Github Link Here

---

masaFarms : AIoT Smart Agriculture Solution Using LoRaWAN Architecture

---

Students:
Haileleul Haile
Kidus Birkayehu

September 09, 2021

Abstract

For our final project we have chosen to work on a data driven agriculture project using machine learning. One of the factors that limits agricultural production is the scarcity of agricultural data which can help farmers make informed-decisions. This problem can be addressed by deploying a system which can supply data in real-time. Our project proposed to deliver such a system by utilizing the relatively new LoRaWAN technology. This will enable the usage of battery powered sensors for a longer time. Furthermore, because of the larger distance coverage of the network system, internet connection is required at a single central point. This is an important aspect of the system given the limited internet coverage in rural farming areas. These factors make the project feasible given the country's condition. End-user of this system is Ethiopia's Agricultural Transformation Agency. This agency has a good practice of delivering information to farmers in ways the farmers find convenient. ATA has already deployed a call-line system called 8028. The information gathered from this project may be integrated to 8028. to be finally delivered to farmers.

For ease of implementation, we have divided our tasks in two parts. In the first phase we have planned to lay out the necessary platforms that are needed for input data to be collected, communicated by and processed in the second phase. These tasks mainly include interfacing the sensors with the LoRa end devices, building and setting up the database (on a local machine for now) and building a suitable machine learning algorithm that predicts soil water moisture from temperature, humidity and pressure values. This documentation will give a detailed description about the literature review done, and design implemented and the results obtained regarding these tasks.

Table of Contents

List of Tables/Figures

Introduction

The world is big and it keeps getting bigger. It is estimated that the global population is going to reach 8.6 billion by mid-2030, 9.8 billion by mid-2050 and 11.2 billion by 2100[1]. In Ethiopia, the population growth is more drastic. While the world is showing a 1.1% annual change in population, Ethiopia shows 2.6% annual change [2]. Even though the relationship between agricultural production and population growth is complex, it is quite evident that where there is a large number of populations there is a high demand for food. This, together with Ethiopia's abundant agricultural resources, makes it one of the various reasons that Ethiopia's labor force is predominantly involved in agriculture. Ethiopia is one of the least developed countries in Sub-Saharan Africa with more than 100 million people, of which 80.5% of the rural population relies on agriculture for their livelihoods.[1][2] In Ethiopia, one-third of rural households cannot produce adequate food to meet their subsistence needs as they cultivate less than half hectares of land per capita. Whatever size of farm the farmers can cultivate, it is a necessity that they cultivate it efficiently.

What also seems to be happening in every agricultural revolution that happened in this world was the adoption of better and better technology. From the use of oxen to plough a field, to fabricating genetically improved fertilizers and pesticides and spraying them with planes, humans have been up to date with whatever is available to them to make the most out of the lands they have. While this is the case for agricultural and economic superpowers like the USA and China, poor countries like Ethiopia are still farming using the technology they have been using thousands of years ago. With little or no technological advancement in the agricultural sector, we have reached the age of information; an age where information is key in every economic sector. If we can get accurate, computer-aided information to circulate within the agricultural sector in Ethiopia, we can jump several steppingstones to modernize our agriculture and methodically use our resources.

For our final project we proposed providing the farmer with accurate soil moisture data by measuring pressure, humidity and temperature values that can be fed to a machine learning algorithm. These values are collected using their respective sensors that are connected through LoRaWAN technology. The LoRaWAN specification is a Low Power, Wide Area (LPWA) networking protocol designed to wirelessly connect battery operated 'things' to the internet in regional, national or global networks, and targets key Internet of Things (IoT) requirements such as bi-directional communication, end-to-end security, mobility and localization services.[6] The data collected by the sensors will be used as a feature dataset to predict the soil moisture levels of the entire farming field.

From a larger viewpoint, to make the soil moisture information readily available to the farmers, we chose the Agricultural Transformation Agency Office (ATA) as the end user of our technology. ATA is a strategy and delivery-oriented government agency created to help accelerate the growth and transformation of Ethiopia's agriculture sector. They collect and analyze agricultural data to provide farmers with information. In their previous project, EthioSIS, they have manually collected and tested soil samples for soil nutrients from the four biggest regions in Ethiopia, Tigray, Amhara, Oromia and SNNPR, to create a soil nutrient map throughout these regions in order to give the right soil fertilizer recommendations. They have deployed an automated phone call system where farmers can simply dial 8082 on their phone and choose the right sequence of characters to get the information, they need to make a decision. The output of our technology is planned to be delivered to go to ATA and be integrated into the 8082 system. Farmers will use the 8082 automated phone call system to ask about the soil moisture data of their field and ATA will present to them with what they need.

