KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

DESIGN AND IMPLEMENTATION OF A SMART IRRIGATION SYSTEM POWERED BY DEEP LEARNING ALGORITHMS

By

Akola Mbey Denis

Louis Marie Atoluko Ayariga

SUPERVISOR: Eur Ing. Benjamin Kommey

CO-SUPERVISOR: Dr. Eliel Keelson

MAY 2019

LIST OF ABBREVIATIONS

Abbreviations	Meaning
IoT	Internet of things
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
RSME	Root Mean Squared Error
MAPE	Mean Absolute Percentage Error
CRBM	Conditional Restricted Boltzmann Machines
MAE	Mean Absolute Error
LED	Light Emitting Diode
LCD	Liquid Crystal Display

DECLARATION

We hereby declare that except for specific references that have been properly acknowledged, this work is the result of our own research and it has not been submitted in part or whole for any other degree elsewhere.

Signature ………………………………… Date ……………………………………

Akola Mbey Denis (Candidate)

Signature ………………………………… Date ……………………………………

Louis Marie Atoluko Ayariga (candidate)

Signature ………………………………… Date ……………………………………

Eur. Ing Benjamin Kommey (Supervisor)

Signature ………………………………… Date ……………………………………

Dr. Eliel Keelson (Co-Supervisor)

ABSTRACT

Agriculture contributes greatly to the economy of Ghana. According to a report from the Food and Agriculture Organization (FAO) of the United Nations in 2012, Agriculture employs 53.33% of the Ghanaian population. Agriculture, therefore, is the source of livelihood for most people in Ghana.

However, over the last few years, the climatic conditions in Ghana have not been stable, and consequently, it has affected the rainfall pattern in Ghana as many farmers in the country solely rely on rainfall to carry out their farming activities. Notwithstanding, a few farmers used artificial irrigation systems to combat the erratic nature of the rainfalls, however which waste a lot of water. Water is becoming a scarce resource and as the human population is increasing rapidly, there is the need for us to exploit methods to converse water for future use.

The focus of this project is to design and implement a novel smart irrigation system that is powered by deep learning algorithms that can mitigate the problems associated with the current irrigation systems in use, thus conversing water and use rainfall prediction information to decide to irrigate a farm.

DEDICATION

This project is dedicated to our project supervisors and to our parents for their tremendous support. God richly bless them.

Table of Content

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

1.1. BACKGROUND OF STUDY

In recent years, Ghana has been experiencing an erratic rainfall pattern. Agriculture activities in Ghana heavily depend on rainwater. Due to the uncertain rainfall pattern in the country, it has affected the agriculture industry, thus, the poor yield of food crops in the last two years.

However, large scale farmers have adopted mechanical irrigation systems to provide constant water to their farms. These mechanical irrigation systems are not efficient in their use of water because most of these systems pump large volumes of water into crop farms beyond the soil water requirements. A recent report by [1] revealed that water is becoming a scarce resource and thus there is the need for us to make judicious use of water to converse water for the future generation. Due to the inefficient nature of the existing irrigation solutions in Ghana, a lot of water is wasted in farm irrigation.

This project seeks to employ deep learning to build a model to predict the amount of rainfall for the next day for the city of Kumasi in Ghana. The information about the chances of rainfall can be used in smart irrigation systems to supply the right quantity of water to farms by getting information about the soil moisture content from sensors deployed on the farm.

1.2. Problem statement

Currently, the artificial irrigation systems in use in Ghana are manual operated, however, there are a few automated irrigation systems. Farmers use the manual irrigation systems pump water into their crop farms especially during drought periods and also during the dry season which occurs from December through February every year in Kumasi. Also, the smart irrigation systems in use, measure the soil moisture content using soil moisture sensors they deploy on the farm. The irrigation algorithm in these systems pumps water into the farm when the measured soil moisture content of the farm is less than some threshold.

There are cases where farmers irrigate their farms, but after a few hours, rain happens to fall. Since these systems don't have future information about rainfall, under such conditions, these systems waste water. Therefore, the current irrigation systems in Ghana still waste water which is a delicate resource.

1.3. PROJECT OBJECTIVES

This project seeks to achieve the following:

- Predict the rainfall pattern using Deep Learning Algorithms
- Apply the rainfall information to build an efficient and smart irrigation system.
- Build a mobile application to present rainfall prediction and measured farm sensors data to the farmer.

1.4. SIGNIFICANCE OF THE STUDY

The use of smart irrigation systems on farms to converse water and prevents the wastage of water is important today because water is becoming a scarce resource. This study is therefore is an important one as it seeks to build a novel smart irrigation system to converse water and ensure there is a continual existence of water for agricultural activities.

1.5. ORGANIZATION OF THE STUDY

The remaining chapters are organized as follows:

Chapter 2 provides an overview of some existing smart irrigation systems, their system design, and some shortcomings of such implementation. Also, this chapter is a review of the existing methods used to predict rainfall.

Chapter 3 presents the design and implementation of our proposed smart irrigation system. The details of the system architecture, schematic diagrams, rainfall prediction model, software development, and hardware design are also provided in this section.

Chapter 4 presents the results and discussion, and finally the conclusions and recommendations to be taken in the advance study of this project.

Chapter 2: LITERATURE REVIEW

2.1. Introduction

The agricultural sector of Ghana employs almost 40% of the Ghanaian populace [2]. The major occupation of the people in Kumasi and its environs is mainly farming. The crops mostly grown in and around Kumasi are plantain, cocoyam, maize, cassava, various cereal crops, and legumes. The problem that affects agricultural activities in Kumasi is the inconsistent rainfall pattern farmer's experience. Most farmers in Kumasi rely on rainwater to carry out their farming activities. However, for the past decade, the rainfall pattern in Kumasi has not been stable and it has affected the yield of food crops from those years until now. In Kumasi, there are typically two seasons (i.e. dry and wet seasons), the dry season spans from Mid-November through to March and the dry season begins in Mid-November and ends in February or March depending on the prevailing climatic conditions.

Most of the information detailed in this section share a similar background to smart irrigation systems and rainfall prediction. In this section, we present the different approaches used by various researchers to the built smart irrigation systems that converse water in farmlands. Also, in retrospect is the work of various authors on how weather data could be used in control systems to effectively and efficiently provide a controlled amount of water supply to agricultural farms.

With the latest improvements in technology particularly in the Internet of Things (IoT) and artificial intelligence, various authors have demonstrated various methods of building smart irrigation at the minimum cost. This section is sub-divided into previous work done on rainfall prediction and smart irrigation systems.

2.2. Related work on Rainfall Prediction

Traditionally, weather predictions are performed using complex models of physics that run on high-performance computing (HPC) environments. Most of such computing infrastructure is expensive. Even apart from the expensive nature of such computing devices, there are often errors in their forecast which is largely due to the incomplete understanding of the atmospheric processes [3].

In [4] the authors employed the use of linear regression algorithms to predict the rainfall for various seasons in India. They used a dataset that consisted of the previous year's seasonal rainfall information. Linear regression is a technique for predicting the value of a dependent variable from an independent variable when there is a relationship between the variables described by a linear model. The cost function of a typical linear regression is stated in the form:

$$J(\theta) = \frac{1}{2m} \sum_{j=1}^{n} (k_{\theta} a^{j} - b^{j})^{2}$$

Where m is a number of the sample taken. k_θ variant are 0, 0.5, 1, 1.5, 2; a, b are the variables;

The authors sought to help farmers with the results of their findings to make informed decisions on how to harvest their crops based on the various crop seasons in India. However, the findings of their research showed that there was a weak correlation between the various crop seasons (Rabi, Zaid, and Kharif).

Afan et al [5] exploit the use of various deep learning models such as Recurrent Neural Networks

(RNNs), Conditional Restricted Boltzmann Machines (CRBMs), and Convolutional Neural Networks (CNNs) on time-series weather data that they obtained from Indonesia Agency for Meteorology, Climatology, and Geophysics. Recurrent Neural Networks are a type of neural network in which the output of the previous step is passed as input to the next step. RNN can remember previously learned data and a result has been adopted in time-series predictions. However, RNNs suffer from the vanishing gradient problem which affects their performance. Conditional Restricted Boltzmann Machines are rich probabilistic models that can learn a probability distribution over its sets of inputs. Convolutional Neural networks are a class of neural networks that are predominately used in image recognition and classification. The dataset they used obtained contained weather data over three decades (from 1973 to-2009) from some weather stations in the Aches area in Indonesia. The parameters in the dataset included mean temperature, minimum temperature, precipitation temperature relative humidity, mean sea level pressure, mean station pressure, visibility, average wind speed, maximum wind speed, and rainfall. The authors used the weather dataset to train the RNN, CRBM, CNN models and they compared the performance of the various models. Figure 1.1 shows the block diagram used by the authors in their research activity.

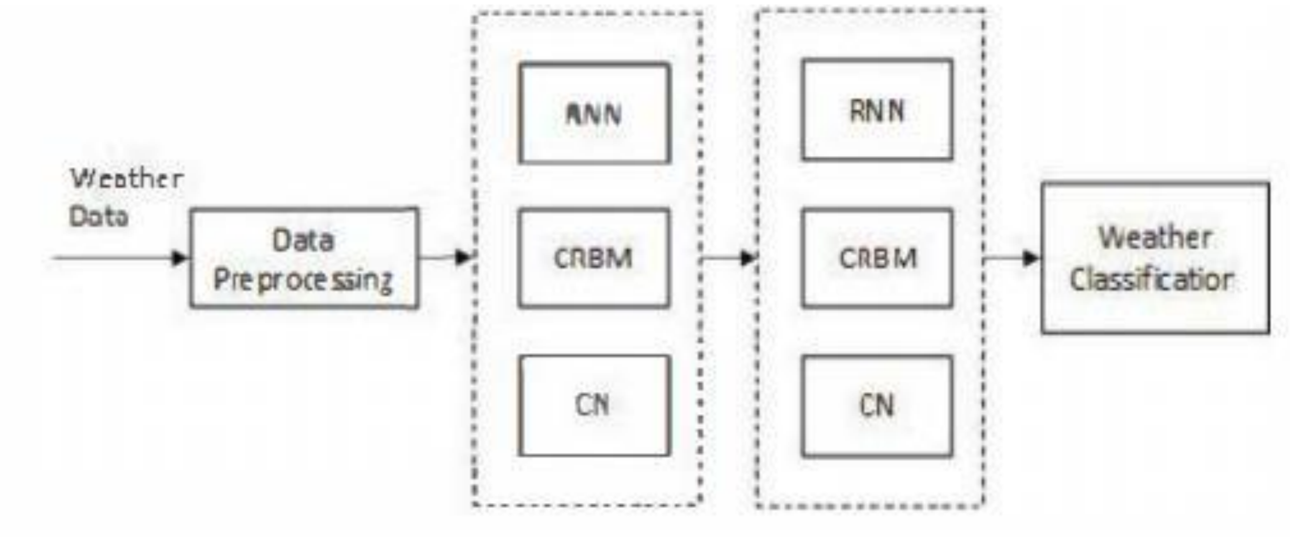


Figure 2:1Block Diagram used by [3]

The forecasted variable in the networks was rainfall. The results they obtained showed than RNN performed better with a good rainfall prediction accuracy.

Furthermore, according to [6] using neural networks have the potential to achieve a better result in rainfall fall prediction. These authors used recurrent neural networks and Long Short Term Memory (LSTM) networks to predict the set of hourly rainfall levels. However, they compared the performance of their RNN and LSTM models on the radar dataset to that of Random Forest and XGBoost classifiers. Long Short Tem Memory Networks is a kind of recurrent neural network that is capable of learning long term dependencies in memory. LSTM can remember information and data for long-time periods. An added advantage to LSTMS is that they do not suffer from vanishing gradient problem present in Recurrent Neural Networks. LSTMs employs a backpropagation algorithm in training models. Random Forest Classifier is a supervised learning algorithm that generates a forest with less number of trees. The random forest Classifier creates decision trees on randomly selected data samples, gets a prediction for each tree, and chooses the best solution utilizing majority voting. However, XGBoost Classifier is one of the popular supervised machine learning algorithms. XGBoost stands for Extreme Gradient Boosting It is used on structured data and it has good performance as compared to the Random Forest Classifier. The motivation for using XGBoost Classifier is its speed and performance. The parameters that were in the dataset are identification number, minutes past the top of the radar observation, the distance of the gauge of the radar whose observations are being reported, the radar reflectivity in Kilometers, Maximum reflectivity in the vertical column above gauge in dBZ, correlation coefficient, Differential reflectivity in dB and the Specific differential phase (deg/km). Their research showed that the use of these neural networks improved the rainfall forecast performance significantly.

The result of their work showed that neural networks outperformed XGBoost and Random Forest Classifiers. However, neural networks require efficient training to make accurate predictions.

