2022

A Rainfall prediction model using long short-term neural networks for improved crop productivity: a case of maize planting in Machakos County.

Wangome, Brian Mwathi

Strathmore School of Computing and Engineering Sciences

Strathmore University

A Rainfall Prediction Model Using Long Short-Term Neural Networks for Improved Crop Productivity: A Case of Maize Planting in Machakos County

By

Brian Mwathi Wangome

80677

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Master of Science in Information Technology at Strathmore University.

School of Computing and Engineering Sciences
Strathmore University
Nairobi, Kenya

September 2022

Declaration and Approval

I Brian Mwathi Wangome declare that this research has not been submitted to any other University for the award of a Degree in Masters in Information Technology. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

Brian Mwathi Wangome

……

7th July 2022

Approval

The thesis of Brian Mwathi Wangome was reviewed and approved by the following:

Dr. Joseph Orero,
School of Computing and Engineering Sciences,
Strathmore University

Dr. Julius Bitume,
Dean, School of Computing and Engineering Sciences,
Strathmore University

Dr. Bernard Shibwabo,
Director of Graduate Studies,
Strathmore University

Abstract

Climate variability is a factor that affects crop productivity in Kenya. The unpredictable nature of weather patterns during the traditional long and short rain seasons has resulted in the rains starting earlier or later than expected. This unpredictability results in rainfed agriculture farmers experiencing losses on capital, fertilizers, and labor input and consequently declined agricultural productivity. The decline in food production also poses an existential threat to our nation's food security and farmers' incomes. Weather forecasts are aimed are reducing this uncertainty however, the sparse distribution of synoptic weather stations in Kenya that collect and monitor surface level meteorological conditions makes it hard for the Kenya Meteorological Department to guarantee a high spatial and temporal resolution. Therefore, the current forecast data disseminated to farmers is 'coarse', at the county and town level, which is of less significance to the smallholder farmer since this data does not factor in the topographical nuances within locations. The format of the weather forecasts is also technical for the farmers hence they resort to traditional methods in terms of planning for planting.

The study proposed the use of deep learning techniques to build a rainfall forecasting model that accepted historical weather data and returned forecasted rainfall values in millimeters. The historical weather data was satellite data sourced from NASA's Modern-Era Retrospective Analysis for Research and Applications Version 2 (MERRA-2). The historical data was used to train a Long Short-Term Memory neural network. An experimental approach was used to determine the number of epochs used in training the model and the number of timesteps/days into the future in which the most optimal model would forecast. In this study, the model forecasts 30 days into the future by looking at the past 60 days observed. The 30-day prediction model had a Root Mean Squared Error of 2.45 millimeters. Therefore, given the farmer's Global Positioning System coordinates, the system can fetch past 60-day weather data and forecast the rainfall for the coming 30 days to help farmers to determine when to sow.

Keywords: Rainfall forecasting, Deep Learning, Long Short-Term Memory, Neural Networks

Table of Contents

List of Figures

List of Abbreviations

ANN – Artificial Neural Network

FAO – Food and Agriculture Organization of the United Nations

GDP – Gross Domestic Product

GPS – Global Positioning System

IPCC – Intergovernmental Panel on Climate Change

LSTM – Long Short-Term Memory

MAE – Mean Absolute Error

MERRA - Modern-Era Retrospective Analysis for Research and Applications

MLR – Multi-Linear Regression

MOALFI – Ministry of Agriculture, Livestock, Fisheries and Irrigation

MSE – Mean Squared Error

NASA – National Aeronautics and Space Administration

RMSE – Root Mean Squared Error

RNN – Recurrent Neural Network

SLR – Simple Linear Regression

Acknowledgments

I would like to thank and acknowledge the guidance offered to me by my supervisor Dr. Joseph Orero throughout the thesis proposal and implementation. I also sincerely acknowledge and thank my lecturers Prof. Ismael Ateya, Dr. Vincent Omwenga, Dr. Henry Muchiri, Dr. Allan Omondi, Dr. Dickson Owuor, Dr. Esther Khakata, Dr. Nelson Ochieng, Dr. Bernard Shibwabo and Mr. Nicodemus Maingi who have positively impacted my learning in Strathmore University and helped shaped my career.

Chapter 1. Introduction

1.1 Background

Agriculture is a key sector of Kenya's economy, contributing to over half of the country's Gross Domestic Product-GDP. The sector employs over three-fifths of the working population and accounts for about 65% of the country's exports underscoring its economic significance (World Bank Group, 2018). The crop productivity, however, has been on the decline. The productivity of maize-Kenya's staple food has been on the decline with the World Bank Group (2018) noting that there has been approximately a 15% decline in the production of maize in twenty years from 1994 to 2014. The low productivity in maize and the continued growth in population have contributed to the country's food shortage and pose a risk to overall population nutrition.

Studies that were done by Ochieng et al. (2016) and World Bank Group (2018) highlight that variability in climate conditions such as rainfall and temperature has an impact on maize productivity and consequently on farmers' livelihood. Weather forecasts are intended to help reduce this uncertainty, however, as Ireri (2020) notes, one of the challenges that the Kenya Meteorological Department (KMD) faces is the lack of a well-distributed weather station network throughout the country. This type of distribution is essential in the monitoring of fine-grained climate conditions influenced by factors such as topographical differences (Ireri, 2020). Fine-grained climate conditions are essential for smallholder farmers in planning for farming and yet current weather forecasts span larger areas such as towns and cities.

In another study conducted by Ali-Olubandwa et al. (2011) to find out the challenges faced by small-scale maize farmers in Western Kenya, they found out that a majority lacked knowledge of modern agricultural practices and technical expertise which resulted in less yield. This lack of information was attributed to a high extension staff to farmer ratio. This is compounded by the fact the extension officers lacked adequate financial and transport means to enable the officers to reach and facilitate training and demos to a large number of farmers. The weak link between farmers, extension officers, and researchers culminate in a lack of information for both farmers and extension officers.

In light of the above factors that negatively impact maize productivity, an opportunity exists to help farmers who depend on rainfed agriculture to be able to get insights into the planting season, for instance, determining the appropriate time to plant during the expected long and short rain seasons. Such insight will help maize farmers improve their planning and risk management and consequently get better yield and income from their crops (Hansen & Indeje, 2004).

This study aims to develop a rainfall forecasting tool, that uses historical climatic data extracted for given latitude and longitude coordinates. This information will help farmers make both tactical and strategic decisions concerning maize production. A farmer, through a mobile application, receives recommendations such as the optimal time to sow during a given planting season based on the forecasted rainfall.

To achieve this, the study leverages a deep learning algorithm – Recurrent Neural Networks to develop a rainfall forecasting model. Deep learning is a subject under machine learning that enables computers to understand the world through a layer of concepts with each concept building upon its previous simpler concept. It is through this hierarchy of concepts that computers can learn from experiences (Goodfellow et al., 2016).

1.2 Problem Statement

Climate variability affects agricultural productivity for instance high rainfall positively impacts maize production. Maize yield has been on the decline with the yield per hectare falling from 1918 kg/ha in 1994 to 1628 kg/ha in 2014. (World Bank Group, 2018). This poses a threat to Kenya's food security and economic prosperity. Since weather forecasts are aimed at reducing climatic variability, the Kenya Meteorological Depart lacks the proper distribution of weather stations throughout the country that can enable finer-grained weather forecasts (Ireri, 2020).

A manner in which farmers can get insights into a planting season such as the optimal time to sow maize and risk mitigation measures that need to be taken within the planting season would help farmers reduce losses and take appropriate measures to ensure high yields. Such information is essential to farmers' earnings and national food security. To achieve the above,

the study utilizes remotely sensed information that can be extracted for a given location to develop a context-aware rainfall forecasting model.

1.3 Objectives

1.3.1 General objectives

This study aimed at developing a rainfall prediction model based on historical climate data using a Long Short-Term Memory (LSTM) neural network. Past observed weather data based on a farmer's GPS coordinates are fed into the model to forecast future rainfall values in millimeters. The forecasted rainfall is presented to the farmer in a user-friendly manner on both mobile and web to give an outlook on the amount of rainfall in the coming weeks. This is aimed at helping the farmer to determine the optimal time to plant.

1.3.2 Specific objectives

i. To analyze how rainfall affects maize productivity in Kenya
ii. To analyze current rainfall forecasting techniques
iii. To review how historical climate data has been used in the prediction of rainfall
iv. To develop a rainfall regression model
v. To validate the developed rainfall regression model

1.4 Research Questions

i. How does rainfall affect maize productivity in Kenya?
ii. What are the existing rainfall estimation techniques?
iii. How has historical data been used in the prediction of rainfall?
iv. What is the most suitable technique to develop the rainfall regression model?
v. What is the performance of the developed model?

1.5 Justification

This study aims to address the issue of climate variability by developing a rainfall prediction model using Long Short-Term Neural Networks (LSTM). The LSTM Neural Network technique is used in developing the regression model since its best suited for time forecasting problems. This study aims at forecasting rainfall weeks into the future; hence LSTM is used in developing the rainfall regression model.

The knowledge of the expected rainfall and intensity in the future will enable farmers to know the optimal time to sow maize during a planting season as well as enable them to take mitigation

measures such as stalling farming. This information would be of value for farmers by helping in reducing losses from inputs such as seeds, fertilizer, pesticides, and labor. The model can then be interfaced through a mobile application, this is in a bid to democratize this information to the farmers.

1.6 Scope and Limitation

This study is limited to using online secondary data to develop a prediction model for forecasting the rainfall values for Machakos County. In as much as rainfall prediction can be used for most crops, the scope of this study focuses on maize.

Chapter 2. Literature Review

2.1 Introduction

This chapter discusses the state of farming in Kenya, highlighting the current state of weather forecasting and methods used by farmers to approximate their harvest. Scientific research describing rainfall prediction techniques is discussed as well as existing crop productivity tools existing. Finally, a conceptual framework is developed as a foundation for subsequent chapters.

2.2 Maize farming in Kenya

2.2.1 Maize production

Maize is regarded as Kenya's staple food and accounts for about 36% of consumed calories. Maize is also used as fodder for livestock and in the production of oils underscoring its significance in Kenya's economy and national food security. However, in terms of its value to Kenya's agricultural production, it contributes around 8%, and other agricultural commodities such as tea, cattle, and milk have a higher value (Short et al., 2012).

The United Nations Food and Agriculture Organization (FAO), indicated that maize production from 2015 to 2019 averaged 3.5 million tonnes annually (FAOSTAT, 2021). An initial study by Short et al. (2012) indicated that over six years from 2005 to 2010, the average annual production of maize was 3 million tonnes. A slight increase in maize production from 2010 to 2019 highlights a stagnation in maize production which is underscored by the fact Kenya's maize production has been at a deficit, relying on imports to account for the locally uncatered demand (Ali-Olubandwa et al., 2011). The figure below shows the trend in maize production and yield from 1994 to 2019.

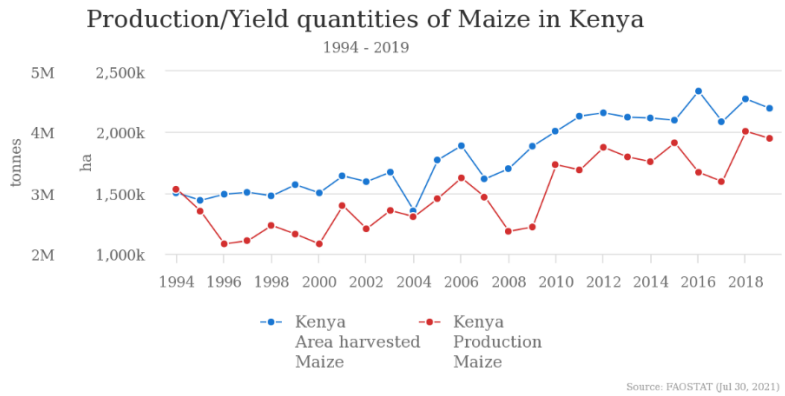

Figure 2.1 Kenya Maize Production and Yield Data (Adapted from FAOSTAT (2021))

Data from Kilimo Open Data, (2021) for the year 2016 indicates maize production was highest in the counties of Uasin Gishu, Trans Nzoia, and Bungoma whereas Wajir, Mombasa, and Wajir counties had the lowest maize production. The majority of the produce is from medium and large-scale farms with 15% of the maize produced by millers and the National Cereals and Produce Board (NCPB) being sold, a large share of the produce (Short et al., 2012).

2.2.2 Effect of climate change on maize productivity

Climate change has resulted in a change in rainfall and temperature, significantly affecting agricultural production in recent years. Maize is a crop whose production is positively impacted by adequate rainfall and temperatures. A study by Belloumi (2014) projects that rainfall levels are expected to drop with "unequal distribution and repartition in time and space" (Belloumi, 2014). A climate change report by the Intergovernmental Panel on Climate Change (IPCC) states that an overall increase in temperature levels has been recorded from a period of 1980 to 2012 indicative of global warming. IPCC estimates that global warming will eventually undermine food security, with maize production in tropical areas projected to be negatively impacted if climate adaptation is not done (IPCC, 2014).

2.2.3 Maize production in Machakos County

Machakos is a semi-arid area that receives an average temperature of 18.9 degrees Celsius and an annual average rainfall of 829mm. The area is characterized by soils with low water retention capabilities and scant organic matter (Mwangi & Mundia, 2022). Machakos county

experiences two rainy spells: long rains starting from March to May and short rains that start from October to December (Dowker, 1963; Velesi, 2018).

According to the Kenya Agricultural & Livestock Research Organization (KALRO), certified hybrid drought-tolerant seeds have shown improved yields over traditional seeds. Drought-tolerant seeds such as DH01, DH02, KCB, KDV 1, 4, and 6, Sungura and Sawa varieties are ideal for areas that receive low annual average rainfall between 400mm and 800mm like Machakos. The seeds have a maturity period of 72 to 75 days and have a yield output of 12 to 14 bags per acre (KARLO, n.d.; Ngotho, 2015).

2.3 Empirical Literature

This section evaluates existing frameworks and systems, analyzing the techniques used to forecast climatological factors such as rainfall and related works.

