 hour.

Bibliography

1. Qing, X., Niu, Y. "Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM". Energy, Elsevier, vol. 148(C), pp 461-468,2018
2. N. Sharma, P. Sharma, D. Irwin and P. Shenoy, "Predicting solar generation from weather forecasts using machine learning," 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm),pp. 528-533, doi: 10.1109/SmartGridComm.2011.6102379,2011
3. Xingjian Shi, Zhouong Chen, Hao Wang, Dit-Yan. "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting", Proceedings of the 28th International Conference on Neural Information Processing Systems, vol.1,pp 802-810,Dec. 2015
4. Akram, M., El, C. "Sequence to Sequence Weather Forecasting with Long Short-Term Memory Recurrent Neural Networks". International Journal of Computer Applications, vol.143,no. 11, pp. 7-11. doi:10.5120/ijca2016910497,2016
5. Wadman, Wander Deng, Aijun Maniachari, Gopikrishna Kim, Younghun. "Breakthrough accuracy of shorter-term power forecasting using deep learning", Wind Integration Workshop 2017,Oct. 2017
6. Čiurlionis, A. "Nowcasting Precipitation Using Weather Radar Data for Lithuania : The First Results" ,CEUR workshop proceedings, vol. 2147, p. 55-60,2018
7. Yalavarthi, Radhika Shashi, M. "Atmospheric Temperature Prediction using Support Vector Machines" International Journal of Computer Theory and Engineering,vol. 1,no. 1, pp. 55-58, 2009.
8. Holmstrom, M., Liu, D. " Machine Learning Applied to Weather Forecasting" , 2016.[Online]. Available: <https://semanticscholar.org/paper/Machine-Learning-Applied-to-Weather-Forecasting-Holmstrom-Liu/e2ed8aba53b4688808d57a0512496beb3548fc2c/>
9. B. Quinn and E. Abdelfattah, "Machine Learning Meteorologist Can Predict Rain," 2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile

Communication Conference (UEMCON),2019,pp. 0057-0062, doi: 10.1109/UEMCON47517.2019.8992997.

10. Scher, S., Messori, G. (2018). “Predicting weather forecast uncertainty with machine learning”. Quarterly Journal of the Royal Meteorological Society. vol. 144, pp 2830-2841, No. 717.. doi:10.1002/qj.3410.
11. Ingsrisawang, L. Ingsriswang, Supawadee Somchit, S. Aungsuratana, P. Khantiyanan, W. (2008). “Machine learning techniques for short-term rain forecasting system in the northeastern part of Thailand”. Proceedings of World Academy of Science, Engineering and Technology. Vol. 31, pp 248-253.
12. Sunil Ray. “Understanding Support Vector Machine(SVM) algorithm from examples”, 2017. [Online]. Available: <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>
13. Mohit Gupta . “ML — Linear Regression”, 2018. [Online]. Available: [https://www.geeksforgeeks.org/ml-linear-regression/#:~:text=Linear%20Regression%20is%20a%20machine,value%20based%20on%20independent%20variables.&text=Linear%20regression%20performs%20the%20task,given%20independent%20variable%20\(x\)](https://www.geeksforgeeks.org/ml-linear-regression/#:~:text=Linear%20Regression%20is%20a%20machine,value%20based%20on%20independent%20variables.&text=Linear%20regression%20performs%20the%20task,given%20independent%20variable%20(x))
14. Rohit Gandhi. “Introduction to Machine Learning Algorithms: Linear Regression”, 2018. [Online] Available: <https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a>
15. Chris Dinat. “What’s so naïve about naïve bayes?”, 2018. [Online] Available: <https://towardsdatascience.com/whats-so-naive-about-naive-bayes-58166a6a9eba>
16. S. Prabakaran, P. Naveen Kumar and P. Sai Mani Tarun. “RAINFALL PREDICTION USING MODIFIED LINEAR REGRESSION”, ARPN Journal of Engineering and Applied Sciences, vol. 12, no.12,June 2017
17. S. Poornima and M. Pushpalatha. (31 October 2019). “Prediction of Rainfall Using Intensified LSTM Based Recurrent Neural Network with Weighted Linear Units”, Department of Computer Science Engineering, SRM Institute of Science and Technology, Vol. 10, No. 11, pp. 668, October 2019. <https://doi.org/10.3390/atmos1011066>
18. Imon, A. Roy, Manos Bhattacharjee, S. “Prediction of Rainfall Using Logistic Regression” , Pakistan Journal of Statistics and Operation Research,DOI:10.1234/pjsor.v8i3.535 2012.
19. Simeon Kostadinov. “Understanding GRU Network”,2016.[Online].Available: <https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be>

20. Michael Phi. “Illustrated Guide to LSTM’s and GRU’s: A step by step explanation”,2018.[Online]. Available: <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>
21. Shipra Saxena. “Introduction to Gated Recurrent Unit (GRU)”,2021.[Online]. Available: <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-gated-recurrent-unit-gru/>
22. Atik Mahbub , Al-Zadid Sultan Bin Habib (2019). “An Overview of Weather Forecasting Model for Bangladesh Using Regression Based Machine Learning Techniques”,Data Science Pre-press, pp. 1-36,2019
23. Jason Brownlee. “Overfitting and Underfitting With Machine Learning Algorithms”,2016.[Online]. Available: <https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/>
24. Jason Brownlee . “Gentle Introduction to the Adam Optimization Algorithm for Deep Learning”,2017.[Online]. Available: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>
25. Baeldung. “Epochs in Neural Networks”,2021.[Online]. Available: <https://www.baeldung.com/cs/epoch-neural-networks>
26. Songhao Wu. “3 Best metrics to evaluate Regression Model”,2021.[Online]. Available: <https://towardsdatascience.com/what-are-the-best-metrics-to-evaluate-your-regression-model-418ca481755b>
27. Swati Deval. “ Mean Squared Error – Explained”,2021.[Online]. Available: <https://www.mygreatlearning.com/blog/mean-square-error-explained/>
28. Changhyun Choi, Jeonghwan Kim, Jongsung Kim, Donghyun Kim, Younghye Bae, Hung Soo Kim, ”Development of Heavy Rain Damage Prediction Model Using Machine Learning Based on Big Data”, Advances in Meteorology, vol. 2018, Article ID 5024930, 11 pages, 2018. <https://doi.org/10.1155/2018/5024930>
29. ANWAR, Muchamad Taufiq et al. “Rain Prediction Using Rule-Based Machine Learning Approach”, Advance Sustainable Science, Engineering and Technology, vol. 2, no. 1, <https://doi.org/10.26877/asset.v2i1.6019>.May2020.
30. Shreya Agrawal, Luke Barrington, Carla Bromberg, John Burge, Cenk Gazen, Jason Hickey, “Machine Learning for Precipitation Nowcasting from Radar Images”,Cornell University,vol.1,Dec. 2019.
31. How Reliable are Weather Forecasts?. [Online]. Available: <https://scijinks.gov/forecast-reliability/>