Aswin S et al. [7] sought to predict the rainfall pattern using convolutional neural networks (CNNs) and Long Term Short Memory Networks. They obtained their datasets from Global precipitation Project which contained the records of rainfall data from July 1979 through to January 2018. In training convolutional networks, the data was passed through three convolutional layers with filters (kernel), pooling, and the fully connected layer. The convolutional layer of CNNs is used for feature extraction and as a result, produces a feature map. Notwithstanding, the size of the convolutional layer is determined by three parameters; the depth, stride, and zero padding. This layer also contains an activation function that is used to introduce non-linearity in the dataset. A typical activation function is a sigmoid function. The pooling layer is used to reduce the dimensionality of each output from the convolutional layer. In this work, the authors used max-pooling a type of spatial pooling to extract the largest element from the rectified feature map. The fully connected layer is a multilayer perceptron used for the vectorization of the feature map matrix. It has an activation function such as sigmoid or softmax used to classify the outputs. The proposed architecture by the authors split the data into training and testing datasets and passed the training set data to the LSTM network and the CNN for training and similarly the test dataset was used to test the prediction accuracy of the two trained models. The Root Mean Squared Error

(RMSE) and the Mean Absolute Percentage Error (MAPE) of CNN and LSTM were compared. The results of their experimentation proofed the CNN performance in predicting rainfall was promising that LSTM. However, they observed that increasing the number of hidden layers in both CNN and LSTM, the MAPE and RMSE errors kept reducing, and a fine accuracy is obtained. The flowchart diagram of their experiment is shown in figure 1.2 below

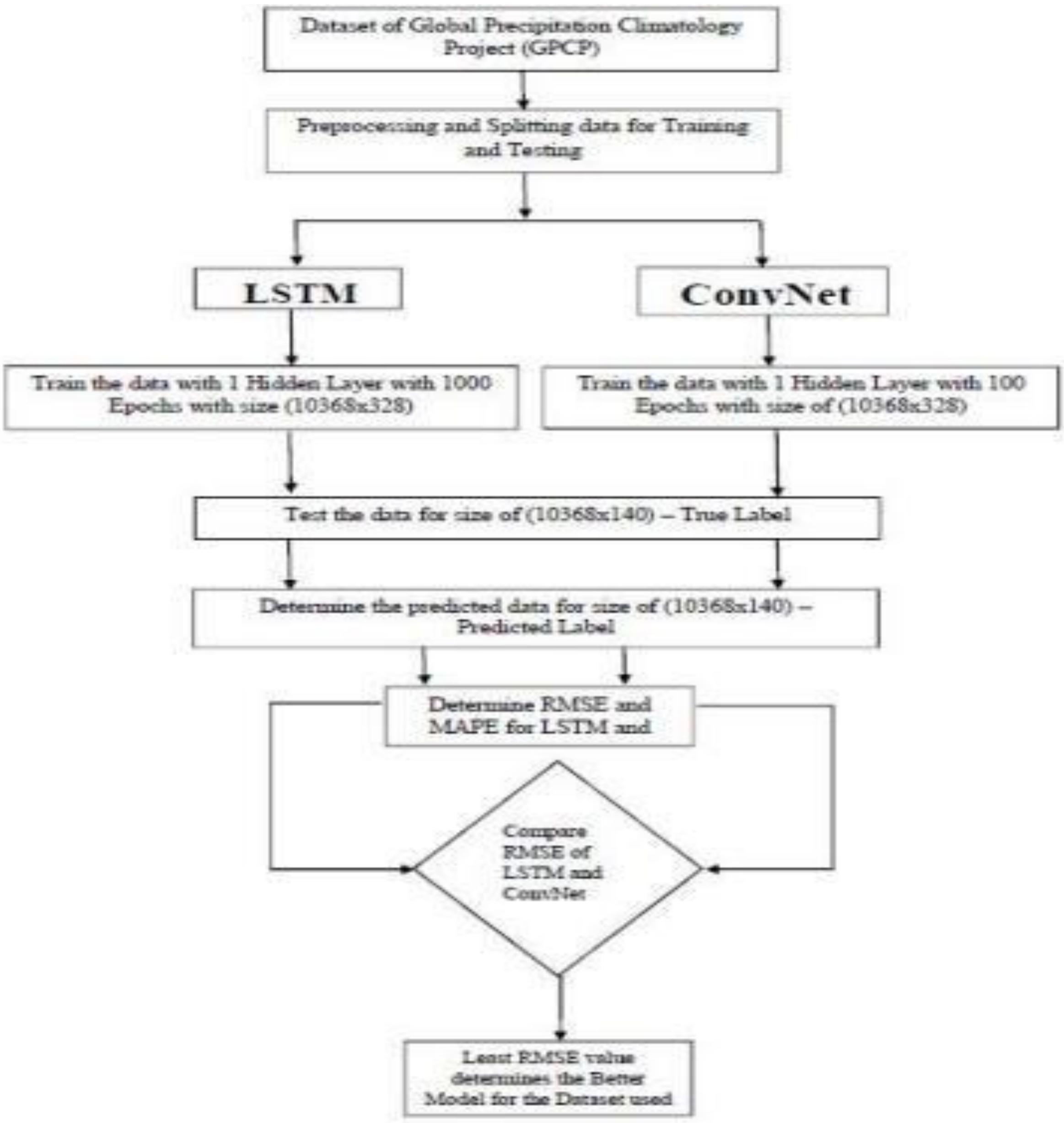


Figure 2:2Flowchart of the comparison of LSTM and CNN in rainfall prediction by [7]

The authors in [8] exploit the use of the Convoluted LSTM, ROVER, and FC-LSTM network to predict the future rainfall intensity within a very short time (0-6 hours) using radar weather dataset that was collected in Hong Kong, China from 2011 to 2013. In their work, they formulated the precipitation nowcasting as spatiotemporal sequence forecasting problem in which both input and output are spatiotemporal sequences. Convoluted LSTM was used to mitigate the redundancy problem of spatial data in the fully-connected LSTM. They built a model that incorporated multiple-layer convolutional LSTM to solve the precipitation nowcasting problem. In their experimentation with the models (FC-LSTM, ROVER, and CLSTM), they discovered data the Convoluted LSTM outperformed the FC-LSTM and state-of-art ROVER model in handling spatiotemporal correlations, deeper models can produce better results using fewer parameters.

Finally, in [9], the authors understudied the use of Echo State Networks and Deep Echo State Networks also known as reservoir computing to predict rainfall in Southern Taiwan. These algorithms were employed because they were speedy and effective for processing large datasets like the continuous stream of data from weather stations. However, they compared the performance of the Deep Echo State Network to that of some standard algorithms in MatLab Neural Network toolbox (BPN, SVR) as well as ECMWF which is the European Centre for Medium-Range Weather Forecasts. The group obtained their hourly meteorological data from Tainan Observatory in southern Taiwan for years spanning from 2002 through to 2014. The focus of their research was to predict the rainfall and explore the strong factors relevant in predicting rainfall in Southern Taiwan. In the cause of their experimentation, they used some performance metrics such as the accuracy, RSME, Normalized Root Mean Square Error (NRSME), and the correlation coefficient as measures of comparison. The result of their experiment with the different models showed that Deep Echo Networks outperformed ESN and the commercial neuronal network algorithms in Matlab (BPN, SVR). Thus, they discovered it was promising to use Deep Echo State Network to predict rainfall.

2.3. Related work on smart irrigation systems

According to [10], primitive methods had been exploited to build irrigation systems that eased the farmer from the laborious work of watering their farm crops using manual watering tools as the watering can and hose pipes.

In [11], the authors deployed wireless sensors on a farm to monitor some relevant conditions such as temperature, soil moisture content, humidity, and water level of the agricultural land for controlling the amount of water to be pumped to irrigate the farm. The authors leveraged cloud servers to store the continuous real-time sensed data they obtained from the sensors network on the farm. The proposed system by these authors used a combination of a wireless sensor network, IoT communication technology, and cloud server to accomplish the performance and data storage requirements of the proposed irrigation system they designed. Their system provided remote monitoring and control of irrigation systems with real-time sensing of atmospheric and soil conditions like temperature, humidity, and soil moisture. The flow chart diagram explaining their design is shown below in figure 1.3.

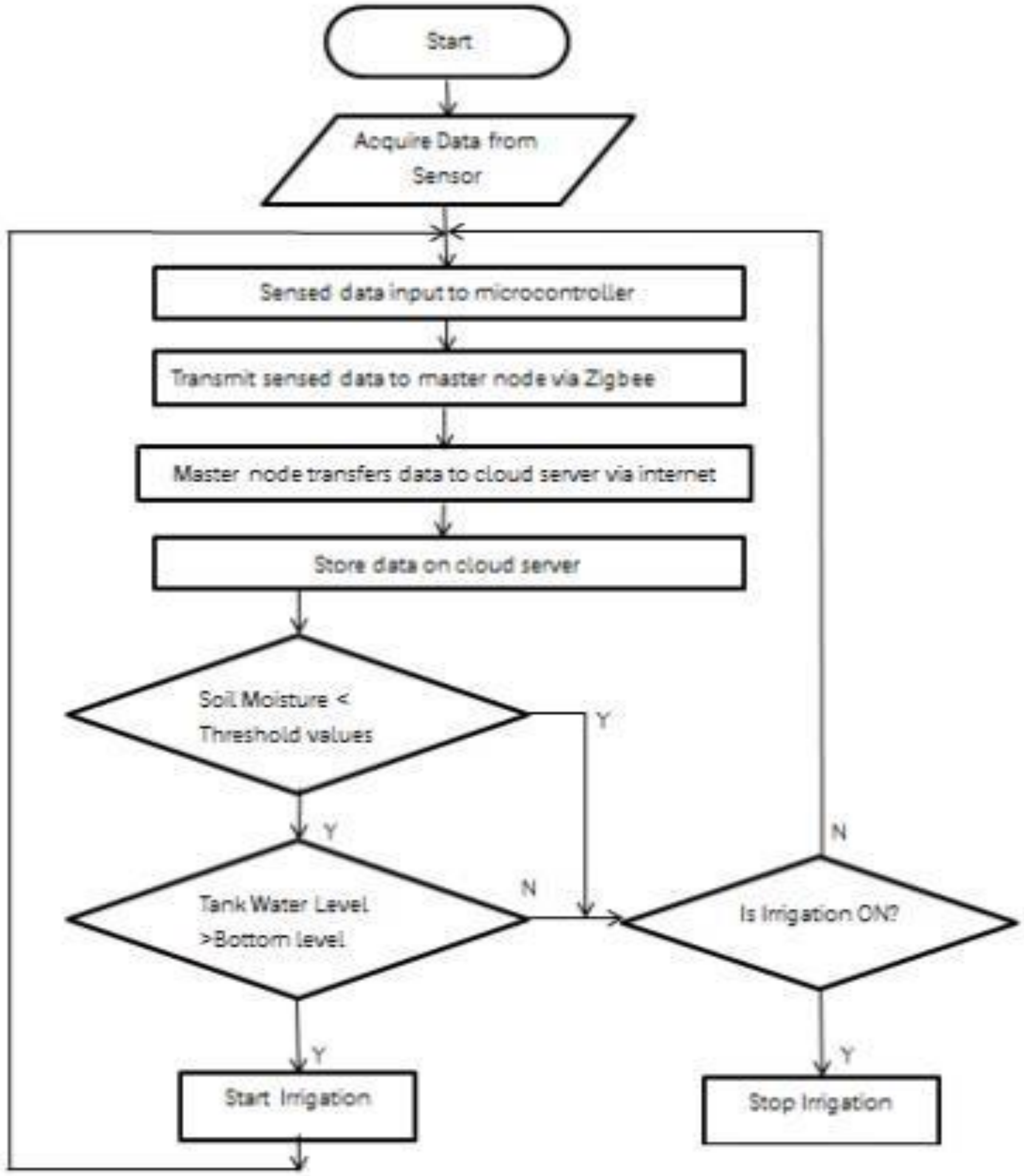


Figure 2:3 Irrigation algorithm proposed by [11]

However, their work could not take advantage of rainfall to converse water in their proposed smart irrigation system

Aashika Premkuma et al**.** [12] used Arduino for moisture sensing and controlling the water supply to the farm. A Node Microcontroller Unit (MCU) to send notifications about the status of the designed irrigation system to the farmer through mobile communication. However, the system did not take advantage of rainfall pattern information to control the amount of water that was supplied to the farm, and thus, it did not converse water efficiently since the farm could be irrigated on a day where there could be rainfall which could have supplied water to the farm to meet its soli water requirements.