2.3.1 Weather forecasting in Kenya

Weather forecasting in Kenya is done by the Kenya Meteorological Department (KMD). KMD is tasked with the monitoring, collection, and storage of climate data. Weather stations deployed locally as well as in collaboration with other institutions enable the collection of climate data. KMD consists of a forecasting division that is responsible for the generation and distribution of climate information to a variety of stakeholders, including the general public, the enterprise sector, and maritime services (KMD, 2021).

KMD's climatological department manages three main stations which comprise at least: “700 rainfall, 62 temperature, and 27 synoptic stations”. Synaptic stations have the capability of observing and recording all surface meteorological data such as “rainfall, temperature, wind speed and direction, relative humidity, solar radiation, clouds, atmospheric pressure, sunshine hours, evaporation and visibility” which are most essential in monitoring weather (Masinde et al., 2013).

KMD also consists of an agricultural meteorological division that runs around 13 stations. Data is sent from these stations at a periodic 10-day interval. In addition to the normal meteorological observations, the agricultural department records the “soil temperature, sunshine duration,

radiation, pan evaporation, and Potential Evapotranspiration". KMD uses the collected data to use collected data for five major types of forecasts: daily, 4-day, 7-day, monthly and seasonal forecasts in Kenya's major cities and municipalities. The 4-day, 7-day, and monthly forecasts are in the form of downloadable PDF reports that conceptually summarize recent past, present, and near-future weather patterns (Masinde et al., 2013).

A study by Masinde et al. (2013) highlights the challenges faced by KMD during the creation, distribution, and application processes of their weather products. For instance, the low coverage of weather synoptic stations, 27, within Kenya. The sparse distribution of weather stations means that data generated is 'coarse' having little to no use at local administrative levels. The forecast information distributed by the Kenya Meteorological Department is regarded as being too technical and broad in scope by farmers. Thus farmers resort to using indigenous forecasts that they feel fit with their local traditions and whose format is appropriate to their surroundings (Masinde et al., 2013; Masinde & Bagula, 2011).

2.3.2 Classification of Rainfall intensity

The total rainfall depth for a specified period, expressed in millimeters, is by far the most prevalent rainfall measurement (mm). We might, for instance, wish to know how much rain fell in 1 hour, 1 day, 30 days, or 1 year. The World Meteorological Organization (WMO) classifies rainfall events recorded in millimeters as shown in the figure below.

Type of Event	Daily Rainfall Intensity (mm/Day)
Tiny rain	<1
Light rain	[1, 2)
Low modeerate rain	[2, 5)
High moderate rain	[5, 10)
Heavy rain	[10, 50)
Violent rain	≥50

Figure 2.2 Classification of rainfall intensity Adapted from (Jimeno-Sáez et al., 2021)

2.4 Machine Learning

Intelligence refers to the ability to reason, learn and solve problems through gaining and applying a variety of skills and knowledge. Artificial Intelligence (AI) is a branch of computing that gives computers the ability to mimic human intelligence (Shabbir & Anwer, 2018).

Machine learning (ML) is a subset of AI that focuses on simulating the ability to acquire new and existing knowledge-learning this is achieved through determining correlations and patterns in data (van Klompenburg et al., 2020; Wang et al., 2009). (Alpaydin, 2010) states there are two kinds of ML models: descriptive and predictive, this is dependent on the research problem and questions. A descriptive model acquires knowledge from data whereas a predictive model predicts future occurrence.

In its basic form, machine learning involves four aspects: environment, learning, repository, and execution as described in the figure below

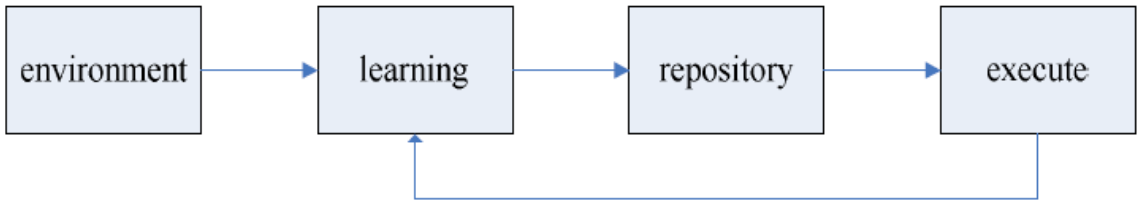

Figure 2.3 Machine Learning Model (Adapter from (Wang et.al, 2009))

The environment refers to external information that is fed into a learning algorithm. The learning process entails processing outside information from the environment and putting it in the repository, a store of knowledge. The repository is a factor that affects the design of a learning algorithm since knowledge can be expressed in various forms such as logic statements, eigenvectors, production rules, semantic frameworks, and networks. Thus, one has to consider the following aspects when considering a repository: “strong inexpression, easy to infer, easy to modify repository” and whether “the knowledge is easy to expand”. The execution stage involves the use of knowledge stored in the repository to accomplish a certain task. Feedback is also gotten at this stage which can be used to allow for further learning (Wang et al., 2009).

Machine learning is considered a pragmatic approach to better estimate crop yield, through the analysis of historical datasets to determine correlations and patterns to derive an output based on previous experience. The parameters of the prediction model during the training phase are gotten from the historical data, this is referred to as the training data. The testing data, historical data that has not been used in training the model, is used to evaluate the performance of the model (van Klompenburg et al., 2020).

2.4.1 Regression

Regression is a supervised machine learning technique that learns from continuous data to derive a function such that Y, a numeric output, is a function of X, the attributes that affect the output. A regression problem would be to forecast the price of a car given inputs such as mileage, brand, year of manufacture, engine capacity, and other relevant information (Alpaydin, 2010).

2.4.1.1 Linear Regression

Linear regression is a technique for predicting a target variable that involves finding the best linear relationship between the dependent and independent variables. This is often referred to as the "line of best fit". The linear model, which is characterized by a straight-line equation, predicts the relationships between two variables. To obtain the line of best fit, the sum of the distance between the actual observations and the line is made to be as small as possible, such that no other location of the line would give a smaller error (Dhumale et al., 2019).

Linear regression is subclassified into Simple Linear Regression (SLR) and Multi-Linear Regression (MLR). SLR involves the use of one independent variable, X, to predict a dependent variable, Y. SLR can be defined as the function:

$$Y = mX + c$$

With m being the slope and c being the intercept of the line.

MLR on the other hand consists of multiple independent variables X_1, …, X_n that linearly determine the dependent variable, Y. It is defined by the function:

$$Y = \beta_0 + \beta_1 X_1 \ldots \beta_n X_n + \varepsilon$$

Where $\beta_0 \ldots \beta_n$ are the intercepts and ε is the error or residual term (Dhumale et al., 2019).

Using the above SLR and MLR functions to determine predicted values y', the error is determined by subtracting y, and y' is given by the equation: $E = Y - Y'$. Since the error may be negative, the value is squared to get rid of the negative, hence the function, $E_T = \sum_{i=1}^{n}(Y_i - Y'_i)^2$ gives the errors for all inputs. Dividing E_T by the number of inputs, n, gives us the Mean Square Error (MSE) denoted as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - Y'_i)^2$$

Minimizing the MSE is what enables one to determine the line of best fit.

2.4.2 Artificial Neural Networks

Deep learning's history dates back to the 1940s with McCulloch and Pitts' (1943) research paper proposing the concept of neural networks (McCulloh & Pitts, 1943). The concept is inspired by the workings of the biological brain, hence deep learning has also been referred to as Artificial Neural Networks.

Currently, deep learning works based on "a more general notion of learning many levels of composition, which can be used in machine learning frameworks that aren't inspired by neuroscience" (Goodfellow et al., 2016). Simple linear models motivated from a neuroscientific standpoint were the forerunners of modern deep learning. The models accept a set of $\boldsymbol{N}$ inputs and associate them with an output $\boldsymbol{y}$. A set of weights $\boldsymbol{w_i, \ldots, w_n}$ is learned by the models which would be used to calculate the output $\boldsymbol{f(x, w) = x_1 w_1 + \ldots + x_n w_n}$ The output $\boldsymbol{y}$, whether positive or negative determines the recognized category.

A neural network is formed through repeated application of the model in a feedforward-manner-the result of one function is passed as an input to the next function. This is achieved through composing functions. For example, if we have four functions, $f1, f2, f3, and\ f4$, they can be linked as $\boldsymbol{f(x) = f_4(f_3(f_2(f_1(x))))}$. This forms the common structure of a neural network. With the first layer being f1-referred to as the input layer, followed by f2 as the second layer, and the others follow respectively. The last layer is known as the output layer. The length of the chain of functions determines the depth of the model, hence the term deep learning.

The training data provides an estimated example of $\boldsymbol{f_i(x)}$ assessed at separate training points with each example, $\boldsymbol{x}$, containing a label $\boldsymbol{y = f_i(x)}$. The output layer's purpose at each point $\boldsymbol{x}$ is to produce a value close to the label $\boldsymbol{y}$. The training data however does not indicate the desired output for the intermediate layers, the learning algorithm does that. It is for this reason the intermediate layers are referred to as hidden layers.

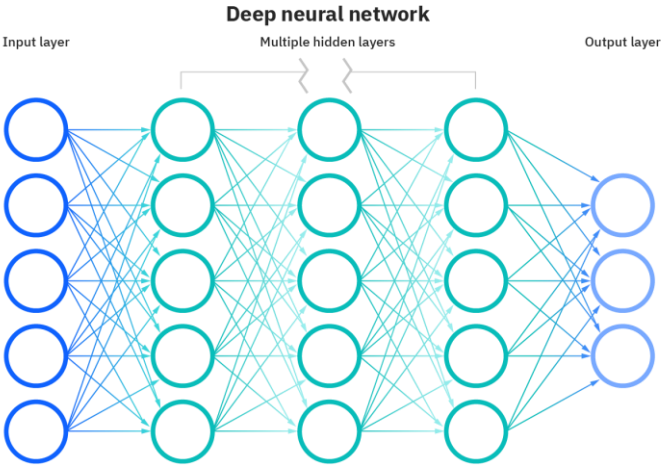

Figure 2.4 Deep Neural Network

2.4.2.1 Recurrent Neural Networks

Recurrent Neural Networks (RNN) are neural networks that can represent a sequence of inputs such as speech data, natural language, and so on. RNNs have a similar structure to a feed-forward neural network or an artificial neural network (ANN), however, unlike ANN, they have cyclic connections as well. A layer's neurons can be connected and even to itself, which was not possible in ANN. Since RNNs are cyclic, prior inputs are utilized to compute outputs at each step, and they have a memory of previous events that they may use to produce subsequent predictions. Previous information can influence the decision-making process thanks to the cyclic links (Prakash et al., 2018). The figure below illustrates the structure of an RNN.

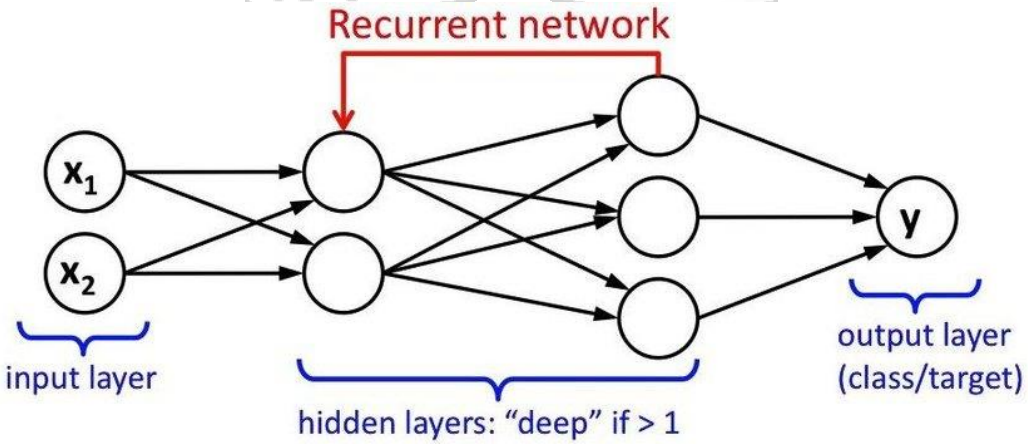

Figure 2.5 Recurrent Neural Network (Adapted From (Mishra et al., 2018))

Time series or sequences data information is captured by RNNs using the following equation:

$$S_t = F_w(S_{t-1}, X_t)$$

With, S_t being the state at time step t, F_w as the recursive function and, X_t as the input at time step t.

2.5 Related Works

2.5.1 Microsoft Sowing App India

In June 2016, Microsoft partnered with the International Crops Research Institute for the Semi-Arid Tropics (ICRISAT) in the Andhra Pradesh region to develop an application-AI Sowing App powered by machine learning and Power BI, that sends text messages to farmers in India advising on the optimal date to plant (ICRISAT, 2017). The Moisture Adequacy Index (MAI) is calculated to determine the optimal time to plant. MAI is defined as the ratio of actual to potential evapotranspiration (Jadhav et al., 2015). MAI is a standardized method for determining how well rainfall and soil moisture meet the prospective water requirements of crops. The daily rainfall data is used to generate the real-time MAI (ICRISAT, 2017).

2.5.2 Soil Moisture Prediction

A study conducted by Adab et al. (2020), was done in the early spring season in a semi-arid region of Iran to investigate the potential of machine learning algorithms. In previously untested conditions in a semi-arid region of Iran, the study used random forest (RF), support vector machine (SVM), artificial neural network (ANN), and elastic net regression (EN) algorithms to retrieve soil moisture using Landsat 8 optical and thermal sensors and knowledge of land-use types. According to the statistical comparisons, the RF technique had the greatest efficiency value (0.73) for soil moisture retrieval across all land-use categories. The study also found that surface reflectance and supplementary geographical data could be combined to provide more useful information for soil moisture content (SMC) estimate, which has applications in precision agriculture.