Background

In this project we aim to predict soil moisture values from pressure, humidity and temperature values that we collect using a LoRaWAN networking scheme. To collect soil moisture values that are used to train our machine learning algorithms we have chosen to use a soil moisture sensor. The fork-shaped probe with two exposed conductors, acts as a variable resistor (just like a potentiometer) whose resistance varies according to the water content in the soil. This resistance is inversely proportional to the soil moisture. The more water in the soil means better conductivity and will result in a lower resistance. The less water in the soil means poor conductivity and will result in a higher resistance. The sensor produces an output voltage according to the resistance, which by measuring we can determine the moisture level.

We chose pressure, temperature and humidity as feature variables for our algorithm. Temperature, humidity, and pressure have been found to give best approximations against soil moisture. Using the BME/BMP280 chip, which has all three kinds of sensors on it, we were able to collect all three-feature data at once. Not only that this way is more convenient to use, but it also gave less variations on all three types of data as they got fed to the algorithm for training.

For reasons described above, we chose to communicate all our sensor data on a LoRaWAN networking scheme. LoRaWAN network architecture is deployed in a star-of-stars topology in which gateways relay messages between end-devices and a central network server[6]. The gateways are connected to the network server via standard IP connections and act as a transparent bridge, simply converting RF packets to IP packets and vice versa. The wireless communication takes advantage of the Long-Range characteristics of the LoRa physical layer, allowing a single-hop link between the end-device and one or many gateways. All modes are capable of bi-directional communication, and there is support for multicast addressing groups to make efficient use of spectrum during tasks such as Firmware Over-The-Air (FOTA) upgrades or other mass distribution messages.[6]

To be able to send our sensor collected data in the form of LoRa packets we have chosen to mount a Dragino shield on top of an Arduino Uno. The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts.[15] The Dragino shield, technically known as SX127x Shield, is an Arduino Shield featuring LoRa® technology and based on Open-source library. This Shield allows the user to send data and reach extremely long ranges at low data-rates. It provides ultra-long range spread spectrum communication and high interference immunity whilst minimizing current consumption. The dragino shield is based on Semtech SX1276/SX1278 chip, it targets professional wireless sensor network applications such as irrigation systems, smart metering, smart cities, smartphone detection, building automation, and so on.[9]

The data that is collected through this LoRa scheme, it will be fed to a machine learning algorithm. Based on a thorough literature review we made in preparation for this project, multivariate linear regression has proven to present accurate results for the type of work we are doing. Multivariate Regression is a method used to measure the degree at which more than one independent variable (predictors) and more than one dependent variable (responses), are linearly related. The method is broadly used to predict the behavior of the response variables associated with changes in the predictor variables once a desired degree of relation has been established.[8][3]

In addition to the multivariate linear regression algorithm, we were able to synthesize, we have used a random forest library to help us compare results based on accuracy to the real value. A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems.

A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms. The random forest algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.

The data is presented to ATA on a grafana panel. Grafana is open-source visualization and analytics software. It allows you to query, visualize, alert on, and explore your metrics no matter where they are stored.[16] In plain English, it provides you with tools to turn your time-series database (TSDB) data into beautiful graphs and visualizations. This way ATA can have an easier time integrating the data we put in our database to their 8082-hotline information system.

Objective

In this project, we identify the lack of reliable soil moisture information as the general problem we will tackle. We started by researching how important soil moisture balance is in Ethiopia's agricultural sector and previous work which has been done to solve the problem. We decided to tackle the problem from a technological perspective and asked, "How can we present computer aided soil moisture data on farms to help farmers make better decisions in order to maximize their agricultural feat in the most efficient way?" By answering this question, we came with a plan to provide a government agency called ATA with a system that provides accurate soil moisture data over a given farm field for them to analyze and outsource extracted data to farmers. To make this happen, we have planned to collect soil moisture data sparsely using soil moisture, temperature, humidity and pressure sensors. The data we collected are transmitted to a gateway using LoRa technology which requires internet connection only at a single point, works in long range and under very low power restrictions. From this gateway, the data is sent to be stored and processed to a server (a process which requires internet connection). The data that is sent is used as an input to a multivariate regression machine learning algorithm that predicts soil moisture data just from the collected feature variables.