In [13], the authors proposed a LoRa-based smart irrigation system. In their system, the irrigation node is mainly composed of the LoRa communication module, solenoid valve, and a hydroelectric generator. LoRa communication is a long-range low power communication module. It uses the spread spectrum modulation derived from Chirp Spread Spectrum (CSS) technology. The LoRa communication module can be operated at the ISM frequency band of 433 MHz, and 915MHz. In this paper, the authors made a significant improvement over the work of previous authors, because they used a low power and long-range communication module which could suffice for long-distance communication from the irrigation systems which could be located miles away and the farmer. However, the system was not incorporated with sensors such as the soil moisture sensor to determine if the farm needed to be irrigated. Though the proposed system automated the watering process, it did not pump the right quantity of water that is needed to irrigate the farm which can the obtained if sensors were deployed on the farm, and thus, the system did not converse water. A diagram of their system architecture is shown in figure 1.4 below

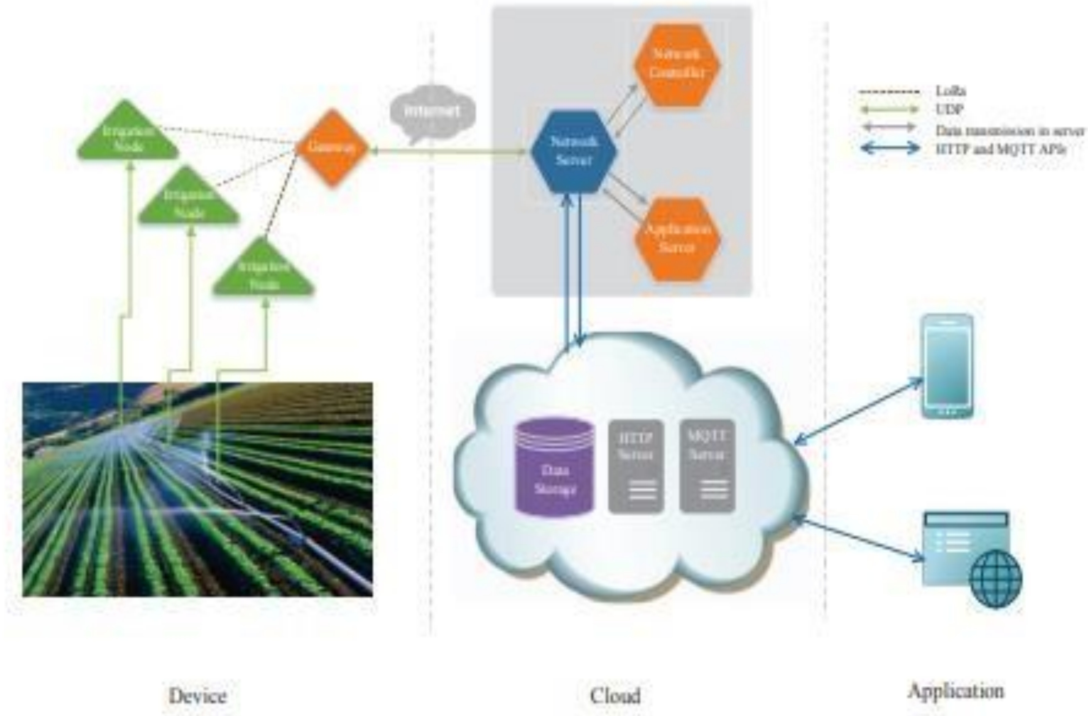


Figure 2:4 System architecture proposed by [12]

Finally, Chao Shang et al. [14] developed a data-driven robust closed-loop control system to intelligently monitor the process of irrigation using uncertainty learning and data analytics. Prediction science is not entirely deterministic and as a result, some degree of uncertainty must be embraced when using prediction based algorithms to solve problems. To reduce the level of uncertainty in predictions, uncertainty learning is applied where different algorithms are used to learn from the same data so that the algorithm that fits the data well is selected as the optimal solution to the problem. Data analytics is the process of inspecting, processing, and modeling raw or semi-processed data to gain insights and make decisions from such insights. In this work, they integrated mechanistic models that described the soil moisture content variations and data-driven models which characterized the uncertainty in forecast errors of evapotranspiration and precipitation. They observed that integrating the outputs of their models had the potential to serve as a control framework to control the automatic irrigation of farms. To formulate their mechanistic model, they used water balance models to describe the dynamics of soil moisture levels. They also took advantage of real-time forecasts and developed data-driven uncertainty sets that accurately described the distribution of uncertain forecast errors. From their research, they realized that the uncertainty prediction errors emanated from evapotranspiration and precipitation. Therefore, they adopted a dynamic model to account for the soil water balance, and a mathematical relation for the water balance was derived from figure 1.5 below.

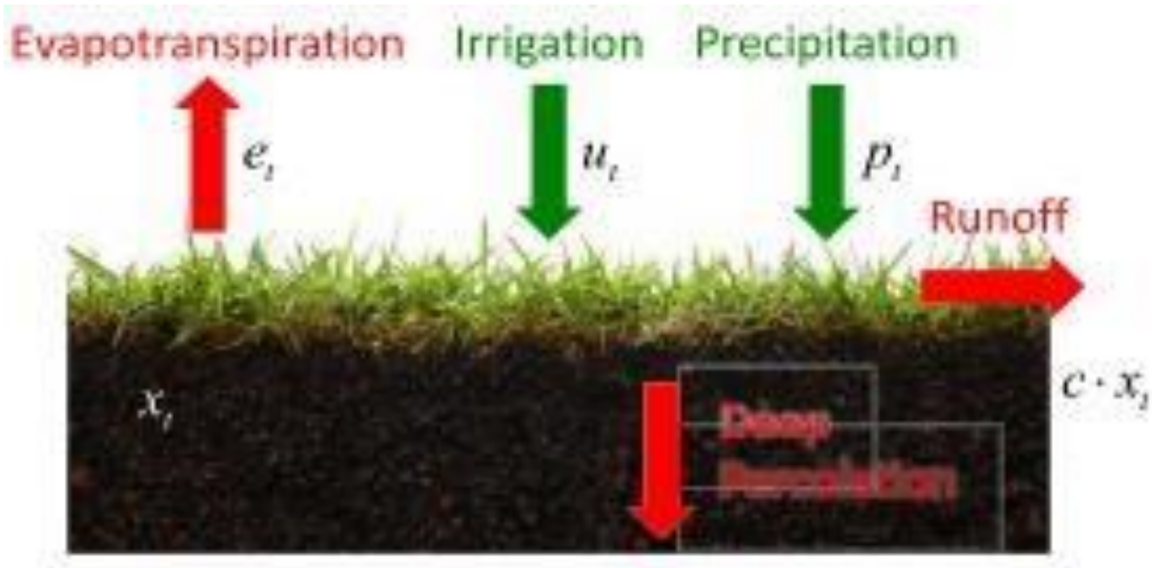


Figure 2:5Diagram for the derivation of the quantity of soil water proposed by [12]

The mathematical relation is a first-order autoregression that describes the water balance in the root-soil zone. From the diagram above, it is clear that water loss in the soil is mainly through runoff, deep percolation, and evapotranspiration. The water inflow is mainly from rainfall and irrigation. The quantity of water in the soil which is denoted as x_{t+1} is related to the water inflow and outflow factors in the equation given below:

$$x_{t+1} = (1 - c) \cdot x_t + u_t - e_t + p_t$$

Where x_t denotes the amount of water in the soil, e_t and p_t are the cumulative evapotranspiration and precipitation respectively in period t, and u_t is the irrigation amount. In their experiment, they performed a closed-loop simulation-based study of real weather conditions data collected from

Des Moines, Iowa, USA to measure the performance of various control methods. In their research, the control goal was to maintain the soil water content above a safety level (x_{min}) of 10mm and maximum water supply (u_{min}) of 10mm. To establish the data-driven uncertainty sets in their closed-loop system, they collected both weather forecast data and measurement from the sensors network in the soil from May 2016 to October 2016 which yielded

729 scenarios in total for forecast errors and forecast errors of precipitation. The cxv solver in MatLab was adopted to solve the optimization problem in their model. Cxv is a Matlab-based modeling system used for convex optimization. An improvement in water conservation was achieved by the researchers. However, the amount of data used in running their simulation was small and the system is likely to fail when it is subjected to data of higher dimensionality and variability. Their approach was simulation-based and it does not apply machine learning methods to conserve water.

Chapter 3: DESIGN AND METHODOLOGY

The design of any complex systems begins with the design of small modules and then integrating the individual modules into the desired system. Therefore, we have adopted a bottom-up design approach. In this chapter, we present an overview of the system architecture and how the overall systems work to achieve our outlined objectives. We then break the system into three main subsystems, the hardware implementation, embedded system software, mobile application, and deep learning rainfall prediction model.

3.1. System Architecture

The network of sensors deployed on the farm is going to provide real-time data about the relevant parameters of the farm (e.g. the soil moisture) and the weather parameters (rainfall, humidity, wind speed, wind velocity) which are transmitted via a wireless communication module to cloud platform. The weather parameters are used by the deep learning model to predict the chances of rainfall. The soil moisture content collected from the soil moisture content sensor is used by our irrigation algorithm to determine if the soil water content is less than the threshold soil moisture content value and to determine the quantity of water need by the crop farm. The real-time farm measures, the status of irrigation pump or motor, and the rainfall prediction data are made available to the farmer through our mobile application. The system architecture diagram is shown in figure 3.1 below

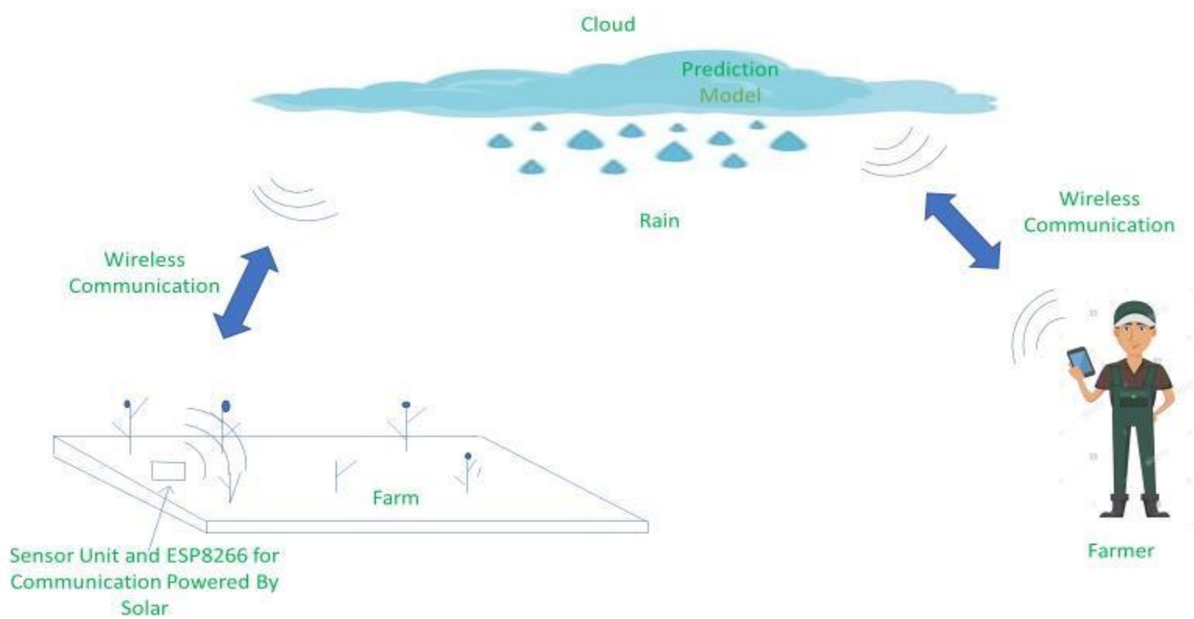


Figure 3:1 System architecture diagram

HARDWARE IMPLEMENTATION

3.1.1. Mobile application

The mobile application allows the farmer to monitor the irrigation process on the farm. The requirements of the mobile application were inspired by the objectives of the project. The mobile application provides an interface between our system and the farmer. This section describes both the functional and non-functional requirements of our mobile application. The section also presents the use case and activity diagrams and some of the application views. Moreover, we discuss some of the development tools we used.

3.1.1.1. Functional Requirements

The functional requirements of our mobile application give a detail description of what our application should do. In this section, we present the user requirements and other requirements the user may not directly interact with. These are a description of what our mobile application is expected to do.

These are the functional requirements of our mobile application:

- It should provide a login functionality to the user in our case a farmer. This functionality shall authenticate a user when they provide a valid email address and a password.
- It should provide a signup functionality to new users. A new use shall provide a valid email address and a password.
- It shall display a dashboard to successfully authenticated user on login. The dashboard shows real-time farm sensors measurements, the status of the irrigation pump or motor, and the rainfall predictions for a week.
- It shall provide a logout functionality to allow users to logout of the app.

3.1.1.2. Non-functional requirements

The non-functional requirements are not directly linked to the services provided by our mobile application. However, they are important since they are concerned with how easy it is for users to interact with our application.

The non-functional requirements of our application are:

- Performance: This refers to how fast our mobile application takes to respond to user inputs. Since our mobile application is to present real-time measured data from the farm, it should be able to do that with minimal delays.
- Security: To enforce security, only registered farms shall be able to use our application
- Ease of use: Our mobile application should be easy to use by farmers. Farmers shall be able to use our system to monitor their irrigated farms with or without little training. Our mobile

application uses a simple interface that allows users to easily navigate through the application,

3.1.1.3. Use case diagram

Figure 3.2 shows the use case diagram of our system. The use case diagram provides a visual representation of the relationship that exists between actors (internal or external entities that interact with our system) and our system. The system has two actors: the farmer who monitors his/her farm irrigation status and our server which provides real-time updates of the farm measurements obtained from the farm sensors and rainfall predictions.