2.5.3 Rainfall Estimation in Sri Lanka

A study by Weerasinghe et al. (2010) set out to predict daily precipitation in the dry region of Sri Lanka based on observed ground-level measurements. The technique of feed-forward back-propagation neural networks was used in the study. The dataset used was comprised of thirty years of data (1970-1999) with twenty years of observed daily precipitation used as the training

set and ten years of data used as the testing set. Two models were developed in the study; the first predicted for happenings of rain in the form of "rain" or "no rain" output while the second model forecasted the precipitation amount using fuzzy techniques. On average the models had an accuracy score of approximately 79% (Weerasinghe et al., 2010).

2.5.4 Linear Regression prediction of Rainfall in India

A study by Thirumalai et al. (2017) based in India, involved the prediction of rainfall using the linear regression technique. Precipitation is predicted based on the values of the previous crop seasons, for instance, the rainfall amount recorded in the spring season - Rabi, determines the rainfall in the autumn season – Kharif; in this case, the Rabi season is the determinant. The mean and standard deviations of the past seasons are calculated to predict the rainfall for the next season. The mean formula is given as follows:

$$Mean, \mu = \sum (RABI_i)/n$$

While the standard deviation is calculated as

$$\sigma = SQRT(\sum (RABI_i)^2/n$$

With RABI referring to the rainfall values of the past years as shown in the table below,

Years	RABI	μ-RABI	(μ-RABI)^2
2006	26.41	7.24	52.51
2007	41.68	22.51	506.85
2008	27.45	8.28	68.55
2009	26.5	7.33	53.72
2010	27.83	8.66	75.05
2011	29.06	9.89	97.94
2012	31.93	12.76	162.90
2013	20.13	0.96	0.92
2014	25.26	6.09	37.16
2015	36.2	17.03	290.02
2016	28	8.83	77.96
	210.88		1017.55

Figure 2.6 Planting Season Data Adapted from (Thirumalai et al., 2017)

The researchers discover a low correlation between the crop planting seasons and the rainfall values predicted.

2.6 Research Gap

With the few numbers of synoptic weather stations, a gap exists in leveraging remotely sensed weather data to predict future weather conditions such as rainfall. This study seeks to address this gap by using remotely sensed climatic data of a farmer's location to develop a model that can forecast the rainfall for the coming days to offer advisory on planning such as planting time.

2.7 Conceptual Framework

In this section, the reviewed literature is linked to the research problem and objectives. The first step involves sourcing historical climatological datasets. To achieve this, a farmer opts in to receive recommendations via phone, upon subscription, the historical data is acquired via an API request sent to NASA POWER (2021) Application Programming Interface (API) to download the data. The data then undergoes normalization, that is cleansing of data to remove unnecessary data and/or impute missing values. Upon cleaning, the data is split into training, test, and validation sets, and the training set is fed into both deep learning algorithms resulting in a prediction model. The prediction model undergoes a performance assessment and is then deployed for production and can be interfaced via an API, this will enable the farmer to receive recommendation alerts via a push notification. The system will then pull the current weather conditions of the farmer's location from OpenWeatherMap API (OpenWeatherMap, n.d.). This data is passed as input to the generated regression model to predict the future rainfall values.

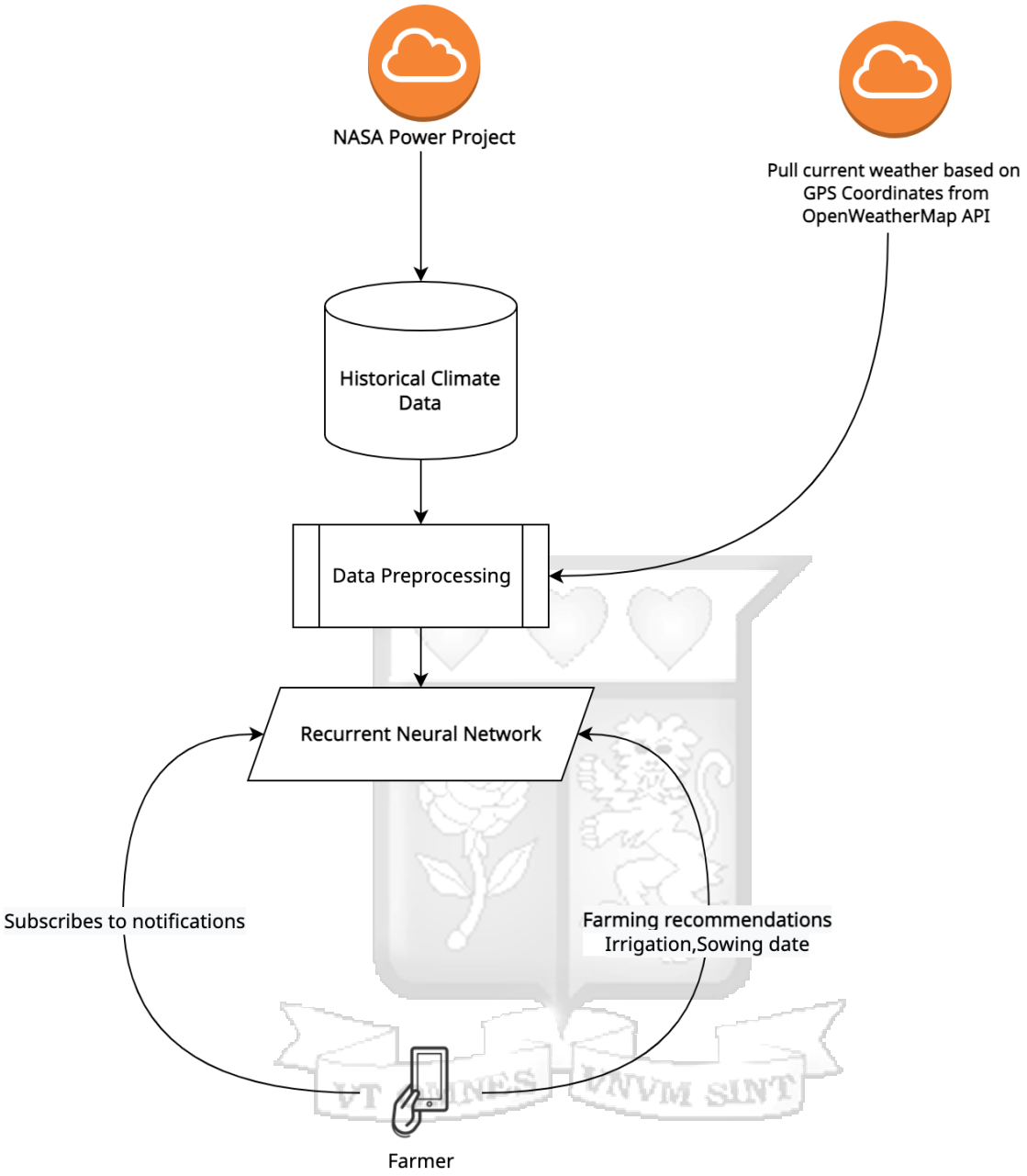

Figure 2.7 Conceptual Framework

Chapter 3. Methodology

3.1 Introduction

This section details the research approach conducted during the study. The intended objectives discussed in Chapter 1 have guided the approach of this study. The chapter describes the research design and the methodology used in the study. It then proceeds to discuss the population and sampling techniques used in the study. Data collection and analysis are then discussed and finally, the ethical considerations for the study are reviewed.

3.2 Research Design

According to Kothari (2004) research design is the task that comes after defining the research problem that involves deciding on "what, where, when, how much and by what means concerning an inquiry or a research study constitute a research design". Research design is thus a template for data gathering, measuring, and evaluation. It consists of a description of what the researcher will accomplish, beginning with the formulation of a hypothesis and its implications through to data analysis (Kothari, 2004).

This study is an applied research study as it is intended to help farmers receive farming planning recommendations such as the optimal planting period with a bid of addressing the issue of climatic variability. The system sends recommendation alerts to farmers based on the forecasted amount of rainfall in their respective locations.

3.2.1 Population and Sampling

The population target was comprised of historical time-series agroclimatology data collected from the National Aeronautics and Space Administration (NASA) Power Project (NASA POWER, 2021a). The study used a systematic sampling technique to acquire data. The historical climate data collected is specific to Machakos county, whose latitude and longitude coordinates are -1.5177 and 37.2634 respectively.

3.3 LSTM Model Development

The Long Short-Term Memory neural network model was developed in the following steps:

i. Data Collection
ii. Data pre-processing

iii. Training and fitting the model

iv. Validating the model

3.3.1 Data Collection

The study uses secondary data that is publicly available through an Application Programming Interface (API) and consists of five parameters including rainfall, temperature, humidity, and surface pressure with precipitation being the target variable in this study (NASA POWER, 2021a). The data collected is time-series in nature, comprising weather data recorded at daily intervals. In this study the data was spanned 21 years starting from January 1st, 2000 through to January 1st, 2022, this consisted of 8037 rows and 5 columns.

3.3.2 Data pre-processing

The data obtained had to undergo checks for duplicates and missing values in the pre-processing phase. The data is also scaled to values between 0 and 1 using Scikit Learn's MinMaxScaler function. Scaling of the data allows for faster training of the model. The scaled data was then reshaped into a format that the LSTM model can use to make subsequent predictions for the next n number of days. The shape of X-the predictor was changed from a two-dimension array consisting of rows and columns into a three-dimensional array consisting of the number of samples, the number of timesteps to look back, and the number of features. The predicted variable became a two-dimension array consisting of the number of samples and n values for the forecasted rainfall in millimeters per day (mm/day).

3.3.3 Model training

The model was trained and fitted with training data using the Tensorflow and Keras built-in LSTM function. The parameters passed into the function included the number of neural network layers-64, and the activation function to be used, which in this case was the Rectified Linear Unit (RELU).

3.3.4 Model Validation

Model validation was done using by calculating the error rate in terms of the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). MSE is defined as the average of the squared difference between the predicted values and the actual values. The MSE is a measure of how well the model fits the data, values closer to zero indicate

a better performing model (Brownlee, 2017b; Hiregoudar, 2022). The formula of MSE is given by:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

Where Y denotes the true/expected value and $\hat{Y}$ denotes the predicted value.

3.4 System Development Methodology

This research implemented the rapid application development (RAD) methodology. The RAD approach enables the swift development of applications from concept to completion at a cheap cost. The software is broken down into smaller pieces, making it easier to make changes as the project progresses. There are set delivery deadlines (time-boxes) for project components that should not be exceeded. The features are prioritized, and if necessary, the requirements are decreased to match the timeframe. RAD allows users to interact with variations of the system early in the initial stages since it uses prototypes (Geambaşu et al., 2011).

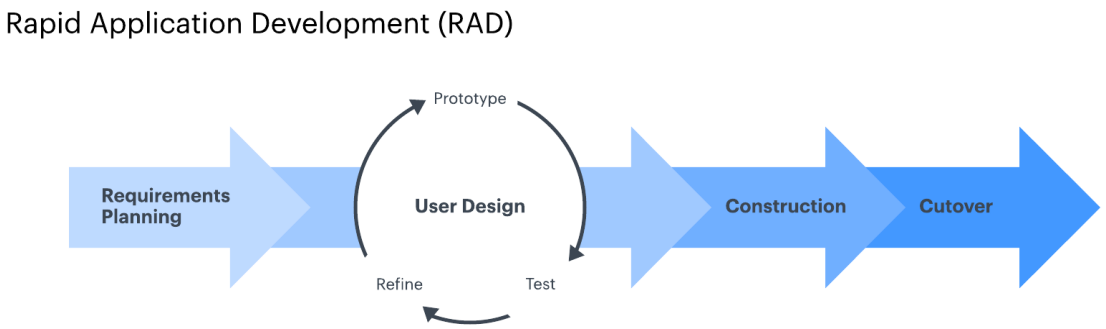

Figure 3.1 Rapid Application Development (Adapted from (Lucid Chart 2018))

3.5 Research Quality

Quality is a key aspect of this study. The study is conducted with objectivity, reliability, reproducibility, and reliability being the key drivers. The data is collected from credible and reputable sources. The resulting model is assessed using Mean Squared Error (MSE) and the Mean Absolute Error (MAE) described in section 3.3.4 which covers the model validation techniques implemented in the study.

3.6 Ethical Considerations

The study has been reviewed by the Strathmore University ethics committee. The study avoids plagiarism through attribution of other researchers' works using correct citations and does not

make up any data for the study. Technically, the study is intended to provide farmers with recommendations that will help in the planning and management of risk when farming.

Chapter 4. System Analysis, Design, and Architecture

4.1 Introduction

This chapter covers the architecture used to develop the rainfall prediction model; the architecture is based on the conceptual model discussed in Chapter 2. The system requirements, both functional and non-functional are also outlined in this section. User interaction with the system and interaction between components in the system are documented through use case diagrams, system sequence diagrams, and a flow chart.

4.2 Requirements Analysis

This section discusses the requirements that were derived from the research objectives. The requirements were classified into functional and non-functional requirements.

4.2.1 Functional Requirements

i. The system should accept historical data loaded from the NASA Power API
ii. The system should be able to infer the estimated rainfall
iii. The system should be able to integrate with OpenWeatherMap API for harvesting current weather condition
iv. The system should enable users to sign up with the platform for the first time and subsequently log in to gain access.
v. The system should send farmers push notifications concerning planting season recommendations

4.2.2 Non-functional requirements

i. The system should be able to operate on an Android mobile smartphone
ii. User data is to be transmitted in an encrypted mode to ensure the safety of the user data
iii. The mobile application should have an intuitive user interface that is easy to use and responsive to the users
iv. The prediction model should be able to predict and return the results in a reasonable amount of time
v. The system should be highly available for the user and in case of an error, the system will gracefully fail and the User shall be notified of the situation, allowing for a good user experience

vi. The system should be easily maintainable and robust, in case of a crash, the system should be back online within a reasonable period.

4.3 System Architecture

The system architecture for the rainfall prediction model is illustrated in Figure 4.1. Historical climatological profile time-series data is obtained from the NASA POWER (2021b). The data is then fed into a recurrent neural network to fit the data. The model is then validated and tested to determine its performance.