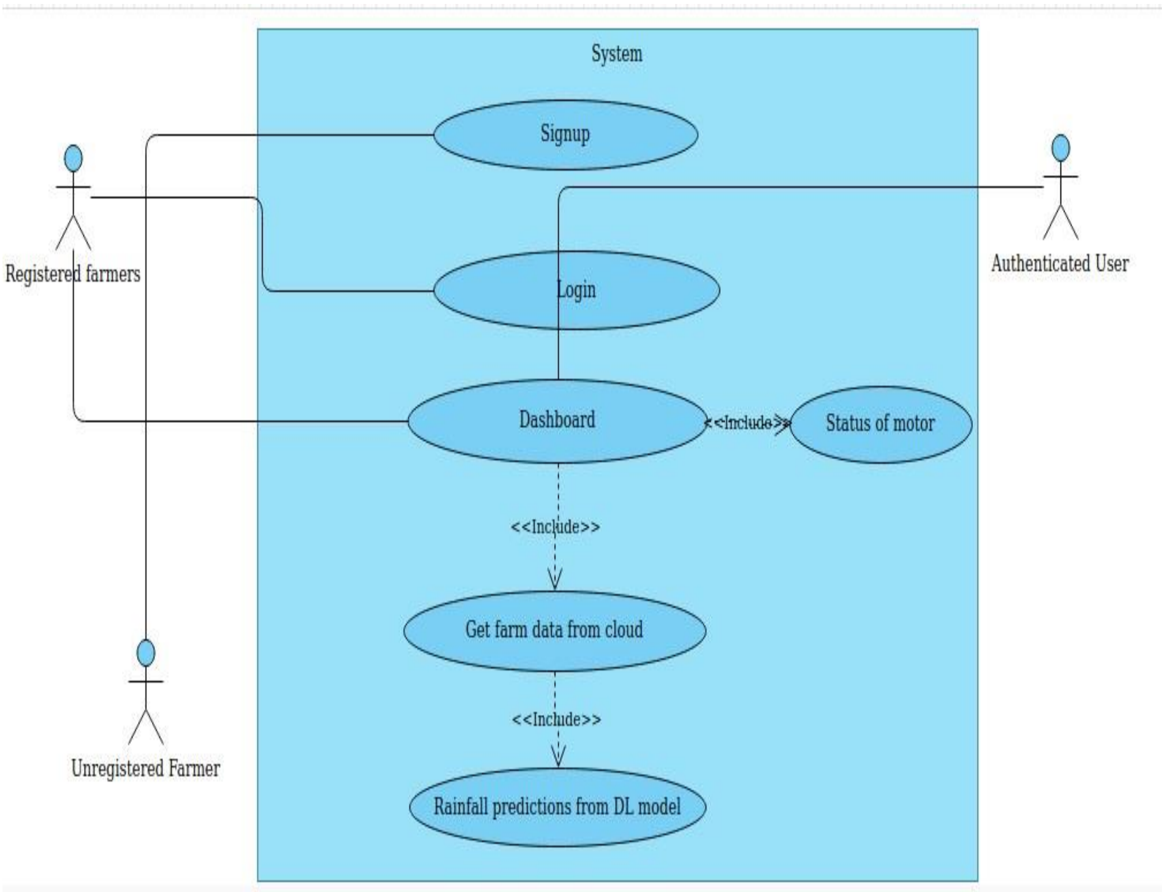


Figure 3:2Use case diagram for our mobile application

3.1.1.4. Login and authentication

When the application starts, it checks for an authenticated farmer, if an authenticated farmer exists, the farmer is directed to the dashboard of the application. However, in the absence of an authenticated user, the user in our case the farmer is redirected to the login view, where the user is provided with fields to enter his/her credentials to be granted access to his or her account and to use the application.

To successfully log in to our application, one is required to authenticate oneself with a valid email address and a matching password. Figure 3.3 shows the login view of our application.

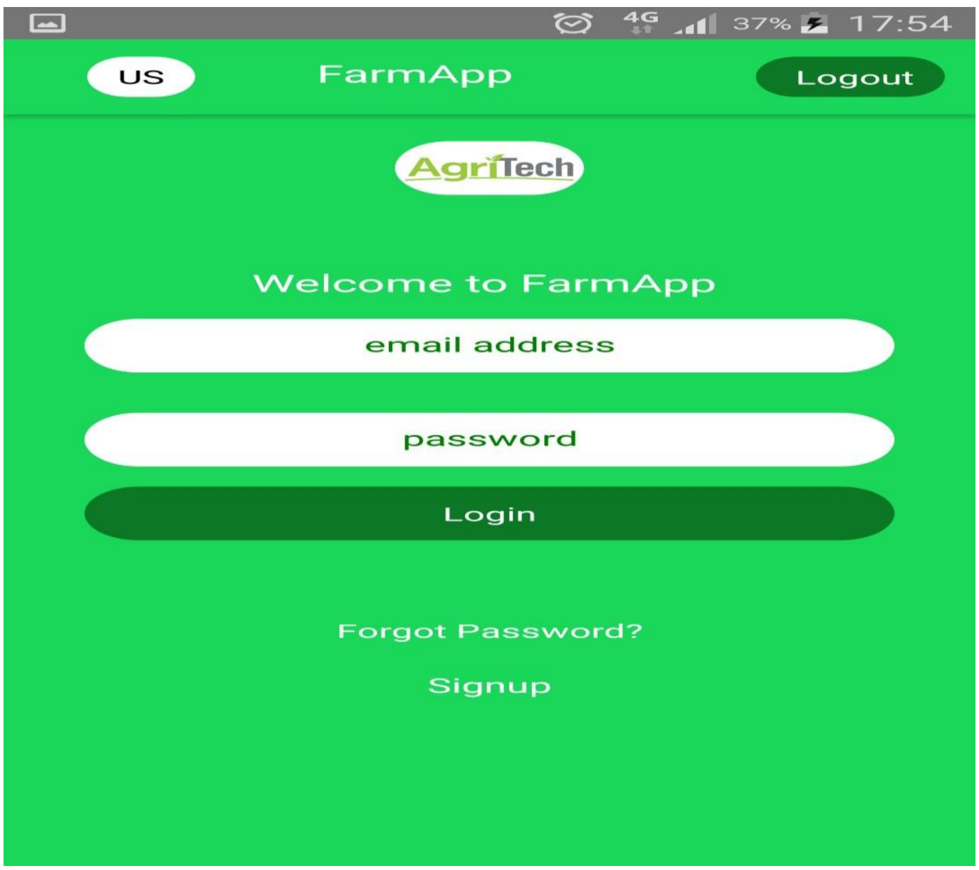


Figure 3:3Mobile application login screen

3.1.1.5. Dashboard

The dashboard displays the measured farm data, the status of the irrigation motor, and the rainfall prediction for the next day.   Figure 3.5 shows a section of the dashboard screen.

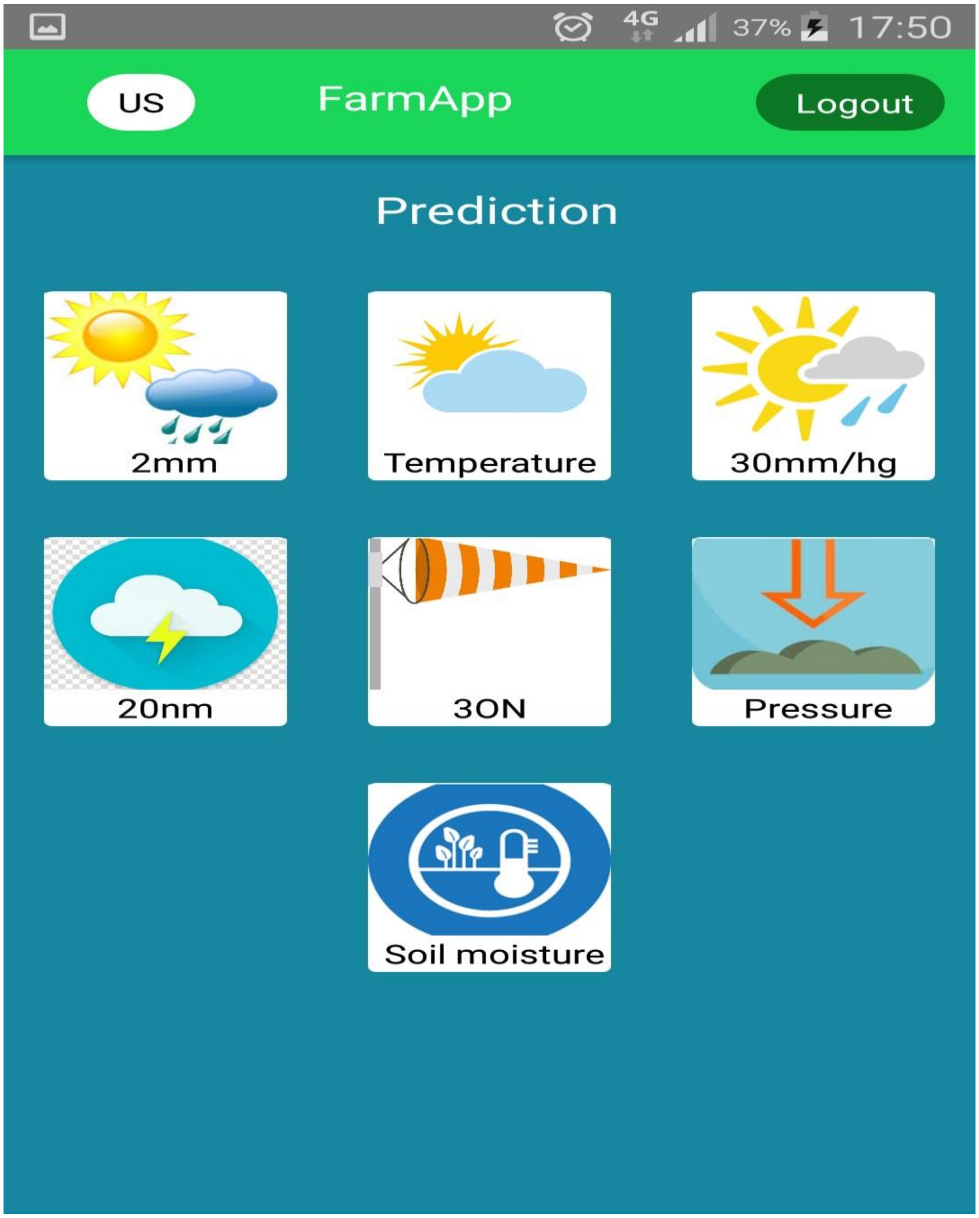


Figure 3:4 Section of the dashboard showing the farm measurements.

3.1.1.6. Activity diagram

The activity diagram shown in figure 3.6 shows the workflow of our application. When an authenticated user successfully logins in, the user is redirected to the dashboard screen which shows the measured farm data as well the status of irrigation motor. For an unsuccessful login, the user is redirected to the login view to try to login again.

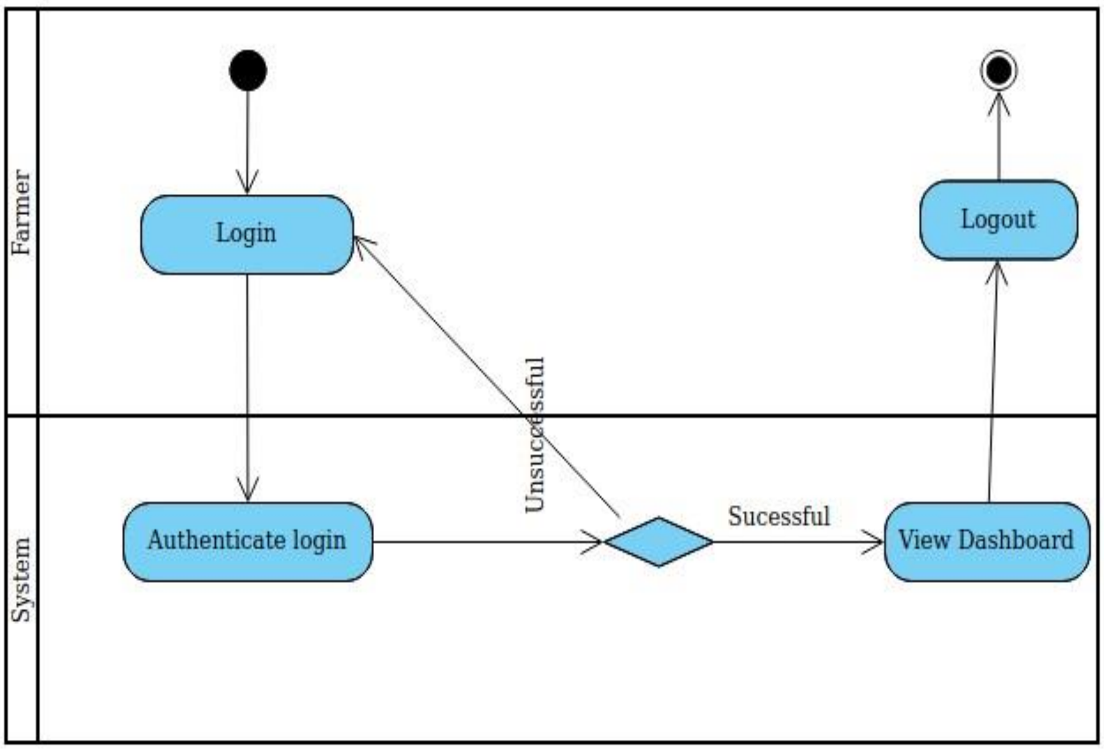


Figure 3:5 Activity diagram of our mobile application

3.1.1.7. Development tools

React Native

React Native is an open-source JavaScript framework used for creating cross-platform mobile applications. React Native is one of the popular, and widely used mobile application frameworks. It was developed by Facebook. There are a lot of advantages of using React Native for mobile application development. Some of them are; user interface (UI) focused, large community size, short development time, and good support for third-party libraries.

Firebase

Firebase is a backend service (Baas) provided Google for mobile and web application development [15]. Firebase has several features that make this service awesome. Some of the services provided by Firebase are; unlimited reporting, cloud messaging, authentication, hosting, and many more. It also provides real-time database service as a service that can be used to model a database for both mobile and web applications. The figure below summarizes the features of Firebase. Some of the benefits of using Firebase include; email and password, Google, Facebook, and GitHub authentication, real-time data, ready-made application development interface (API), and built-in security at the data node level. The features of Firebase is shown in figure 3.6 [16]

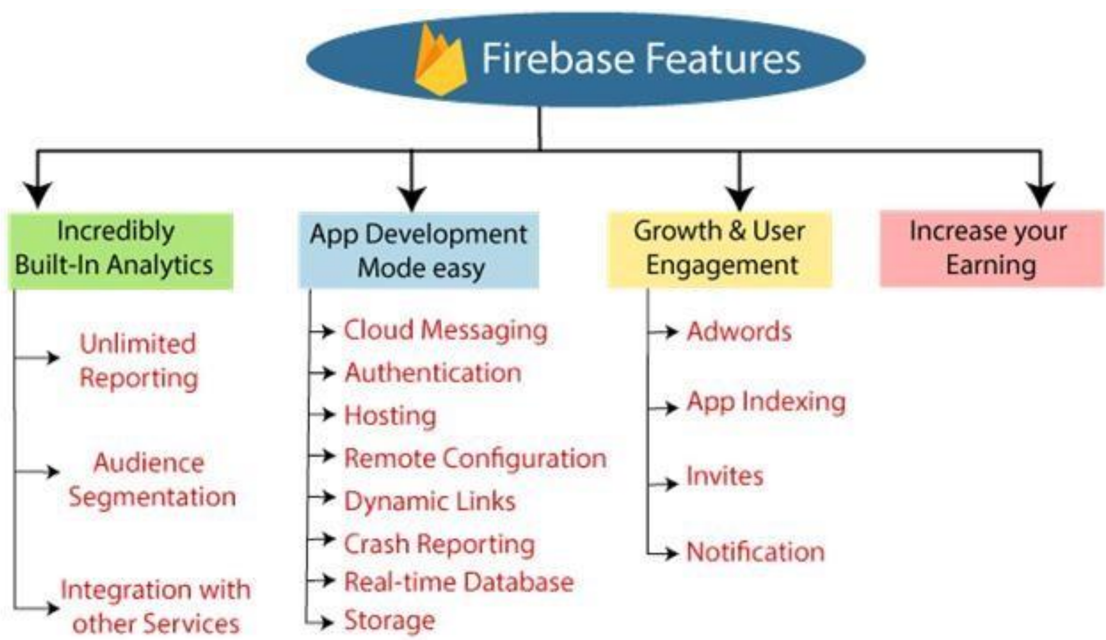


Figure 3:6 Firebase features diagram

3.1.2. Rainfall prediction model

This section is subdivided into three sections. Section 3.1.2.1 deals with the description of the data, and Section 3.1.2.2 presents the details of the model and hyper-parameters selection. Section 3.1.2.3 deals with the architecture of the deep learning model. Section 3.1.2.4 deals with the results and discussion.

3.1.2.1. Data Description and preprocessing.

The data that we used in this work was obtained from the Raspisaniye Pogodi Ltd website [17] which contains archives of global weather data. The weather data for Kumasi was available for a period from January 2010 to May 2020. The outcome of the dataset incorporated precipitation estimates obtained from rain gauge observations in weather stations in Kumasi. The dataset contained several parameters, but the relevant parameters for our prediction task were the atmospheric pressure at station level, the air temperature, the mean wind speed, the amount of precipitation, relative humidity, and the dew-point temperature. The choice of model parameters was informed by the sensor network we deployed on the farm. The objective was to train an LSTM rainfall prediction model to forecast the daily amount of rainfall in Kumasi. The prediction of the amount of rainfall daily is an important input into our irrigation system as such information helps us to decide the amount of water to pump into our farm. The data was prepossessed to remove irrelevant data fields and to fill null valued fields of the desired parameters with zeros. After preprocessing the data, we had a dataset with 13493 records. The histogram plots of some of the parameters in the preprocessed dataset are shown in the figures below. The data was then split into training, and test datasets. The training data had 10794 records and the test data 2699records.

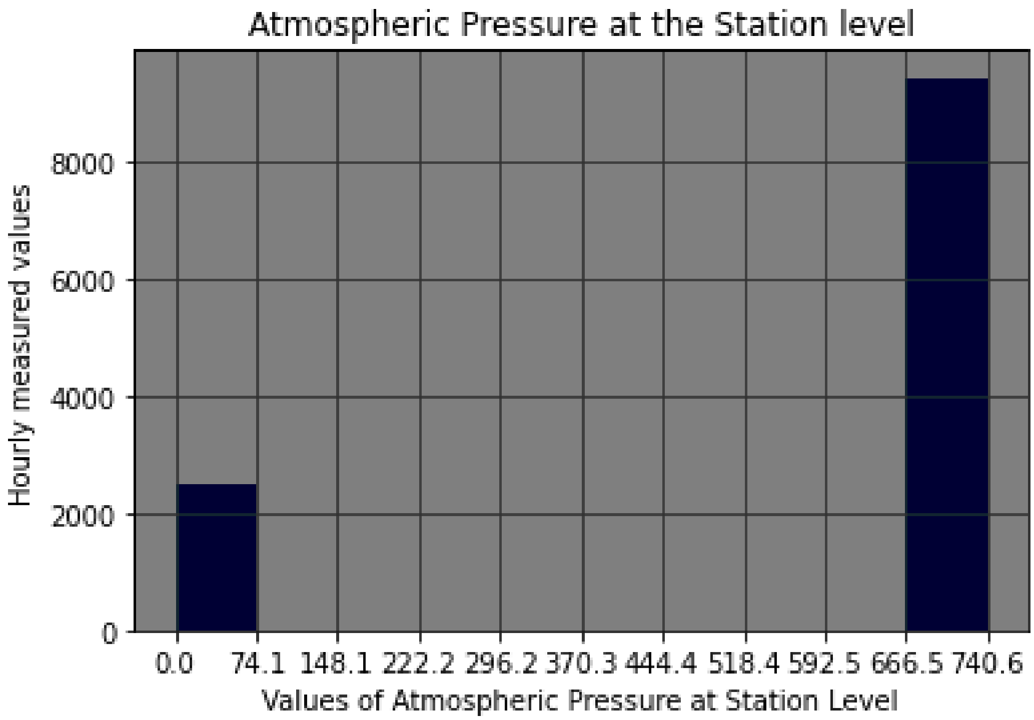


Figure 3:7 Histogram plot of the atmospheric pressure at station level from the preprocessed data

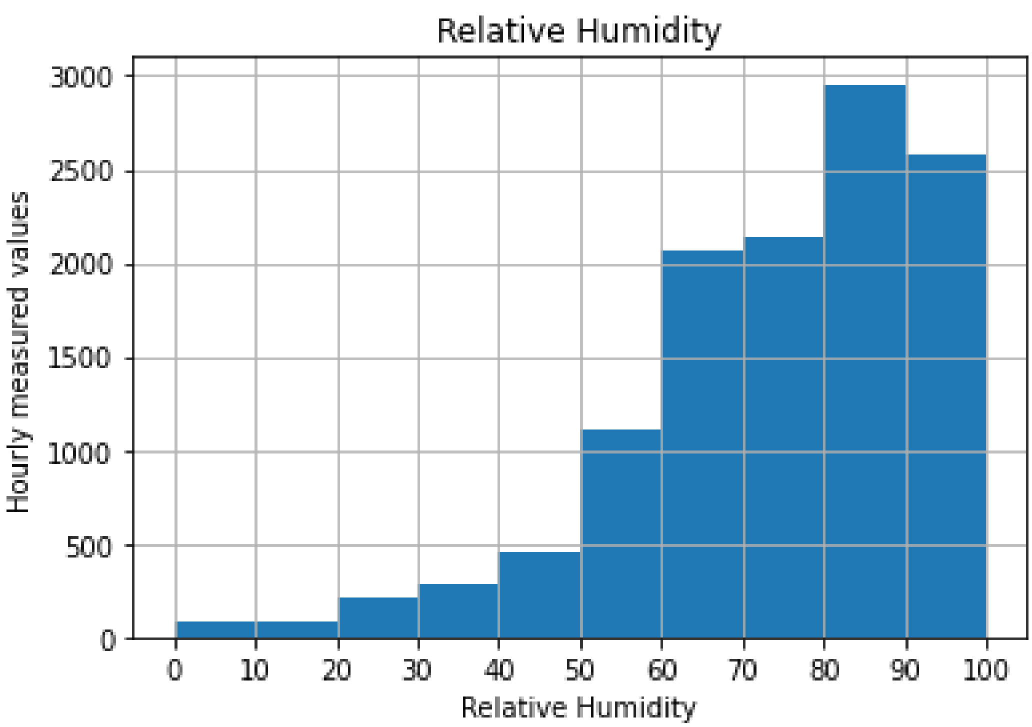


Figure 3:8 Histogram plot of the relative humidity from the preprocessed dataset

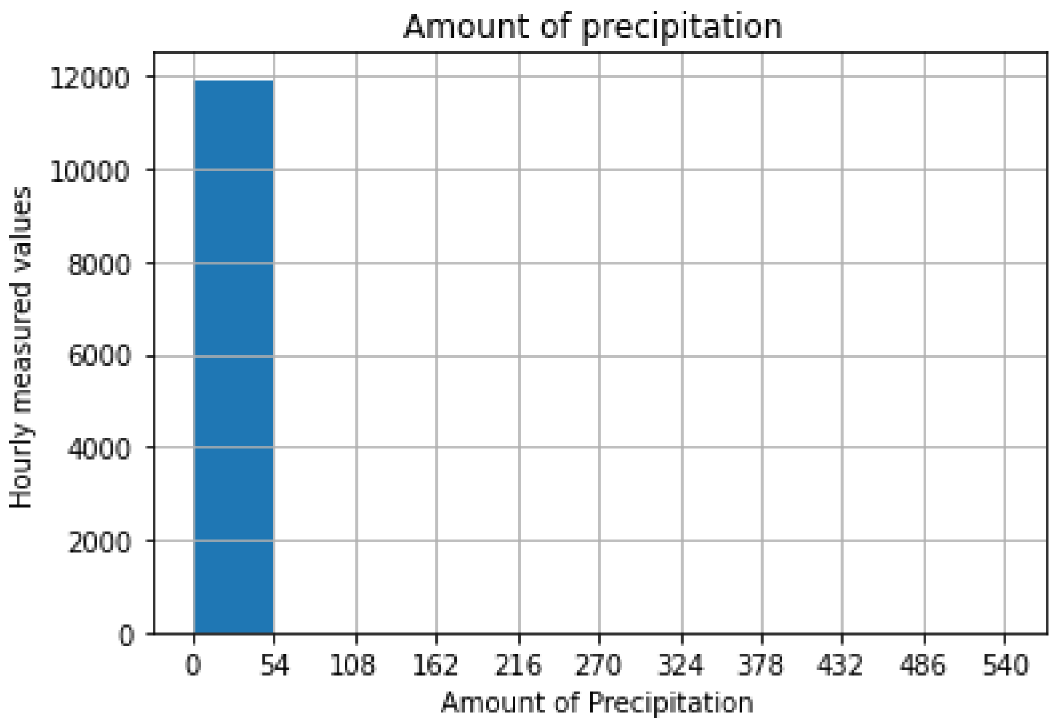


Figure 3:9 Histogram plot of the amount of precipitation from the preprocessed dataset

3.1.2.2. LSTM model training

From our literature review, we identified that using recurrent neural networks in rainfall prediction gave some promising results, therefore, we adopted the LSTM kind of recurrent neural networks to train our rainfall prediction model. An LSTM unit consists of three gates that control the input going through the model. In general, the data to an LSTM architectures is $x = (x_1, x_2, x_3, x_4 \ldots, x_n)$, which updates the output $y = (y_1, y_2, \ldots y_n)$ by updating the three cell state or gates .The forget gate is responsible for removing information from a cell state. The input gate adds new information to existing data. The LSTM input layer captures the rainfall data features across the time series data input. The loss function of our model is computed using the mathematical equation given below. The mean absolute error (MAE) $= \frac{1}{N} |y_{true} - y_{predicted}|$. The performance metrics of the model were the mean absolute error (MAE), the root mean squared error (RSME), and the accuracy. The activation functions we used for the model were the rectified Linear (ReLu) and sigmoid activations. Activation functions are mathematical relations that determine the output of a neural network. Finally, we applied batch normalization to normalize the activation effect [18].