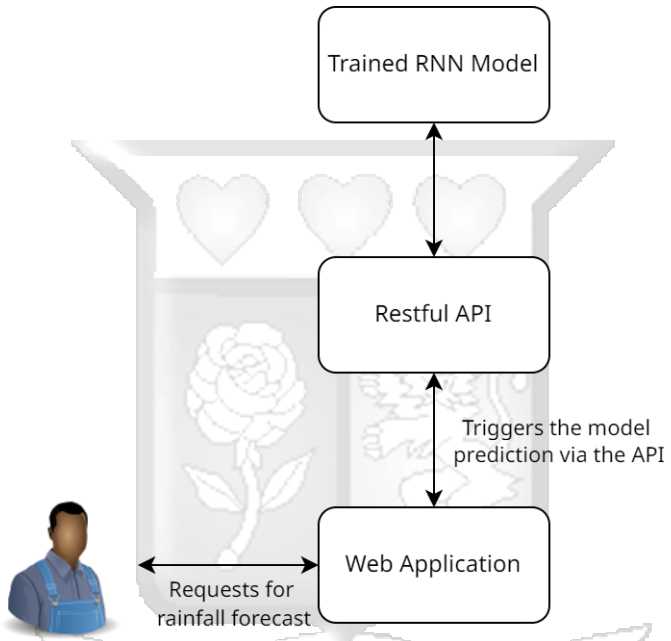

Figure 4.1 System Architecture

4.4 Use case diagram

This system is comprised of the following actors: the administrator, the farmer, and the database. The system loads historical data from the NASA Power Project, once the data is uploaded, feature extraction which entails determining the features with the highest information gain selected. Historical climate data that is collected from the NASA Power Project is stored in the database and is used when a farmer queries the estimated rainfall for a planting season. The farmer interacts with the system by signing up. The farmer can then login and opt-in to receive push notifications and view reports on the application.

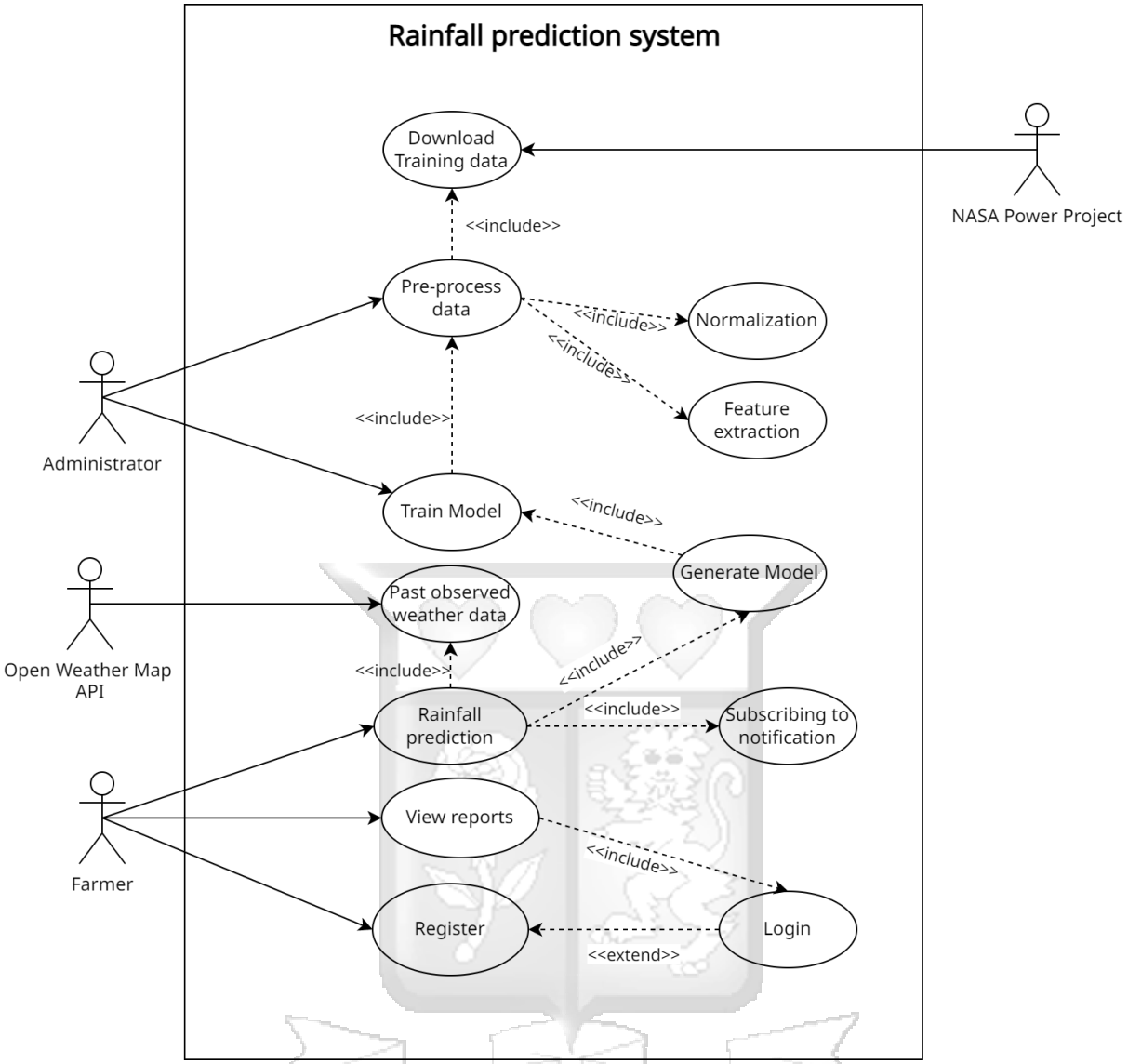

Figure 4.2 Use-case diagram

4.4.1 Detailed Use Case Scenarios

Use Case	Pre-process data
Primary Actors	Administrator
Pre-condition	Historical climatological time-series data are available
Post-condition	Pre-processed data that contains desired data
Main Success Scenarios	
Actor	**System**

Administrator acquires data in CSV format and places it in requisite folder	
The administrator defines the machine learning parameters	
	Feature extraction from the dataset is done by the system
	Saves the extracted features

Use Case	Model training
Primary Actors	Administrator
Pre-condition	Data is pre-processed, machine learning algorithms are available
Post-condition	A trained model that can predict rainfall
Main Success Scenarios	
Actor	**System**
The administrator triggers the training of pre-processed data	
	The system split the dataset into train, validation, and test set
	The system fits the data on the predefined algorithm
	The system validates the trained model using the test data
	The system generates a new model and saves it

Use Case	Rainfall prediction
Primary Actors	Farmer
Pre-condition	• Farmer is a registered user

	• Farmer is logged into the system • Farmer opts in to push notifications
Post-condition	Predicted rainfall
Main success scenarios	
Actor	**System**
The farmer logs into the system	
The farmer subscribes to push notification	
	The system predicts the rainfall
	System save the predicted rainfall
	The system generates the recommendation based on forecasted rainfall
The farmer receives recommendations via push notifications	

Use Case	View Reports
Primary Actors	Farmer
Pre-condition	• Farmer is registered • Farmer is logged in
Post-condition	Forecasted results are stored and compared with actual values
Main success scenarios	
Actor	**System**
Farmer signs in to view reports	
	System displays reports

4.5 System Sequence Diagram

The figure illustrates the system's sequence diagram. The administrator enters data which is in the Comma Separated Value (CSV) format. The uploaded data is then extracted for features with the most significant information gain. The data is then split into the training and testing datasets with the training data being fed to a recurrent neural network. The algorithm is then

deployed for prediction. The farmer then subscribes to receive farming recommendations, which prompts the system to forecast future rainfall amounts and return a recommendation.

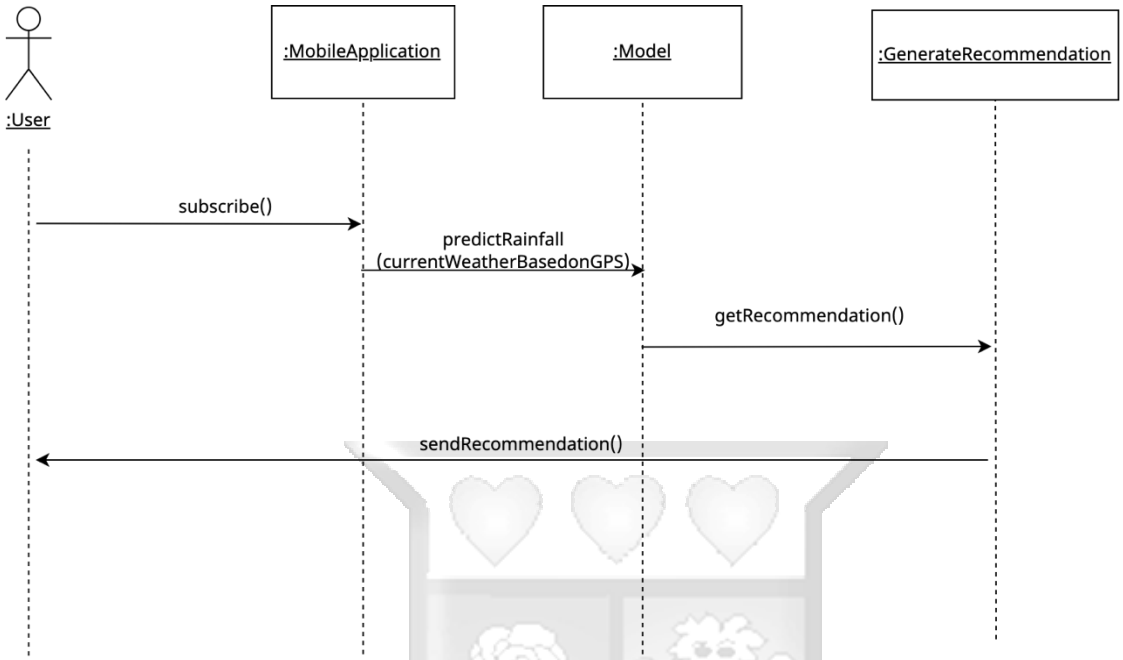

Figure 4.3 Sequence Diagram 1

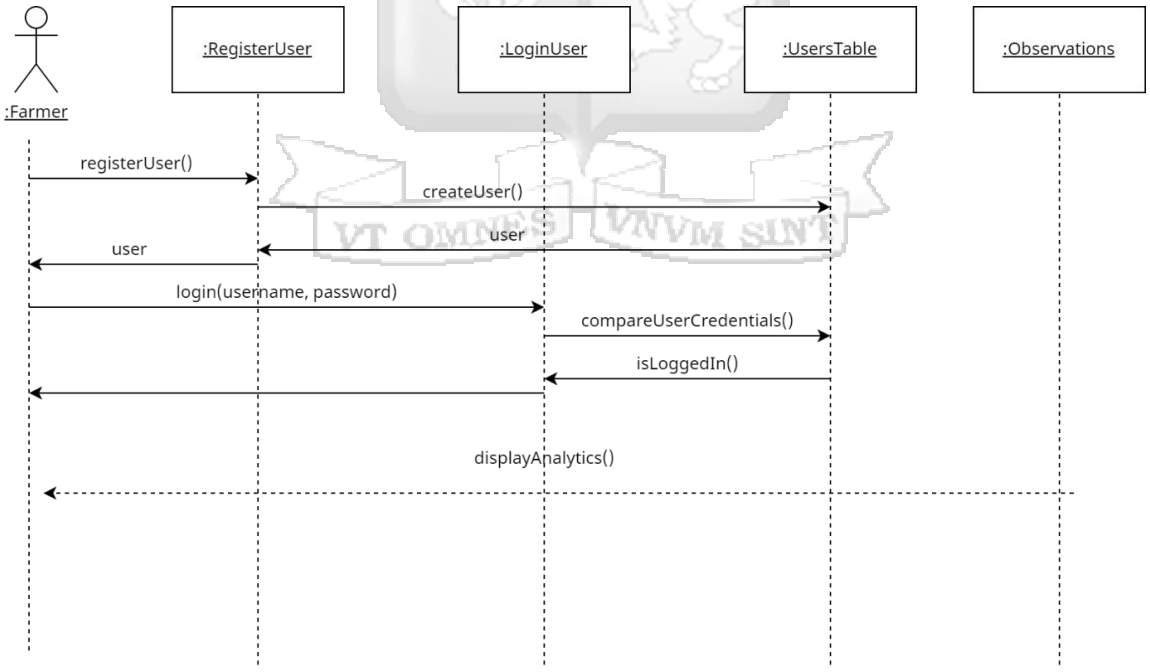

Figure 4.4 Sequence Diagram 2

4.6 Database Schema

The system uses a NoSQL database, MongoDB, which comes as a standalone, and a cloud-based database offering, Atlas. The flexible schema allows for fast iteration enabling rapid application development. The illustration below describes the relationship among entities within the system. The system consists of users, who are farmers. Their data is stored as well as their selected preferences on the mobile application.

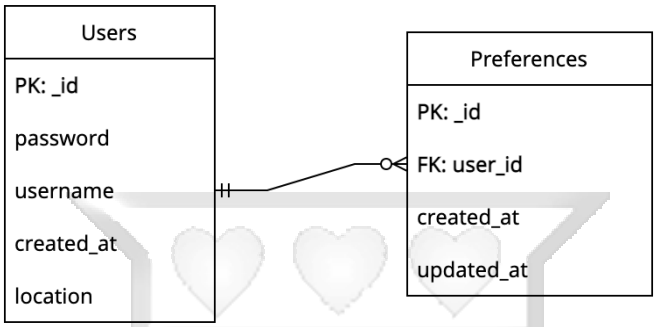

Figure 4.5 Database Schema

4.7 Wireframes

The figures below illustrate the mobile application's wireframes. The user can get recommendations after signing up in the system.