The loss plot of our model is shown in **figure 3**.10 below.

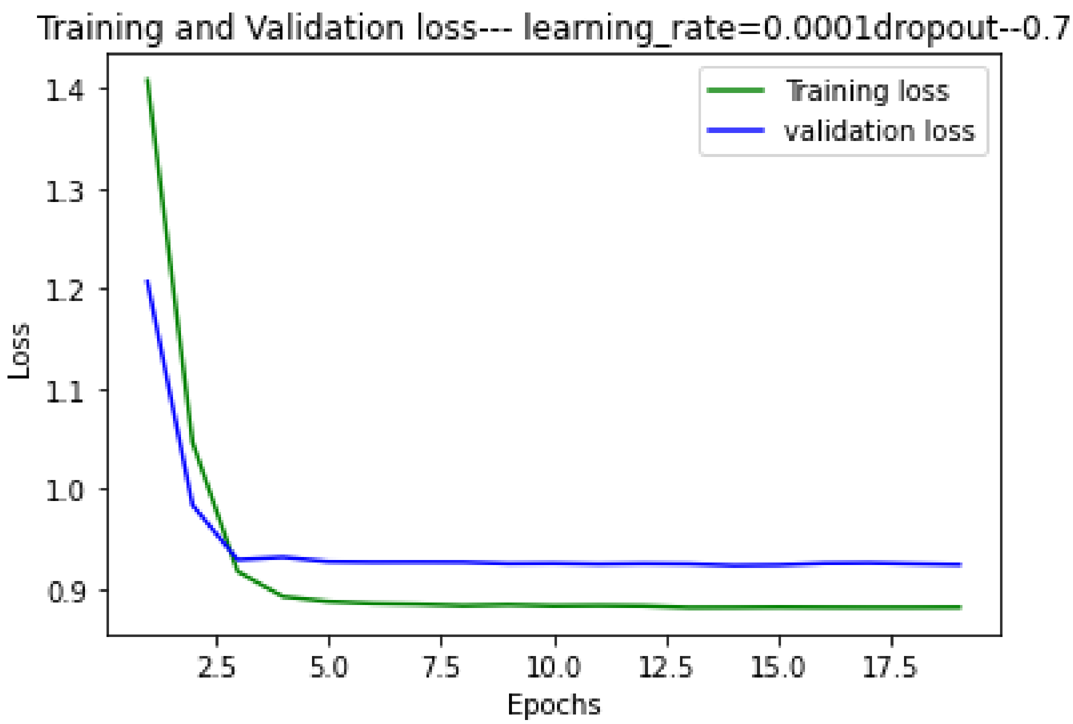


Figure 3:10 Training and validation loss plot for our deep learning model.

3.1.2.3. The Choice of hyper-parameters

Deep learning algorithms such as LSTM are parameterized and their performance depends on carefully selecting the right parameters. The hyper-parameters for the LSTM model are the learning rate, the number of training epochs, the batch size, and the number of neurons. Experiments were done to select the optimal values of these hyper-parameters to obtain a very low RSME score. To tune the hyperparameters, we used grid search and randomized search to search for the hyperparameters that yields optimum model performance. After several experiments with different possible values using the hyperparameter searching algorithms, a learning rate of 0.0001, training epochs of 35, and a batch size of 10 yielded the best model performance.

3.1.2.4. Model architecture

LSTM input data is supposed to be three-dimensional, and five feature vectors, and therefore we reshaped our training data into a three-dimensional dataset to pass it as input to our LSTM model. The model had four dense layers, two bidirectional layers, time distributed and batch normalization layers, and one flatten and dropout layers. Batch normalization is a technique used to normalize the output of the previous activation layer. Batch normalization solves the co-variant shift problem in deep neural networks and therefore improves the speed, performance, and stability of a neural network. ReLu and sigmoid were the activation functions we used in the dense layers. The input layer contained only one feature, which is the amount of precipitation. The optimization algorithm used in this model is Stochastic Gradient Descent (SDG). The detailed architecture diagram is shown below.

Model: "model_1"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 1, 5)	0
dense_1 (Dense)	(None, 1, 32)	192
bidirectional_1 (Bidirection	(None, 1, 256)	164864
time_distributed_1 (TimeDist	(None, 1, 128)	32896
batch_normalization_1 (Batch	(None, 1, 128)	512
bidirectional_2 (Bidirection	(None, 1, 512)	788480
time_distributed_2 (TimeDist	(None, 1, 128)	65664
batch_normalization_2 (Batch	(None, 1, 128)	512
flatten_1 (Flatten)	(None, 128)	0
dropout_1 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 1)	129

Total params: 1,053,249
Trainable params: 1,052,737
Non-trainable params: 512

Figure 3:11 Deep learning model architecture diagram

3.1.3. Hardware Design and implementation

This section discusses in detail the hardware implementation of our smart irrigation system. In this section, we shall discuss the sensor network design, the block diagram, and the flow chart of our irrigation algorithm. This section also highlights the hardware components used in the design of the hardware system.

3.1.3.1. Block diagram of our proposed smart irrigation.

The block diagram below shows the major sub-components of our system and the relationships that exist between them. The system block diagram is shown in the figure below.

The table below shows the various components we used in our hardware design.

Table 3:1: Table of hardware components

component	Function
Wifi Communication Module	Provides communication between sensors and analytics platform
Liquid Crystal Display	Displays values read from the sensor node and the battery power remaining for the system
Microcontroller	For the arithmetic and logical operations of the system.
Light Emitting Diode (LED)	An LED tells whether the pump is turned or not and the others are for debugging

Voltage Regulator	Regulates the voltage coming from the power supply to the required voltages required by every component in the system.
Power Jack	Serves as an adapter for the connection of an AC power source.
Resistor	For effective regulation of the voltage and to prevent all current from passing through some components.
Power Supply	Provides voltage for the entire system
Air Pressure Sensor (BMP280)	Measures the air pressure
Relay Module	This is used to switch the water pump on or off
Water pump	Pumps water through the system
Jumper Wires	This is used to connect all parts of the system
Temperature and Humidity sensor (DHT)	Measures the temperature and humidity of the weather

The figure below shows a diagram of the basic connections between the micro-controller and the highlighted components.

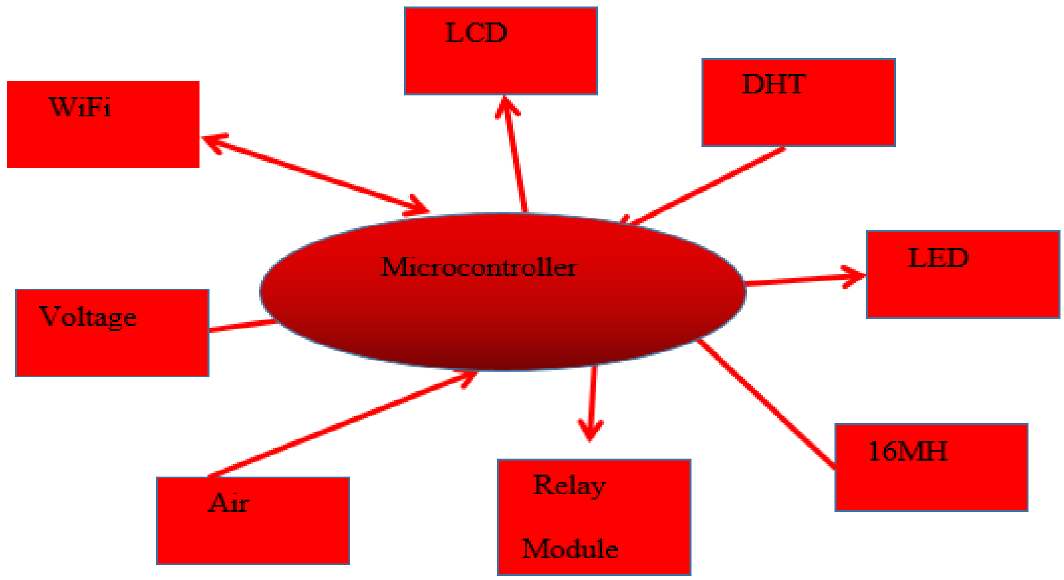


Figure 3:12 component diagram

3.1.3.2. System Modelling

To provide a well-presented system with its functions all spelled out, a flow chart is used. The figure below is used to explain the workflow of the system processes. The processes are divided into two groups: non-preemptive and preemptive processes. This division is based on how important a process is and whether a process can be interrupted at any time or even stopped. A non-preemptive process has high priority and has been executed at all times to realize the key functions of the system as such, these processes must not fail or be interrupted. When there is low power, for the system to be able to realize functions, lower priority processes are halted for these processes to run. These processes only stop when they need a resource or the output of another process. They then wait for the process to finish and then it takes the resource and continues.

A preemptive process can always be interrupted and even stopped at any time of their execution because they do not directly contribute to the key functions of the system. Because of this, the process can be stopped at any time.

3.1.3.3. Flowchart diagram

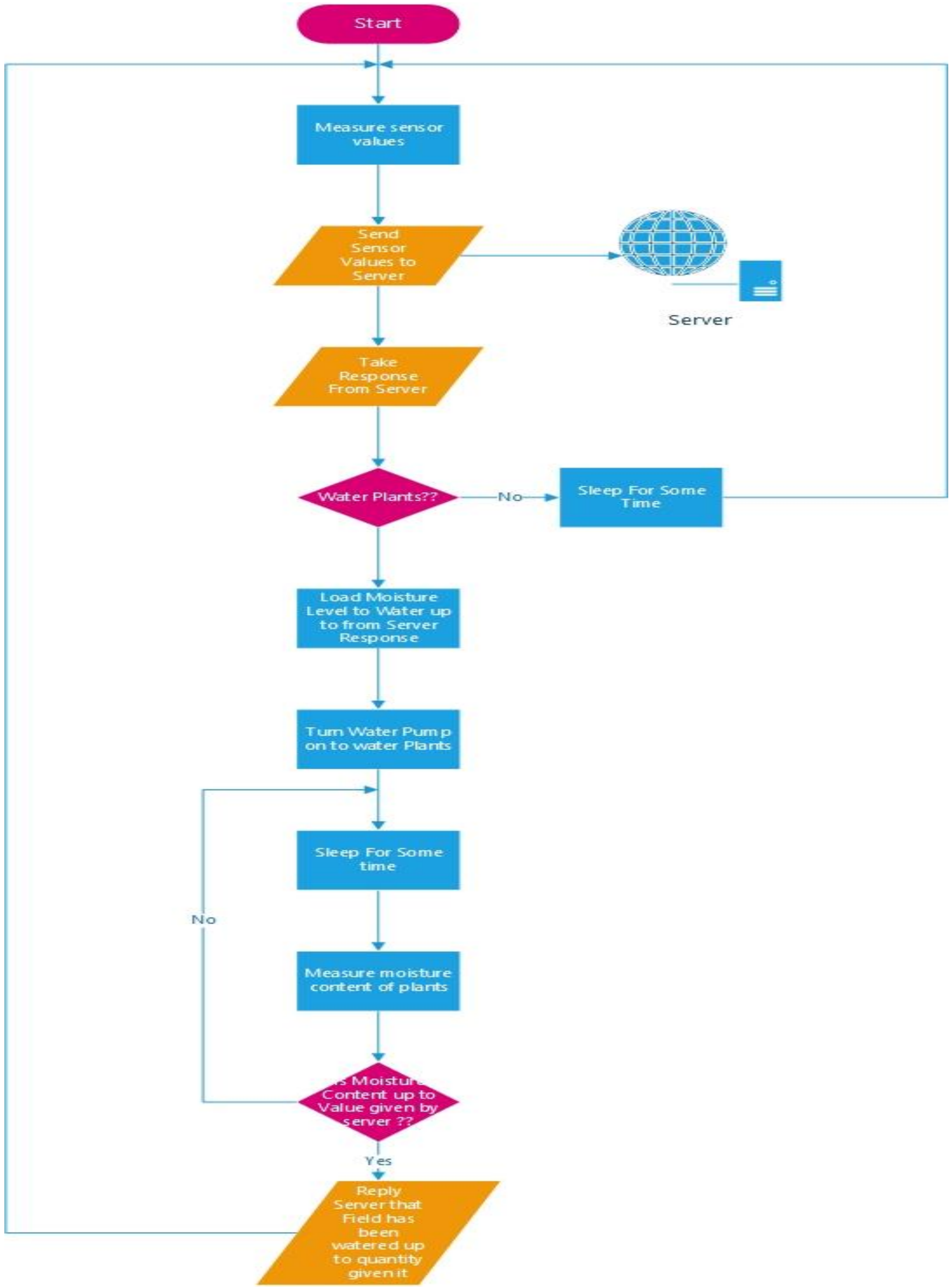


Figure 3:13 Flow chart diagram of irrigation algorithm

The flow chart above shows the algorithm for which the system works. The micro-controller measures the values of the sensors. It takes the air pressure readings from the air pressure sensor (BMP280), the temperature and humidity readings from the Temperature and humidity sensor (DHT). These values are checked to ensure that there are no errors in them, and then they are sent to the server over the Wi-Fi connection in the system. The micro-controller then takes in input from the server and determines whether to water the system or not. In the server, the deep learning model predicts the amount of precipitation and so the amount of precipitation is compared with the quantity of water the soil needs. If the amount of rainfall cannot accommodate for the soil water requirement at that time, the microcontroller performs some computation to determine the quantity of water it has to pump into the farm. However, if the amount of precipitation can meet the soil water requirements at that time, the microcontroller goes to sleep some interval before the next reading is to be taken, otherwise, it goes ahead to pump water into the farm, thus turning on the motor. When it is done watering the system to the threshold, it replies to the server that the system is watered well enough. Also, if the soil moisture content is not below a certain threshold, the preliminary checks about the amount of precipitation from the deep learning model would be ignored and the microcontroller goes to sleep.

3.1.3.4. Component Selection

In this section, the various components which were used in the design and implementation of the smart irrigation system will be explained in detail and the very reason for their selection will be justified. These components have been carefully selected based on performance, availability, mode of operation, robustness, and system requirement. Diagrams of elements in this section are not drawn to be scale, they are just arbitrarily enlarged and placed to enhance legibility.

NodeMCU

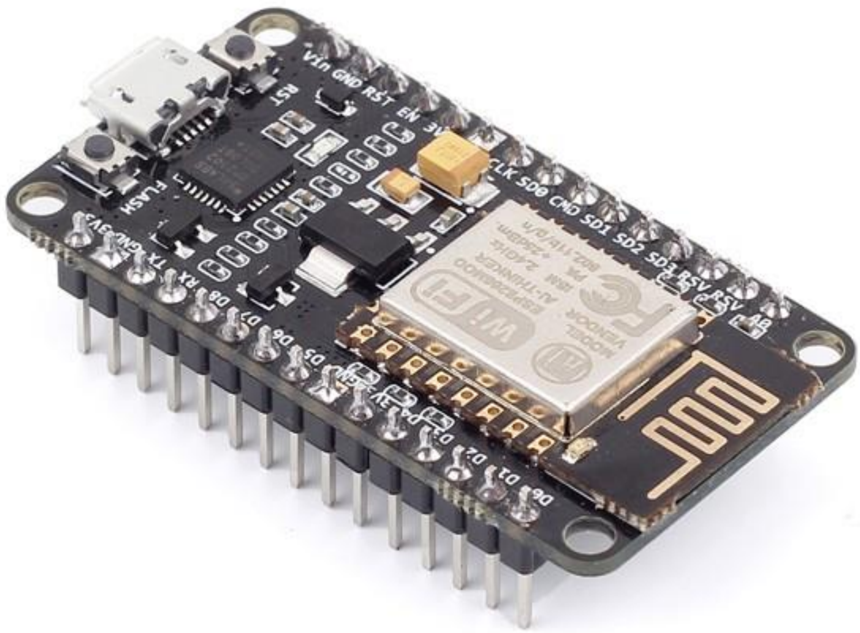


Figure 3:14 NodeMCU

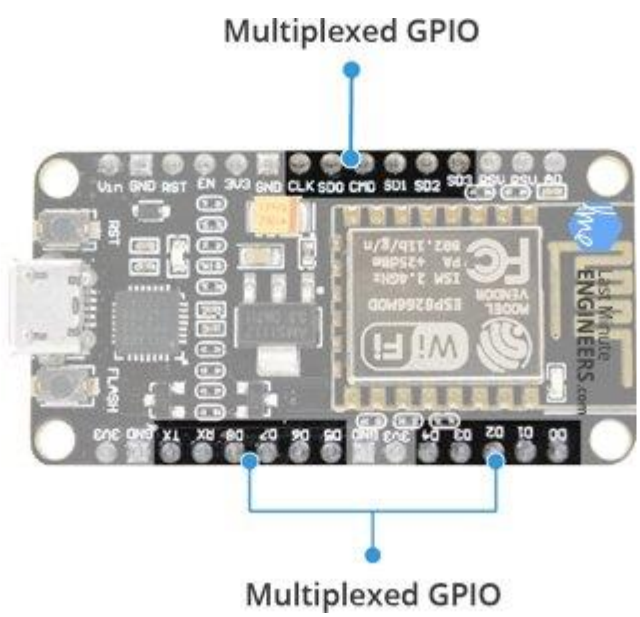
NodeMCU is a low-cost open-source IoT platform. It is used for prototyping. It has an onboard Wi-Fi module that can be used to connect to a Wi-Fi network. The NodeMCU board has an ESP-12E module that has an ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor which operates at 80 to 160 MHz adjustable clock frequency. It also has 128 KB RAM and 4MB of Flash memory (for program and data storage) which is enough to hold web pages, JSON/XML data, and every code IoT devices are made to function on. This allows it to be able to create and host its website. The ESP8266 has an onboard 802.11b/g/n HT40 Wi-Fi transceiver**,** so it can connect to a Wi-Fi network and interact with the Internet, and also set up a network of its own, allowing other devices to connect directly to it. This makes the ESP8266 NodeMCU even more versatile. The operating voltage range of ESP8266 is **3V to 3.6V**, the board comes with an LDO voltage regulator to keep the voltage steady at 3.3V. It can supply up to 600mA, which leaves enough current after it pulls as much as 80mA during RF transmissions**.** Power can be supplied to the NodeMCU over the onboard Micro USB connector or through the VIN pin.

The ESP8266 NodeMCU has a total of 17 GPIO pins broken out to the pin headers on both sides of the development board. It has pins for:

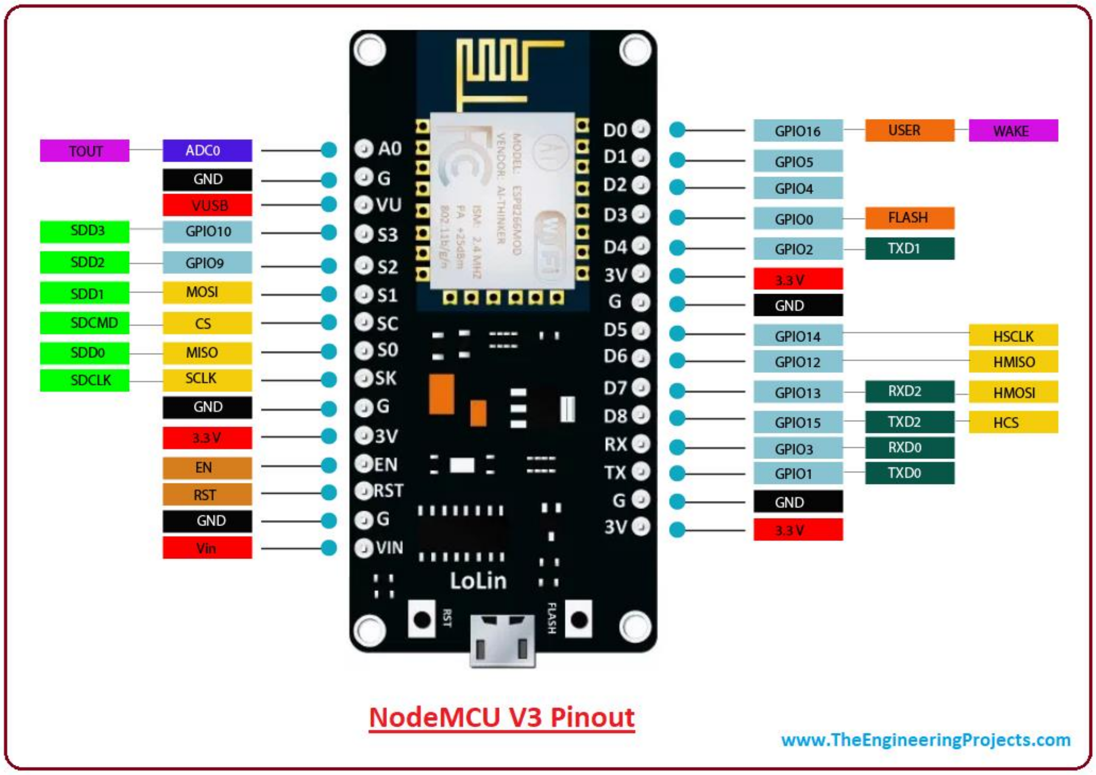
- **ADC channel** – A 10-bit ADC channel.
- **UART interface** – UART interface is used to load code serially.
- **PWM outputs** – PWM pins for dimming LEDs or controlling motors.
- **SPI, I2C & I2S interface** – SPI and I2C interface to hook up all sorts of sensors and peripherals.
- **I2S interface** – I2S interface if you want to add sound to your project.

Multiplexed I/Os

- 1 ADC channels
- 2 UART interfaces
- 4 PWM outputs
- SPI, I2C & I2S interface



Thanks to the ESP8266's pin multiplexing feature (Multiple peripherals multiplexed on a single GPIO pin). Meaning a single GPIO pin can act as PWM/UART/SPI.



NodeMCU V3 Pinout

www.TheEngineeringProjects.com

LED

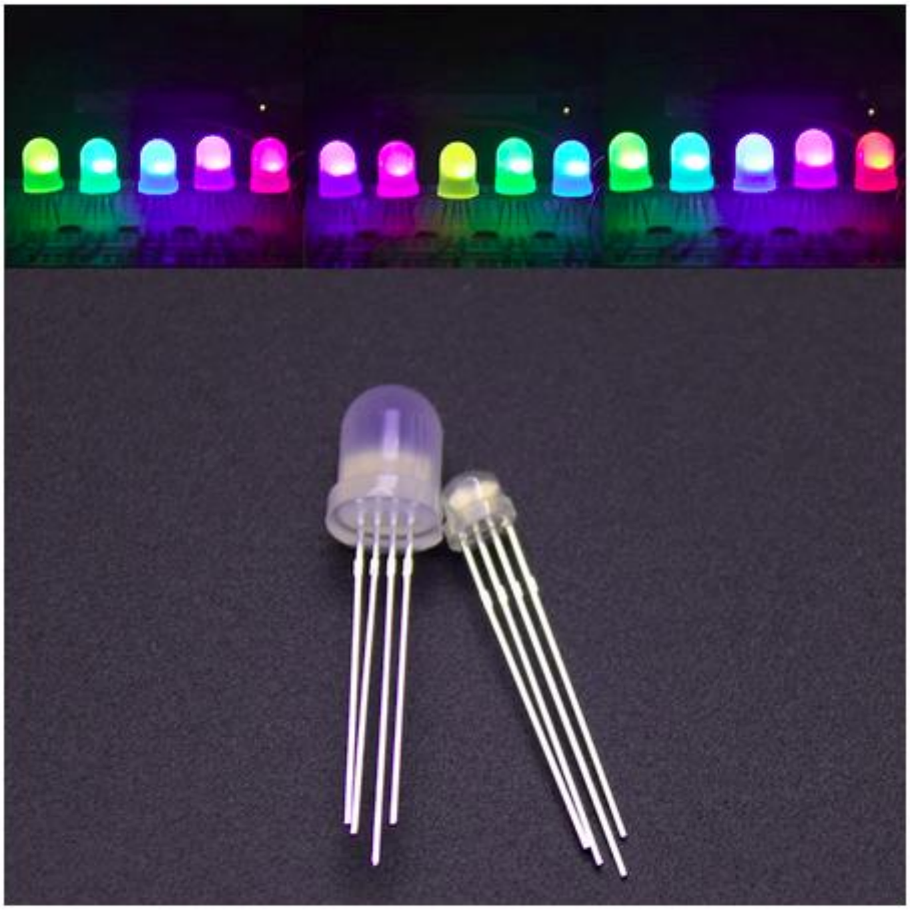


Figure 3:15 LED

An LED was used to show whether or not the pump is running. The others were used for debugging. RGB LEDs were used since these could change color. With these, we did not need to use a lot of LEDs

LCD

As stated in the features and functional requirements, the data computed and received from the sensors need to be displayed. This led to the use of a display. Although there are some displays in the market, we settled on the 16 x 2 Liquid

Crystal Display because it is cheap, and utilize power effectively and efficiently.

Figure 3:16 LCD

It has the following features:

1. LED Backlight to illuminate the screen.
2. Positive 5 volts power supply.
3. Expected lifetime of ~30,000 hours.
4. Adjustable contrast.
5. Operational Temperature range of 0 to +50 ^{o}C

Also, this module is preferred over other multi-segment LCDs because LCDs are economical, easily programmable, have no limitation of displaying special and even custom characters. The LCD in this state is very complex to program, to ease the workload and the computational requirements from the Microcontroller unit, the system included an Inter-Integrated-Circuit (I2C).

Power Supply

The micro-controller and sensors run on DC and so was supplied with DC. The power was supplied from a battery that was connected to the NodeMCU VIN pin and from there all other sensors were powered. The water pump runs on DC and so it was supplied with DC, but since it required a lot more current and voltage that the micro-controller could supply, the power taken from the mains and converted using a normal phone charger and was then passed through a relay to the pump so the micro-controller could control the pump using the relay.