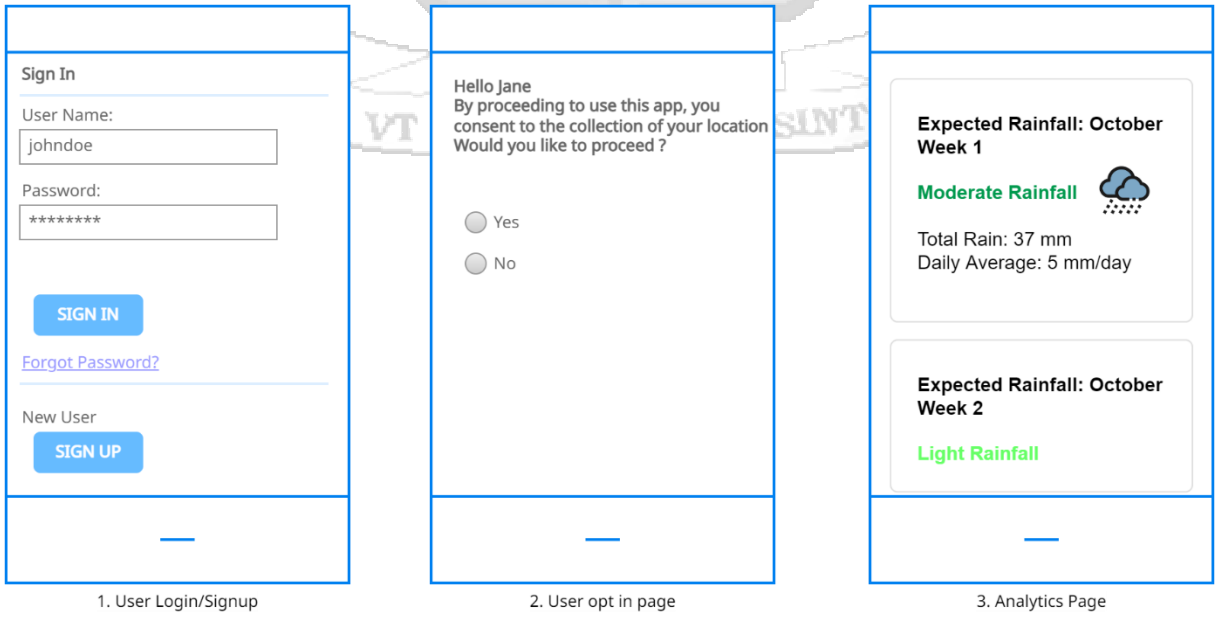

Figure 4.6 Wireframes

Chapter 5. System Implementation and Testing

5.1 Introduction

This section describes the components that make up the rainfall prediction system. The development of the prediction model is covered in this section. The process of processing the collected data and feeding it to the Long Short-Term Model is explained. The testing of the model is discussed in this chapter. The development environment is detailed in this section in terms of the programming language, integrated development environment as well as the prerequisite software libraries needed to replicate the system in another hardware and/or operating system.

5.2 Model Development

As discussed in Chapter 2, the model to be used in predicting rainfall will be the Long Short-Term Memory (LSTM) recurrent neural network. This section covers the development environment, hardware, and software resources needed to develop the model.

5.2.1 Development environment

The model was developed using Visual Studio text editor which had the Jupyter Notebook and Python extensions installed. The extensions assist with code linting and IntelliSense auto-completion. Jupyter notebook makes it easier to carry out data science projects by allowing for running sections of code as opposed to running a whole python script as well as in documentation through the use of markdown.

5.2.2 Hardware requirements

Table 5.1 indicates the specifications of the device laptop that was used in the development of the system.

Hardware	Specifications
Central Processing Unit (CPU)	Intel(R) Core (TM) i7-6820HQ CPU @ 2.70GHz (8 CPUs), ~2.7GHz
Memory	8GB RAM
Disk	256 Solid State Drive
Integrated Graphics Chipset	Intel(R) HD Graphics 530

Table 5.1 Hardware Specification

5.2.3 Software requirements

The programming language used in the development of the prediction model was Python version 3.10.8. The Tensorflow and Keras libraries were used to provide a production-ready implementation of the LSTM neural network while the Flask library was used in the development of a RESTful API that would be consumed by a mobile application. Table 5.2 lists the main Python libraries that were used.

Library	Version
Tensorflow	2.8.0
Keras	2.8.0
Pandas	1.4.1
Numpy	1.22.3
Jupyter	1.0.0
Matplotlib	3.5.1
Flask	2.0.3
React	18.0.0

Table 5.2 Software Specification

5.3 Model Architecture

Recurrent neural networks, also known as RNNs are a type of neural network that is used to process sequential data. An RNN is a neural net that is skilled for having to process a sequence x(1),..., x(). RNNs scale to much lengthier sequences than would be feasible for other networks without sequential specialization, just as CNN models can easily scale to photos with large height and width, and some CNN models could process images of various sizes. Most RNNs can also handle variable-length sequences. Long Short-Term Memory (LSTM) networks are a class of RNNs that are referred to as gated RNNs. These RNNs are most reliable in practical applications. Gated RNNs implement weights that respond to change in each time step which helps in the problem of vanishing derivatives that occurs in general RNNs (Goodfellow et al., 2016).

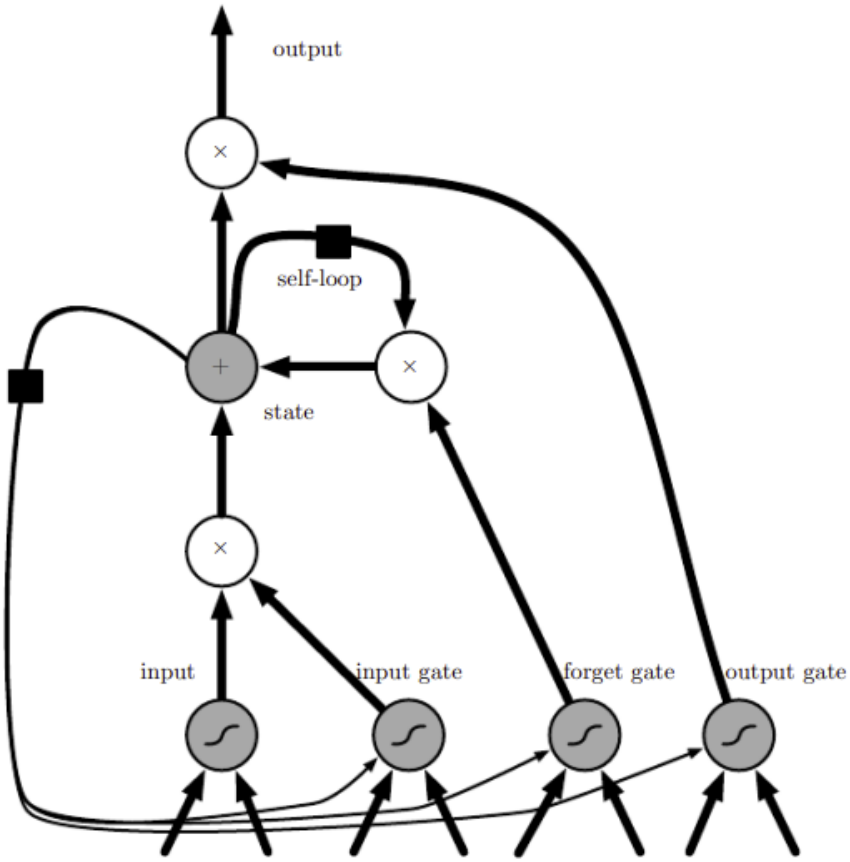

Figure 5.1 LSTM RNN Cell Architecture (Adapted from (Goodfellow et al., 2016))

The LSTM cell's central principle is to organize its internal operations around two markedly distinct, yet complementary, objectives: data and data control. The attribute data signals (between -1 and 1) are prepared by the data components, while the "throttle" signals are prepared by the control components (ranging between 0 and 1). The fractional amount of selection data that is permitted to propagate to its nodes is computed by multiplying the attribute digital signals by the control signal. As a result, if the signal is 0, only 0% of the evaluation data will propagate. In the case of the control signal being 1, the candidate data amount would then propagate in its entirety (Sherstinsky, 2020).

5.4 Model Implementation

5.4.1 Data Collection

Historical data is fetched using NASA Power API in a Comma Separated Value (CSV) format. The CSV file is automatically loaded once it’s downloaded using the Pandas library. The data consists of 5 parameter columns:

- **T2M** - Temperature at 2 Meters (C)
- **PS** - Surface Pressure (kPa)
- **WS10M** - Wind Speed at 10 Meters (m/s)
- **QV2M** - Specific Humidity at 2 Meters (g/kg)
- **PRECTOTCORR** - Precipitation Corrected (mm/day)

The figure below illustrates the NASA Power documentation on how data can be accessed from its platform.

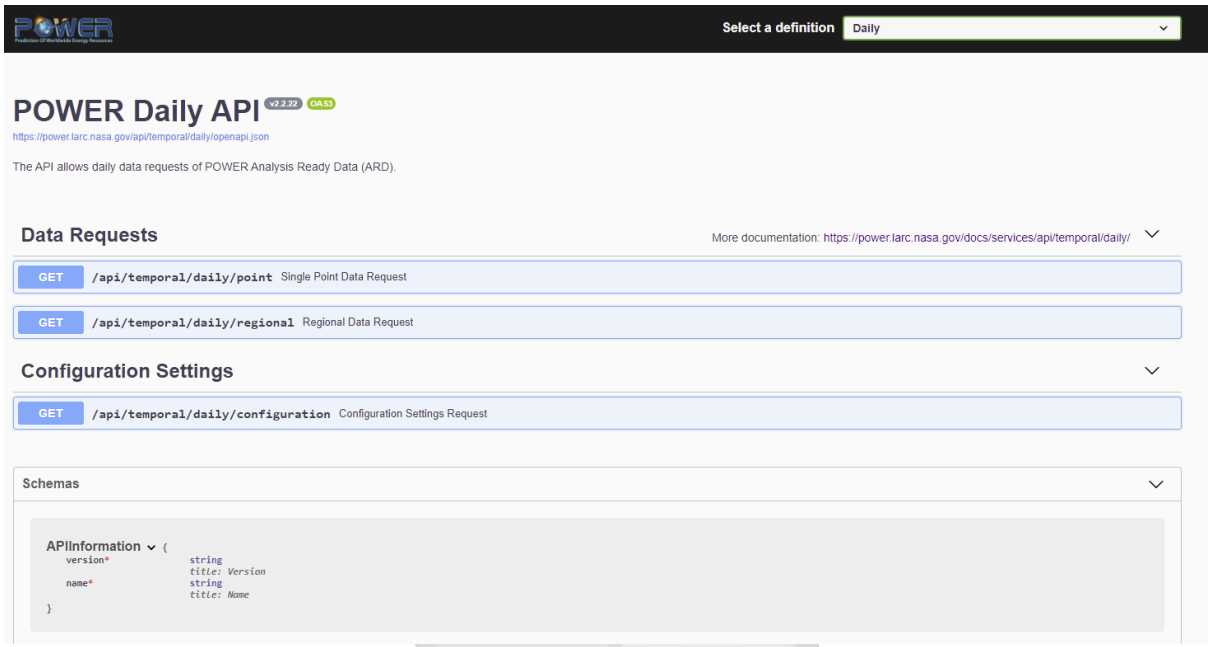

Figure 5.2 Data Access via NASA Power API

5.4.2 Data preprocessing

An exploratory analysis of the data is conducted, and the date values by default are stored in two columns: Year and Day of Year. These two columns had to be consolidated into one date column with the format day/month/year. The data looks as illustrated below

date	T2M	PS	WS10M	QV2M	PRECTOTCORR
2022-03-15	23.58	86.65	2.64	11.17	0.06
2022-03-16	24.25	86.54	2.71	10.74	0.01
2022-03-17	23.58	86.57	3.03	11.66	0.55
2022-03-18	-999.00	-999.00	-999.00	-999.00	-999.00
2022-03-19	-999.00	-999.00	-999.00	-999.00	-999.00

Figure 5.3 Historical data with formatted date

The next step involved the removal of rows that contain missing values. In the case of the collected data, the value -999.00 indicates a null/missing value. The rows with missing values are removed using the Pandas library, drop function.

Box plots are then plotted to indicate the outliers present within the data since has a significant impact on the error rate of the prediction model. Figure 5.3 below outlines the scatter plot of the data:

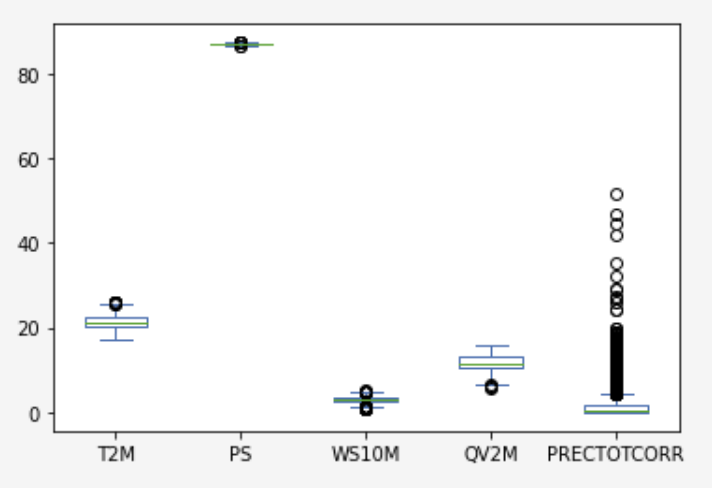

Figure 5.4 Scatter plot for historical data

The outlier data is not removed in this case since genuine outliers within data are "more likely to be representative of the population as a whole if the outliers are not removed. (Osborne & Overbay, 2019)"

A correlation heat map is then plotted to analyze the case of multi-collinearity among the predictor variables. Multi-collinearity occurs when two variables that are assumed to be independent of each other, are found to be closely related to each other. Multi-collinearity of

variables impacts machine learning by leading to decreased statistical significance, increased standard error, and overfitting (Pardeshi, 2020). The correlation map is shown in Figure 5.5 below.