Air Pressure Sensor (BMP280)

To sense the air pressure, the BMP280 air pressure sensor was used. The BMP280 is a barometric pressure sensor. Its small dimension and low power consumption make it suitable for mobile applications and battery-powered devices. It has high accuracy and linearity as well as long-term stability and high EMC robustness. It is optimized in terms of power consumption, resolution, and filter performance. The BMP280 operates based on the I2C protocol. Below is an image of the BMP280 pinouts.

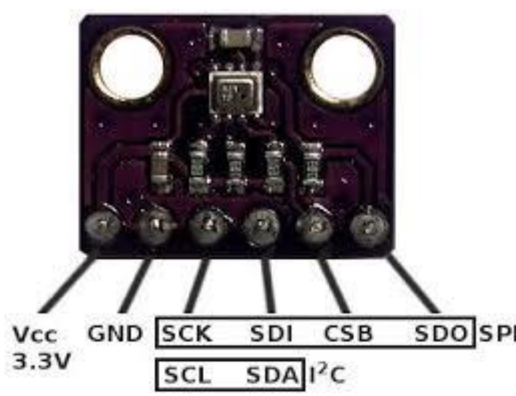


Figure 3:17 Power supply

Table 3:2 : Air Pressure Sensor pins

Description	Function
VCC	It serves as the reference voltage for the air pressure sensor
GND	It acts as a zero voltage reference
CSB	CSB pin to GND to have SPI and to VCC (3.3V) for I2C. It’s an input to the chip.

SDO	**S**erial **D**ata **O**ut / **M**aster **I**n **S**lave **O**ut pin, for data sent from the BMP280 to your processor

For this project, pins 1 - 4 were used.

Temperature and Humidity Sensor (DHT22)

Figure 3:18 Temperature and Humidity sensor

The DHT sensors made of two parts, a capacitive humidity sensor, and a thermistor. The capacitive humidity sensor measures the humidity and the thermistor measures the temperature. There is also an 8-bit microcontroller inside the sensor that does analog to digital conversion and giving out a digital signal through pin 2 as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers. Below are the sensor's specifications:

1. Low cost
2. 2. 3 to 5V power and I/O
3. 2.5mA max current use during conversion (while requesting data)

4. Good for 0-100% humidity readings with 2-5% accuracy
5. Good for -40 to 80°C temperature readings ±0.5°C accuracy
6. No more than 0.5 Hz sampling rate (once every 2 seconds
7. Body size 15.1mm x 25mm x 7.7mm
8. 4 pins with 0.1" spacing

The Sensor has 4 pins:

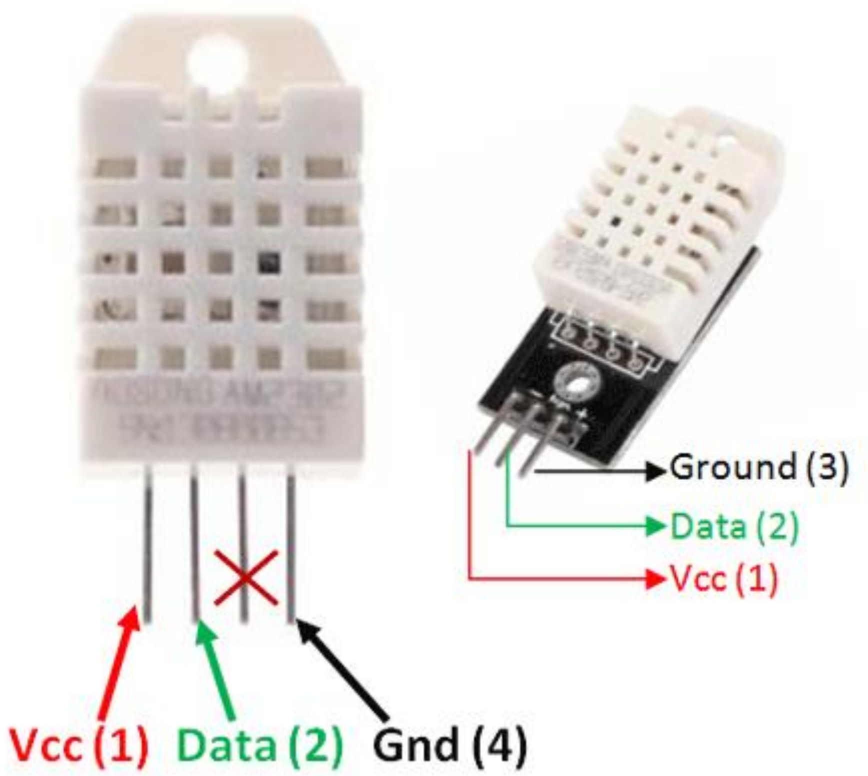


Figure 3:19 Temperature and Humidity sensor pins

Pin 1 connects to 5V, and pin 4 connects to the ground. Pin 2 is the data pin. Pin 3 is not used. Relay

Figure 3:20 Relay

The relay is used to connect the water pump to the micro-controller and the water pump to its power source. The relay is triggered by a 5V or 0V signal from the micro-controller. 5V opens the relay and 0V closes it. Below are the pinouts of the relay module:

Table 3:3: Temperature and Humidity pins

Description	Function
GND	Connects to ground
VCC	Connects to 5V which serves as the reference voltage
Signal	Input from sensor
NC	Normally closed. 5V would make this pin open
C	Common pin. Ground from device to be controlled and the power source is connected here
NO	Normally open. 5V would make this pin close

Figure 3:21Relay Terminals and pins

Pins 1-3 are connected to the micro-controller and pins 4-6 are connected to the power source and pump. Two types of connections are presented here, normally closed and normally open. Normally closed is used when you expect the pump to be closed under normal circumstances. A signal from the micro-controller would make the pump on. This is used if the pump would be on most at times. Normally open is used if the pump would be opened under normal circumstances. This is not what we needed in the project so we used the normally closed.

Water Pump

Figure 3:22 Water pump

The pump was used to pump water to the farm (represented by a small container filled with water). The pump is a 12V water pump, ideal for our project. Since the micro-controller could not provide enough power for it, the power had to be drawn from the mains through a regulator (mobile phone charger).

Chapter 4: SIMULATIONS, RESULTS, AND DISCUSSION

4.1. Rainfall Prediction Model

4.1.1. Rainfall Prediction

After several experiments with our model, the test RMSE score, accuracy, and mean absolute error of 7. 834, 0.9293, and 0.925 were obtained respectively. The plot of the model's predictions is shown below. From the model prediction's plot, we can see that the model performs poorly. The poor performance is attributed to the dataset as the dataset contained a lot of null value fields which were replaced by zeroes.

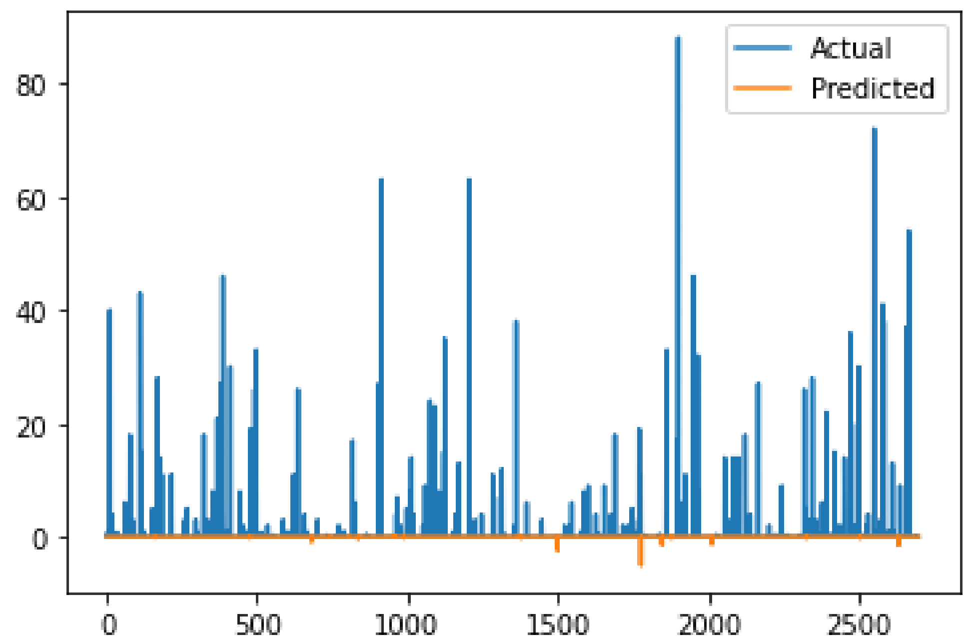


Figure 4:1 Sample results of our deep learning model

4.1.2. Mobile Application

The mobile application works fine and it is easy to use. A farmer can used the mobile application with little training. The mobile application has a short response time.

Chapter 5: CONCLUSION AND RECOMMENDATION

5.1. Conclusion

This project focused on the design and implementation of a smart irrigation system to overcome the challenges of the existing irrigation systems used in Ghana. The proposed solution can overcome the problems the existing irrigation systems face. With a focus on conversing water, our system has the potential to converse water while making sure the soil water requirement of the farm is met.

5.2. Recommendations

The proposed solution has the potential to mitigate the challenges existing in the current irrigation systems. The proposed system provides foresight for research into the conversation of water in smart irrigation systems. We recommend that more research should be done in this area to help the agriculture industry converse water as water is becoming a scarce resource.

5.3. Challenges

The main challenges we encountered in the course of the project were;

- The dataset we had to train our deep learning model had a lot of null-valued fields which does not give a clear and real-life reflection of the daily weather for Kumasi.
- The cost of some of the sensors we needed for the project was so expensive and as a result, we were not able to purchase some of the sensors which greatly impacted the outcome of our deep learning model.
- Due to the Covid-19 pandemic, we were not able to fully implement the hardware aspect of our project.
- We also had challenges acquiring weather data from the Kumasi unit of the Ghana Metrological Agency and we had to resort to the global archives of weather data from our Raspisaniye Pogodi Ltd website.

References

[1] "ScienceDaily," , [Online]. Available: https://www.sciencedaily.com/terms/water_scarcity.htm#:~:text=2-,Water%20scarcity,access%20to%20clean%20drinking%20water.. [Accessed 20 05 2020].

[2] Farmerline, [Online]. Available: https://farmerline.co/2019/05/29/securing-the-future-of-agriculture-in-the-face-of-an-ageing-farmer-population/. [Accessed 20 03 2020].

[3] Jakari, A. H. M, M. M. Hossain, Rahman and M. Ashiqur, "Smart Weather Forecasting Using Machine Learning : A Case Study in Tennessee," in *Proceedings of ACM Mid-Southeast conference*, New York, NY, USA, 2018.

[4] Chandrasegar Thirumala; M Lakshmi Deepak;K Sri Harsha;K Chaitanya Krishna, "Heuristic Prediction of Rainfall Using Machine Learning Techniques," in *International Conference on Trends in Electronics and Informatics*, 2017.

[5] Salman, A. Galih, Kanigoro, Bayu, Heryad and Y. , "Weather Forecasting using Deep Learning," in *International Conference on Advanced Computer Science and Information Systems*, 2015.

[6] A. S. S. T., R. Somula, G. K. and A. Saxena, "Estimating rainfall using machine learning strategies based," *International Journal of Communication Systems*, 2019.

[7] A. S, G. P and V. R, "Deep Learning Models for the Prediction of Rainfall," in *International Conference on Communication and Signal Processing*, India, 2018.

[8] X. Shi, Z. Chen, H. Wang and D.-Y. Yeung, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting".

[9] Meng-HuaYen, D.-W. Liu, Y.-C. Hsin, C.-E. Lin and Chii-ChangChen, "Application of the deep learning for the prediction of rainfall in Southern Taiwan," Scientific Reports, 2019.

[10] A. B. V, "A Study on Smart Irrigation System Using IoT for Surveillance of Crop-Field," *International Journal of Engineering & Technology*, pp. 371-373, 2018.

[11] S. B. Saraf and D. H. Gawali, "IoT Based Smart Irrigation Monitoring And Controlling System," in *2nd IEEE International Conference On Recent Trends in Electronics Information & Communication Technology*, India, 2017.

[12] A. Premkumar, P. Monishaa, T. K, R. Amirtharajan and P. Praveenkumar, "IoT Assisted Automatic Irrigation System using Wireless Sensor Nodes," in *International Conference on Computer Communication and Informatics*, Coimbatore, INDIA, 2018.

[13] W. Zhao, S. Lin, J. Han, R. Xu and L. Hou, "Design and Implementation of Smart Irrigation System Based on LoRa," in *2017 IEEE Globecom Workshops (GC Wkshps*, 2017.

[14] C. Shang, W.-H. Chen, A. D. Stroock and F. You, "Robust Model Predictive Control of Irrigation Systems With Active Uncertainty Learning and Data Analytics," in *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, 2019.

[15] "Firebase Documention," Firebase, [Online]. Available: https://firebase.google.com/docs. [Accessed 21 May 2020].

[16] "Features of Firebase," Javapoint, [Online]. Available: https://www.javatpoint.com/features-of-firebase. [Accessed 21 05 2020].

[17] " Raspisaniye Pogodi Ltd," rp5.ru, [Online]. Available: https://rp5.ru/docs/about/en. [Accessed 23 03 2020].

[18] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv*, 2015.

APPENDICES