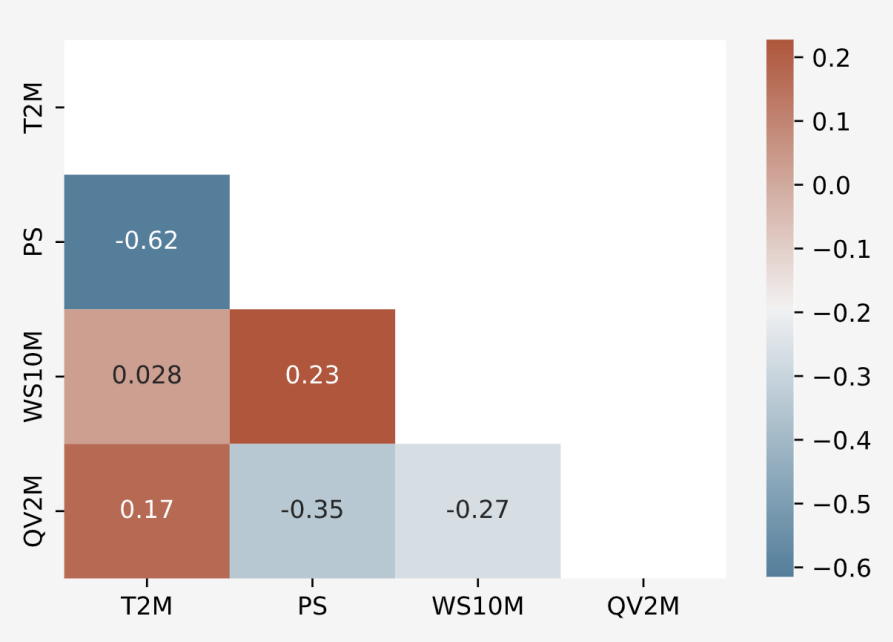

Figure 5.5 Feature correlation heatmap

Correlation values among variables greater than 0.7 generally indicate a case of multi-collinearity (Kumar, 2020). In this case, however, the variables are below the threshold and are thus used as predictors of rainfall.

5.4.3 Scaling data

The data consists of different scales in the SI units of the collected data. This data ought to be standardized since the Long Short-Term Memory neural network expects data in a range between 0 and 1 to allow for faster computation of the model. To scale the data, Scikit Learn's MinMaxScaler is used to generate values between 0 and 1. Below is a snippet of how the scaled data is represented.

0	0.2327981651	0.670360…	0.1165919283	0
1	0.2672018349	0.581717…	0.1031390135	0
2	0.4403669725	0.421052…	0.2679372197	0
3	0.4908256881	0.432132…	0.3284753363	0
4	0.5401376147	0.440443…	0.3497757848	0.0210084034
5	0.4908256881	0.501385…	0.4865470852	0.1369747899
6	0.502293578	0.581717…	0.4585201794	0.0218487395

Figure 5.6 Scaled data

The MinMaxScaler function works as follows as illustrated in (*Sklearn.Preprocessing.MinMaxScaler*, n.d.) webpage:

```
X_std = (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))
X_scaled = X_std * (max - min) + min
```

Figure 5.7 MinMaxScaler Implementation

5.4.4 Training the model

The model was trained using supervised learning. To train the model, the LSTM model expects training data to be in a three-dimensional shape consisting of [number of samples, number of days to look back, number of days to predict], in this case, the shape of the data is transformed from (3617,3) to (3583, 60, 30) since the model forecasts the rainfall values in millimeters for the next 30 days by looking back at the past 60 days. The output of the LSTM model is a two-dimensional matrix of the shape (1,30) indicating one row with 30 days of predicted values of rainfall.

The data is then split into the training and testing set using Sklearn's train_test_split function, which accepts the X and y values which in this case, the X values represent the previous 60-day climate conditions (temperature, wind speed, humidity, atmospheric pressure) while the y values represent the rainfall values for the subsequent 30 days. The data is split such that the test data is 33% of the entire dataset

Upon reshaping and splitting the scaled data, the data was passed into Keras' implementation of the inbuilt LSTM function. To fit the model, 60 epochs were used with a batch size of 64, this was done on an experimental basis. Figure 5.8 shows the LSTM implementation

```
In [22]:
model = Sequential()
model.add(LSTM(64, activation='relu', input_shape=(x_train.shape[1], x_train.shape[2]), return_sequences
model.add(LSTM(32, activation='relu', return_sequences=False, recurrent_dropout=0.2, unroll=True))
model.add(Dropout(0.2))
model.add(Dense(y_train.shape[1]))
model.compile(optimizer='adam', loss='mse')
model.summary()

In [34]:
history = model.fit(x_train, y_train, epochs=100, batch_size=28, validation_split=0.33, verbose=1)
```

Figure 5.8 LSTM Implementation

5.5 Model Evaluation

Upon training of the model, a graph was plotted to highlight the validation and training loss. The loss values indicate the "goodness" of the model with the lesser loss values indicating a better model; caution however needs to be taken to avoid overfitting the data. The training loss decreases as the number of epochs increases to 100, this is also the case with the validation loss as shown in Figure 5.9 below

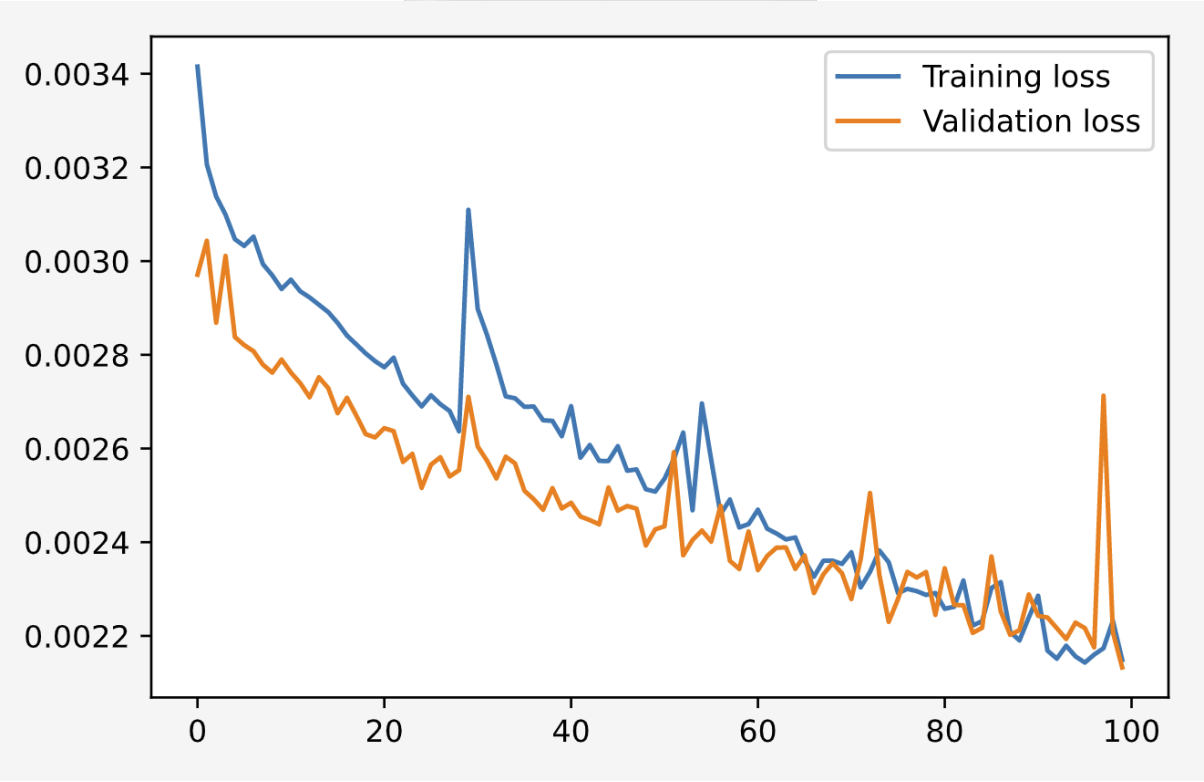

Figure 5.9 Training vs Validation loss

The performance of the model was evaluated in terms of the Root Mean Square Error (RMSE) and was found to be 2.45 millimeters for the 30-day weather forecasting model.

5.6 Forecasting rainfall with deployed model

A backend Restful Application Programming Interface (API) was then developed. The backend was developed using the Python programming language and it provides an interface to the saved model. Using the API, one can be able to trigger the download of historical climate data for the past 60 days for the user location of the farmer using the app. The downloaded data is fed into the LSTM model and a 30-day forecast is returned. This result is presented to the farmer in a user-friendly format through a frontend application.

The frontend application is a web application that is also mobile-friendly and responsive. It acts as an interface between the farmer and the API backend. Upon successfully logging in, the farmer can view the outlook of the coming weeks. Figure 5.10 illustrates the user interface for the web application, and Figure 5.11 illustrates the responsive web application on a mobile phone.

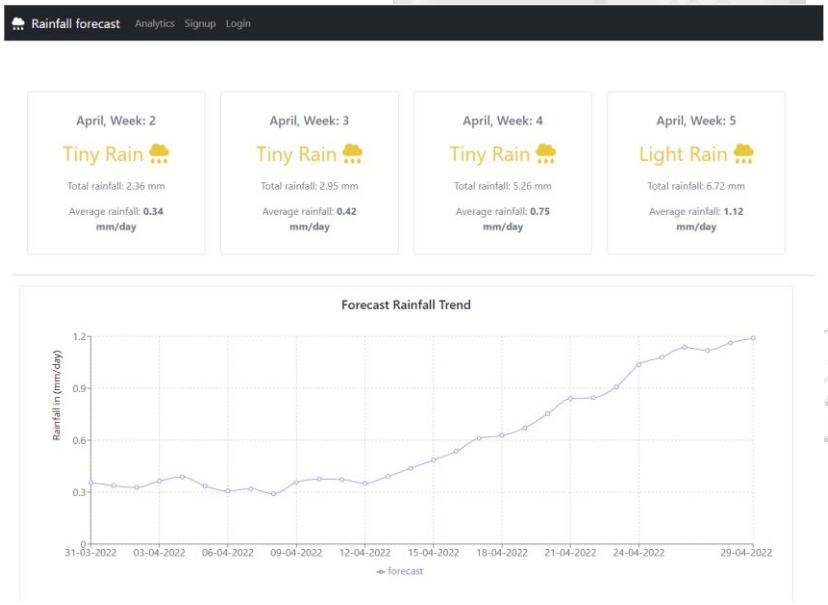

Figure 5.10 Rainfall outlook (Web User Interface)

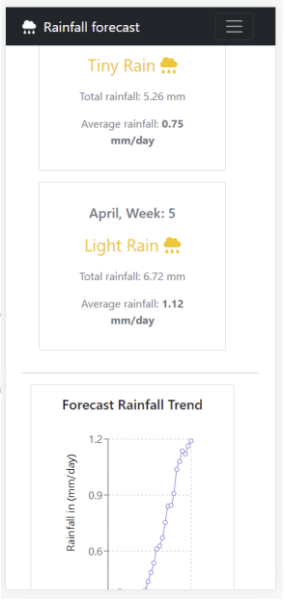

Figure 5.11 Mobile UI

The farmer can view the outlook of the approximated rainfall in the coming weeks as well as the trend of the rainfall patterns over 30 days. This will help the farmer to determine the optimal time to sow his/her maize seeds. Color codes are used to indicate different rainfall intensities for instance Figures 5.10 and 5.11 indicate use orange to indicate tiny to light rainfall. Figure 5.12 shows the various color codes for various rainfall intensities.

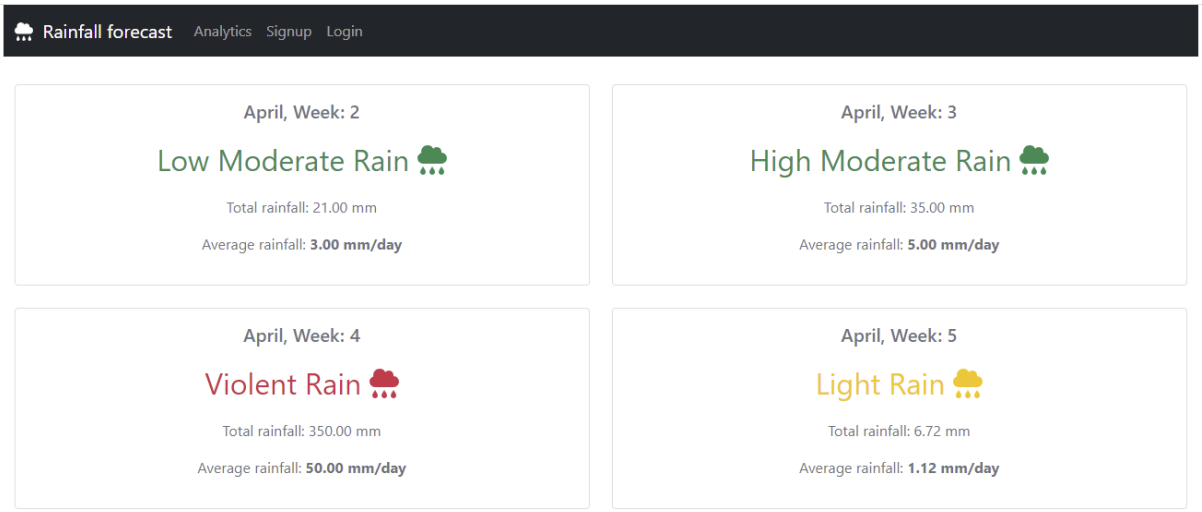

Figure 5.12 Rainfall Intensity Color Coding

Chapter 6. Discussion

6.1 Introduction

Thus far, the research has covered the literature, methodology, data analysis, and the implementation of the rainfall prediction system. In this chapter, section 6.2 analyses the performance of the Long Short-Term Memory (LSTM) model at forecasting the rainfall values of the various number of days into the future based on looking back 60 days of recorded weather parameters. In this case, the number of days testing in the future were 3, 5, 7, and 30 days. In section 6.3 the results of the experiments conducted when developing the rainfall forecasting model are discussed and presented. The experiments conducted indicate that the 3-day rainfall forecasting model has the lowest root mean squared error, followed by the 7-day, 30-day, and 5-day forecasting models respectively.

6.2 Model Validation

Model validation was conducted by assessing the error rates of the model based on validation data the model had not encountered during the training phase. The validation data consisted of 1595 rows which is approximately 20% of the collected dataset. The model was tasked to forecast the rainfall a varying number of days into the future. The number of days into the future used in the experiments were 3, 5, 7, and 30 days respectively. The figure below indicates the training performance of the model.

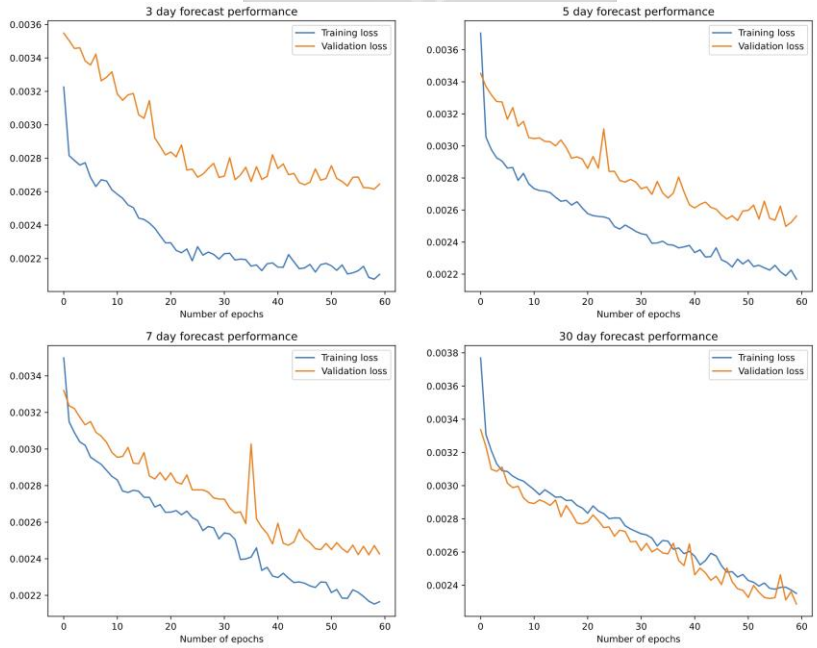

Figure 6.1 Model performance

From the above figure, the 3-day, 5-day, and 7-day models show the training loss being lower than the validation loss with the trend in the validation loss suggesting further improvements being possible, this indicates a case of an underfitting model (Brownlee, 2017b). For the 30-day prediction model, the performance of the training and validation loss decrease steadily hence it is a better-suited candidate for deployment.

6.3 Model Performance Results

The experiment proceeded to determine the performance of each LSTM model by calculating the error rate of the models determined by comparing the predicted rainfall amount to the actual/expected rainfall amount. The subsections below discuss the various metrics that assess the model performance.

6.3.1 Mean Absolute Error

The Mean Absolute Error (MAE) computes the average on the absolute value of the difference between the actual and the predicted values given by

$$MAE = mean\big(abs(expected_{value} - predicted_{value})\big)$$

The *abs()* function makes values of the difference calculation positive, while the *mean()* function calculates the average value of the absolute error. The table below shows the MAE of the different models. The units of the error value error are given in terms of millimeters per day (mm/day).

Model	Mean Absolute Error (MAE)
3-day	1.32
5-day	1.29
7-day	1.26
30-day	1.37

6.3.2 Mean Squared Error

The Mean Square Error (MSE) is a performance metric that calculates the "average of the squared forecast error values" (Brownlee, 2017a). Squaring the error values has the effect of making the error values positive while also magnifying and punishing large prediction errors. An MSE of zero indicates the model has no error or has a perfect skill (Brownlee, 2017a).

The MSE function is given by

$$forecastError = expected_{value} - predicted_{value}$$

$$MSE = mean(forecastError^2)$$

The table below shows the MSE values of each model, with the values being in squared millimeters per day (mm/day) of rainfall

Model	Mean Squared Error (MSE)
3-day	5.88
5-day	6.03
7-day	5.83
30-day	6.01

The values above are large, however, the fact the outliers were not removed from the data ought to be considered.

6.3.3 Root Mean Square

The RMSE calculates the square root of the MSE to get the error values in actual units, mm/day, in this case. An RMSE value of zero as with the MSE is indicative of no error. The table below shows the RMSE value for each model under test.

Model	Root Mean Squared Error (RMSE)
3-day	2.42
5-day	2.45
7-day	2.41
30-day	2.45

6.3.4 Discussion

Based on the validation results of each model generated from the preceding subsection 6.3, there is little difference in the RMSE of the four models. However, based on the ability to properly fit the training data as discussed in subsection 6.2, the 30-day forecasting model is found to have the best fit compared to the rest of the models created in the experiment.

6.4 Contribution to Research

The developed model provided a solution to estimating the rainfall values. The model is then deployed for production to be interfaced via a mobile application. The mobile provides insights to the farmers with regards to expected rainfall in the next 30 days based on the farmer's Global Positioning System (GPS) location. The location, in this case, is limited to Machakos county, given that only Machakos county climate data was used to train the model.

Chapter 7. Conclusion and Recommendations

7.1 Conclusion

This study was intended to develop a tool to forecast rainfall using Long Short-Term Memory (LSTM) aimed at helping maize farmers determine the optimal time to plant. To achieve this, it was essential to understand the current state of maize planting and weather forecasting techniques used in Kenya. Chapter 2 of the study reviewed the existing literature on maize production in Kenya, the section also covered the challenges the farmers faced concerning the current weather forecast information being too "coarse-grained" and too technical for their daily usage. The challenges faced by the Kenyan Meteorological Department are also reviewed. The literature further reviews the application of machine and deep learning algorithms in the forecasting of rainfall as well as the techniques used in evaluating model performance.

The primary objective of the study was to develop a rainfall prediction tool that would estimate the rainfall of a given farmer's GPS coordinates. The GPS coordinates are essential to retrieving observed climatic data from an online source-NASA Power (NASA POWER, 2021b). Once the historical data is collected, the data is fed into a developed LSTM model to produce the forecast for the next 30 days. The data is then presented to the farmer in a user-friendly and actionable manner in a bid to determine the optimal time to plant.

Experiments were conducted by the researcher to determine the optimal number of days into the future that the rainfall forecasting model could do. The experiment consisted of four models which included a 3-day, 5-day, 7-day, and 30-day model respectively. An assessment of the models' ability to properly fit the data and the error rate was used to determine the best model. Using the above criteria, the experiments determined the 30-day rainfall forecasting model to be the most optimal one. The model was then persisted for production and was used in developing a Restful API.

7.2 Recommendations and Future Work

The researcher identifies the following areas as the potential for exploration:

i. The model could be trained with more data to increase the location scope of rainfall prediction.

ii. The model could be repurposed with other backend applications to give suggestions for other plants.

iii. The system could be interfaced with a remote monitoring application that would help in checking the state of the farm for owners who are located far from their tracts of land

References

Ali-Olubandwa, A. M., Kathuri, N. J., Odero-Wanga, D., & Shivoga, W. A. (2011). Challenges Facing Small Scale Maize Farmers in Western Province of Kenya in the Agricultural Reform Era. *Journal of Experimental Agriculture International*, 466–476. https://doi.org/10.9734/AJEA/2011/649

Alpaydin, E. (2010). *Introduction to machine learning 2nd ed*. MIT Press. https://mitpress.mit.edu/books/introduction-machine-learning-second-edition

Belloumi, M. (2014). *Investigating the impact of climate change on agricultural production in eastern and southern African countries*. International Food Policy Research Institute. https://ebrary.ifpri.org/digital/collection/p15738coll2/id/128227

Brownlee, J. (2017a, January 31). Time Series Forecasting Performance Measures With Python. *Machine Learning Mastery*. https://machinelearningmastery.com/time-series-forecasting-performance-measures-with-python/

Brownlee, J. (2017b, August 31). How to Diagnose Overfitting and Underfitting of LSTM Models. *Machine Learning Mastery*. https://machinelearningmastery.com/diagnose-overfitting-underfitting-lstm-models/

Dhumale, R. B., Thombare, N. D., & Bangare, P. M. (2019). Machine Learning: A Way of Dealing with Artificial Intelligence. *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, 1–6. https://doi.org/10.1109/ICIICT1.2019.8741360

Dowker, B. D. (1963). Rainfall Reliability and Maize Yields in Machakos District. *East African Agricultural and Forestry Journal*, *28*(3), 134–138. https://doi.org/10.1080/00128325.1963.11661860

FAOSTAT. (2021). *FAOSTAT*. http://www.fao.org/faostat/en/#data/QCL

Geambaşu, C. V., Jianu, I., Jianu, I., & Gavrilă, A. (2011). Influence factors for the choice of a software development methodology. *Accounting and Management Information Systems*, *10*(4), 479–494.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. https://www.deeplearningbook.org/contents/intro.html

Hansen, J. W., & Indeje, M. (2004). Linking dynamic seasonal climate forecasts with crop simulation for maize yield prediction in semi-arid Kenya. *Agricultural and Forest Meteorology*, *125*(1), 143–157. https://doi.org/10.1016/j.agrformet.2004.02.006

Hiregoudar, S. (2022, March 4). *Ways to Evaluate Regression Models*. Medium. https://towardsdatascience.com/ways-to-evaluate-regression-models-77a3ff45ba70

ICRISAT. (2017, January 9). *Microsoft and ICRISAT's Intelligent Cloud pilot for Agriculture in Andhra Pradesh increase crop yield for farmers – ICRISAT*. https://www.icrisat.org/microsoft-and-icrisats-intelligent-cloud-pilot-for-agriculture-in-andhra-pradesh-increase-crop-yield-for-farmers/

IPCC. (2014). *AR5 Synthesis Report: Climate Change 2014 — IPCC*. https://www.ipcc.ch/report/ar5/syr/

Ireri, D. M. (2020). Influence of ICT Weather Forecasting on Agricultural Productivity in Kenya: A Literature Based Review. *Journal of Information and Technology*, *4*(1), 56–69.

Jadhav, M. G., Aher, H. V., Jadhav, A. S., & Gote, G. N. (2015). Crop Planning Based on Moisture Adequacy Index (MAI) of Different Talukas of Aurangabad District of Maharashtra. *Indian Journal of Dryland Agricultural Research and Development*, *30*(1), 101. https://doi.org/10.5958/2231-6701.2015.00016.0

Jimeno-Sáez, P., Blanco-Gómez, P., Pérez-Sánchez, J., Cecilia, J., & Senent-Aparicio, J. (2021). Impact Assessment of Gridded Precipitation Products on Streamflow Simulations over a Poorly Gauged Basin in El Salvador. *Water*, *13*. https://doi.org/10.3390/w13182497

KARLO. (n.d.). *Maize Agronomy*. Retrieved April 8, 2022, from https://www.kalro.org/maize/maize-agronomy/

Kilimo Open Data. (2020, July). *Datasets—Kilimo Open Data*. http://kilimodata.developlocal.org/dataset

Kothari, C. R. (2004). *Research Methodology: Methods and techniques*. New Age International.

Kumar, A. (2020, September 29). Correlation Concepts, Matrix & Heatmap using Seaborn. *Data Analytics*. https://vitalflux.com/correlation-heatmap-with-seaborn-pandas/

Masinde, M., & Bagula, A. (2011). ITIKI: Bridge between African indigenous knowledge and modern science of drought prediction. *Journal of Knowledge Management*, *7*, 274–290. https://doi.org/10.1080/19474199.2012.683444

Masinde, M., Bagula, A., & Nzioka, M. (2013). SenseWeather: Based weather monitoring system for Kenya. *2013 IST-Africa Conference Exhibition*, 1–13.

McCulloh, W., & Pitts, W. (1943). *A logical calculus of the ideas immanent in nervous activity*. 19.

Mwangi, E. W., & Mundia, C. N. (2022). Assessing and monitoring agriculture crop production for improved food security in Machakos County. *Proceedings of the Sustainable Research and Innovation Conference*, 310–313.

NASA POWER. (2021a). *NASA POWER | Prediction Of Worldwide Energy Resources*. https://power.larc.nasa.gov/#resources

NASA POWER. (2021b). *POWER | Data Access Viewer*. https://power.larc.nasa.gov/data-access-viewer/

Ngotho, A. (2015, July 28). *Stress Tolerant Maize for Africa » Machakos farmers adopt improved maize varieties*. https://stma.cimmyt.org/machakos-farmers-adopt-improved-maize-varieties/

Ochieng, J., Kirimi, L., & Mathenge, M. (2016). Effects of climate variability and change on agricultural production: The case of small scale farmers in Kenya. *NJAS: Wageningen Journal of Life Sciences, 77*(1), 71–78. https://doi.org/10.1016/j.njas.2016.03.005

OpenWeatherMap. (n.d.). *Weather API - OpenWeatherMap*. Retrieved March 12, 2022, from https://openweathermap.org/api

Osborne, J., & Overbay, A. (2019). The power of outliers (and why researchers should ALWAYS check for them). *Practical Assessment, Research, and Evaluation*, *9*(1). https://doi.org/10.7275/qf69-7k43

Pardeshi, V. (2020, August 22). *Handling Multi-Collinearity in ML Models*. Medium. https://towardsdatascience.com/handling-multi-collinearity-6579eb99fd81

Prakash, S., Sharma, A., & Sahu, S. S. (2018). Soil Moisture Prediction Using Machine Learning. *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 1–6. https://doi.org/10.1109/ICICCT.2018.8473260

Shabbir, J., & Anwer, T. (2018). Artificial Intelligence and its Role in Near Future. *ArXiv:1804.01396 [Cs]*. http://arxiv.org/abs/1804.01396

Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306. https://doi.org/10.1016/j.physd.2019.132306

Short, C., Mulinge, W., & Witwer, M. (2012). *Analysis of incentives and disincentives for maize in Kenya*. 50.

Sklearn.preprocessing.MinMaxScaler. (n.d.). Scikit-Learn. Retrieved March 27, 2022, from https://scikit-learn/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

Thirumalai, C., Harsha, K. S., Deepak, M. L., & Krishna, K. C. (2017). Heuristic prediction of rainfall using machine learning techniques. *2017 International Conference on Trends in Electronics and Informatics (ICEI)*, 1114–1117. https://doi.org/10.1109/ICOEI.2017.8300884

van Klompenburg, T., Kassahun, A., & Catal, C. (2020). Crop yield prediction using machine learning: A systematic literature review. *Computers and Electronics in Agriculture*, *177*, 105709. https://doi.org/10.1016/j.compag.2020.105709

Velesi, D. S. (2018). *Factors Affecting Maize Yield in Machakos County*. Machakos University. http://ir.mksu.ac.ke/handle/123456780/692

Wang, H., Ma, C., & Zhou, L. (2009). A Brief Review of Machine Learning and Its Application. *2009 International Conference on Information Engineering and Computer Science*, 1–4. https://doi.org/10.1109/ICIECS.2009.5362936

Weerasinghe, H., Premaratne, H., & Sonnadara, D. (2010). *Performance of neural networks in forecasting daily precipitation using multiple sources*.

World Bank Group. (2018). *Kenya Economic Update, April 2018, No. 17: Policy Options to Advance the Big 4*. World Bank. https://openknowledge.worldbank.org/handle/10986/29676

Appendix A: Originality Report

Curiginal

Document Information

Analyzed document	Thesis-80677-Brian Mwathi.pdf (D133316766)
Submitted	2022-04-11T08:05:00.0000000
Submitted by	
Submitter email	Brian.Mwathi@strathmore.edu
Similarity	3%
Analysis address	library.strath@analysis.urkund.com

Sources included in the report

SA	**6303_Akhil Kumar_Dissertation.pdf** Document 6303_Akhil Kumar_Dissertation.pdf (D116738628)	1
W	URL: https://oaktrust.library.tamu.edu/bitstream/handle/1969.1/173974/KHURJEKAR-THESIS-2018.pdf?sequence=1&isAllowed=y Fetched: 2021-12-21T12:02:16.6570000	1
SA	**_Failure_Prediction_in_Composites_using_Machine_Learning_ (7).pdf** Document _Failure_Prediction_in_Composites_using_Machine_Learning_ (7).pdf (D121148758)	1
SA	**Final Notebook S190031.html** Document Final Notebook S190031.html (D72388779)	3
SA	**The final report.pdf** Document The final report.pdf (D108816284)	1
SA	**ForecastingTheSeasons_Franch_Hansen_full.pdf** Document ForecastingTheSeasons_Franch_Hansen_full.pdf (D38731590)	1
W	URL: https://www.ijert.org/rainfall-prediction-using-linear-approach-neural-networks-and-crop-recommendation-based-on-decision-tree Fetched: 2020-05-01T07:15:19.3100000	1
W	URL: https://programmerclick.com/article/51801186613/ Fetched: 2022-04-11T08:06:17.0370000	1
W	URL: https://iq.opengenus.org/text-generation-lstm/ Fetched: 2020-12-01T06:00:19.7970000	1
SA	**Linear+Regression+-+Breast+Cancer+Dataset+-+a19kimgr.pdf** Document Linear+Regression+-+Breast+Cancer+Dataset+-+a19kimgr.pdf (D120521787)	1
SA	**classification.pdf** Document classification.pdf (D51467442)	1
SA	**46500_FinalReport_KristineKrieger.pdf** Document 46500_FinalReport_KristineKrieger.pdf (D60626450)	1

Appendix B: Ethical Approval Letter

Strathmore UNIVERSITY

23rd March 2022

Mr Mwathi Brian,
brian.mwathi@strathmore.edu

Dear Mr Mwathi,

RE: A Localized Rainfall Prediction Model for Maize Planting using Recurrent Neural Networks

This is to inform you that SU-IERC has reviewed and **approved** your above **SU masters'** research proposal. Your application reference number is **SU-IERC1288/22.** The approval period is **23rd March 2022 to 24th March 2023.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-IERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-IERC within 48 hours of notification
iv. Any changes, anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-IERC within 48 hours
v. Clearance for export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days upon completion of the study to SU-IERC.

Prior to commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

for: **Dr Ben Ngoye,**
Secretary; SU-IERC

Cc: Prof Fred Were,
Chairperson; SU-IERC

STRATHMORE UNIVERSITY INSTITUTIONAL
ETHICS REVIEW COMMITTEE
(SU-IERC)
23 Mar 2022
Email:ethicsreview@strathmore.edu
P.O BOX 59857-00200
NAIROBI-KENYA

Appendix C: Sample historical climate csv data

	A	B	C	D	E	F	G	H	I	J	K	L
1	-BEGIN HEADER-											
2	NASA/POWER CERES/MERRA2 Native Resolution Daily Data											
3	Dates (month/day/year): 01/01/2000 through 01/01/2022											
4	Location: Latitude -1.5177 Longitude 37.2634											
5	Elevation from MERRA-2: Average for 0.5 x 0.625 degree lat/lon region = 1327.62 meters											
6	The value for missing source data that cannot be computed or is outside of the sources availability range: -999											
7	Parameter(s):											
8	T2M	MERRA-2 Temperature at 2 Meters (C)										
9	PS	MERRA-2 Surface Pressure (kPa)										
10	WS10M	MERRA-2 Wind Speed at 10 Meters (m/s)										
11	QV2M	MERRA-2 Specific Humidity at 2 Meters (g/kg)										
12	PRECTOTCORR	MERRA-2 Precipitation Corrected (mm/day)										
13	-END HEADER-											
14	YEAR	DOY	T2M	PS	WS10M	QV2M	PRECTOTCORR					
15	2000	1	20.19	86.59	2.91	12.08	0.04					
16	2000	2	20.49	86.56	2.99	12.7	0.2					
17	2000	3	20.06	86.54	3.21	12.51	0.43					
18	2000	4	19.52	86.54	3.3	11.6	0.02					
19	2000	5	19.9	86.59	3.04	11.29	0.23					
20	2000	6	20.46	86.59	2.73	11.84	0.04					
21	2000	7	20.44	86.51	3.16	11.66	0.08					
22	2000	8	20.74	86.47	3.39	10.86	0.01					
23	2000	9	20.91	86.49	3.36	11.35	0					
24	2000	10	20.3	86.54	3.49	10.19	0					
25	2000	11	20.51	86.6	3.09	10.99	0.31					
26	2000	12	21.08	86.55	3.59	11.78	0.4					
27	2000	13	20.68	86.56	3.8	11.84	0.77					
28	2000	14	19.86	86.57	3.59	9.83	0.21					

machakos-county-2000-2022

Appendix D: Python program (Model Development)

```
import pandas as pd
from tensorflow import keras
from datetime import datetime
import matplotlib.pyplot as plt
import seaborn as sns
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error
from numpy import abs, array, triu, ones_like
from math import sqrt
import pickle


# ### Get historical data from the NASA POWER PROJECT API
# ##### Data is collected from: 1st Jan 2000 through 19th March 2022

url =
"https://power.larc.nasa.gov/api/temporal/daily/point?start=20000101&
end=20220101&latitude=-
1.5177&longitude=37.2634&parameters=T2M,PS,WS10M,QV2M,PRECTOTCORR&com
munity=AG&format=csv"

csv_path = keras.utils.get_file(fname="machakos-county-2000-
2022.csv", origin=url)


def parse_date(x):
    return datetime.strptime(x, '%Y %j')


# Skip the CSV description rows
df = pd.read_csv(csv_path, skiprows=13, parse_dates={'date': ['YEAR',
'DOY']}, date_parser=parse_date, skipinitialspace=True, index_col=0)


df.tail()


# Get column names
df.columns


# Remove empty values
```

```
indexes_to_drop = df.index[df['T2M'] == -999.00]
df.drop(indexes_to_drop, inplace=True)


# Check for outliers
sns.boxplot(x=df['PRECTOTCORR'])


# Visualize trends
df.plot.line(figsize=(12, 4), subplots=True)
# plt.savefig("Sub plot", dpi=1200)


# ### Check for multicollinearity among predictor variables


correlation = df.iloc[:, :-1].corr()


# Generate a mask for upper traingle
#
mask = triu(ones_like(correlation, dtype=bool))

cmap = sns.diverging_palette(230, 20, as_cmap=True)

sns.heatmap(correlation, annot=True, mask=mask, cmap=cmap)


# split multivariate sequence into samples
def to_supervised(sequences, n_steps_back, n_steps_future):
    X, y = list(), list()
    for i in range(len(sequences)):
        # find the end of the pattern
        end_index = i + n_steps_back
        out_end_index = end_index + n_steps_future - 1
        # check if index is out of bound
        if out_end_index > len(sequences) - 1:
            break
        # gather input and output parts of the pattern
        seq_x, seq_y = sequences[i:end_index, :-1],
sequences[end_index-1:out_end_index, -1]
        X.append(seq_x)
        y.append(seq_y)
    return array(X), array(y)



# Normalization
```

```
scaler = MinMaxScaler(feature_range=(0,1))

df_scaled = scaler.fit_transform(df)



def build_model(X, y):
    model = Sequential()
    model.add(LSTM(64, activation='relu', input_shape=(X.shape[1],
X.shape[2]), return_sequences=True))
    model.add(LSTM(32, activation='relu', return_sequences=False,
recurrent_dropout=0.2, unroll=True))
    model.add(Dropout(0.2))
    model.add(Dense(y.shape[1]))
    model.compile(optimizer='adam', loss='mse')
    model.summary()
    return model

training_history = []
models = []
forecast_days = [3,5,7,30]
y_preds = []
train_test_data = []

for index, future_days in enumerate(forecast_days):
    # Convert data to supervised
    X, y = to_supervised(df_scaled, n_steps_back=60,
n_steps_future=future_days)
    # split to train test
    x_train, x_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=0)
    # Append data
    train_test_data.append((x_train, x_test, y_train, y_test))
    # Build model
    model = build_model(x_train, y_train)
    # Fit model
    history = model.fit(x_train, y_train, epochs=60, batch_size=64,
validation_split=0.3, verbose=1)
    # Append histories
    training_history.append(history)
    # Append model
    models.append(model)

for index, model in enumerate(models):
    model.save(f'rainfall-model-{forecast_days[index]}-day')
```

```
# Load models
for index, forecast in enumerate(forecast_days):
    model = keras.models.load_model(f'rainfall-model-{forecast}-day')
    models.append(model)


for i , history in enumerate(training_history):
    plt.subplot(2, 2, i+1)
    plt.gcf().set_size_inches(15,12)
    plt.title(f'{forecast_days[i]} day forecast performance')
    plt.plot(history.history['loss'], label='Training loss' )
    plt.plot(history.history['val_loss'], label='Validation loss')
    plt.xlabel('Number of epochs')
    plt.legend()


from numpy import zeros


def reverse_y_scaler(data):
    data = data.ravel() # convert from 2D array into 1D array
    # create a new numpy array
    shape = (data.shape[0], 5)
    n_array = zeros(shape)
    n_array[:, -1] = data
    return n_array


type(train_test_data)


# ## Evaluation Metric
# Evaluate each time step separately in order to:
# * Comment on the skill at a particular lead time (1 day vs 3 day)
# * Contrast models based on their skills at different lead times
(models good at +1 day vs models good at days +5)


from sklearn.metrics import r2_score


# def coeff_determination(y_true, y_pred):
#     SS_res = K.sum(K.square(y_true - y_pred))
#     SS_tot = K.sum(K.square(y_true - K.mean(y_true)))
#     return (1 - SS_res/(SS_tot + K.epsilon()))


model_performance = {}
```

```
inverse_y_test_pred = []
# Perform fit for each model
for (index, model) in enumerate(models):
    X_test = train_test_data.__getitem__(index)[0]
    y_test = train_test_data.__getitem__(index)[2]
    #  get prediction
    yhat = model.predict(X_test, verbose=0)

    print(y_test.shape)
    print(yhat.shape)

    inv_yhat = scaler.inverse_transform(reverse_y_scaler(yhat))[:, -
1]
    inv_y_test =
scaler.inverse_transform(reverse_y_scaler(y_test))[:, -1]
    inverse_y_test_pred.append((inv_y_test, inv_yhat))
    mse = mean_squared_error(inv_y_test, inv_yhat)
    rmse = sqrt(mean_squared_error(inv_y_test, inv_yhat))
    mae = mean_absolute_error(inv_y_test, inv_yhat)
    R2 = r2_score(inv_y_test, inv_yhat)

    model_performance[f'{forecast_days[index]} day'] = {
        'mean_squared_error': mse,
        'root_mean_squared_error': rmse,
        'mean_absolute_error': mae,
        'R2': R2
    }




[print(key,':',value) for key, value in model_performance.items()]


for i , y_val in enumerate(inverse_y_test_pred):
    actual = y_val[0]
    predicted = y_val[1]
    print(actual[:forecast_days[i]])
    print(predicted[:forecast_days[i]])
    plt.subplot(2, 2, i+1)
    plt.gcf().set_size_inches(15,12)
    plt.title(f'{forecast_days[i]} day forecast performance')
    plt.plot(list(range(0,forecast_days[i])),
actual[:forecast_days[i]], color='g', label='actual')
    plt.plot(list(range(0,forecast_days[i])),
predicted[:forecast_days[i]], color='r', label='predicted')
    plt.xlabel('Number of days')
```

```
    plt.ylabel('Rainfall in mm/day')
    plt.legend()


for i , y_val in enumerate(inverse_y_test_pred):
    actual = y_val[0]
    predicted = y_val[1]
    print(actual[:forecast_days[i]])
    print(predicted[:forecast_days[i]])
    plt.subplot(2, 2, i+1)
    plt.gcf().set_size_inches(15,12)
    plt.title(f'{forecast_days[i]} day forecast performance')
    plt.plot(list(range(0,99)), actual[:99], color='g',
label='actual')
    plt.plot(list(range(0,99)), predicted[:99], color='r',
label='predicted')
    plt.xlabel('Number of days')
    plt.ylabel('Rainfall in mm/day')
    plt.legend()


pickle.dump(scaler, open('scaler.sav', 'wb'))
```