Literature Review

On the Problem:

A Review on Agricultural Problems and Their Management in Ethiopia

Waterlogging is the main drainage problem in the small-scale irrigation schemes in the Vertisols dominated highland areas. Ethiopia ranks third in Vertisols abundance in Africa after Sudan and Chad (Kebede and Bekelle, 2008). Vertisols cover a total of 12.6 million ha (10.3 percent) of the soils in Ethiopia of which more than 60 percent is in the highlands where traditional smallholder mixed farming is practiced (Jutzi, 1990). In irrigated agriculture, efficient water management is an important element.

Waterlogging affects plant growth by reducing soil aeration around the root zone (Singh, 2015). Due to the occurrence of seasonal waterlogging during the main rainy season, early planting is prohibited with traditional management systems in the north central highlands which in turn reduce the length of the growing cycle (Erkossa et al., 2004; Kebede and Bekelle, 2008). Similarly, the Wonji-Shoa Sugar Estate, one of the large-scale irrigation systems within the Upper Awash River Basin of Ethiopia with nearly half of its plantation area covered by heavy black clay soils, has been experiencing large yield reductions (approximately 45% of the created potential of the 1960s) in recent times mostly because of waterlogging and its allied problems (Dinka and Ndambuki 2014). Despite the great potential of improved Vertisols management for increased food and feed production, the potentially productive Vertisols located on gentle slopes continue to be underutilized because of socio-technical problems associated with their management (Erkossa et al.,2004). This tends crops to have limited yield potential and little ability to respond to fertilizers. Under traditional management systems, yield from these soils is far below the potential (Erkossa et al., 2004). Considering their large moisture-holding capacity and relatively high fertility, Vertisols are capable of producing many times more food and livestock feed than they do today in the Ethiopian Highlands. Techniques to modify land features and soil properties are needed in order to create a favorable environment for seedling establishment and crop growth (El Wakeel and Astatke, 1996).

Review of Challenges and Prospects of Agricultural Production and Productivity in Ethiopia

The agricultural production trends, throughout the 1980’s up to mid-1990’s, were characterized by wide fluctuations in total output and weak growth, with grain production increasing at a rate of 1.37% annually compared to population growth of 2.9 % (World Bank, 2004). Despite large scale extension efforts since mid-1990s, agricultural performance over the past decade has continued to be weak, with production gains mainly driven by weather and area expansion, and weak yield gains limited to maize. There are multiplicity of factors explaining this poor performance; high rainfall variability, the lack of irrigation investment, weak rural institution limited modern varieties, the lack of animal traction, the lack of mechanization, under investment in agricultural research, weak rural infrastructure and skills on the demand side, poor market linkages, high transaction costs, and weak purchasing power leading to thin and volatile markets, make agriculture more risky and reduce production incentives (World Bank, 2004). But of all the problems the one that matters the most is lack of involvement of technology in the agriculture sector. It is alarming that to this day that data and information doesn’t circulate enough through the industry or business.

Machine Learning to Estimate Surface Soil Moisture from Remote Sensing Data

Soil moisture (SM) is a significant component of the hydrological cycle regulating runoff, vegetation production and evapotranspiration [1]. Soil moisture is a major soil indicator to define and identify agricultural drought. Estimation of soil moisture has applications for identifying early-stage water deficit conditions and evolving drought events for crop yield uncertainty and food security conditions, agricultural insurance, policymaking and decision-making, and crop planning [2,3] especially for the arid and semi-arid parts of the globe. Agricultural drought has a catalytic effect that contributes to social and political conflicts in developing countries [4]. Therefore, soil moisture modeling and monitoring are of increasing interest. Monitoring the spatial and temporal variations of SM is a prerequisite both for mitigating and adapting to climate changes for the sustainability of cropping systems as well as for developing precision agriculture and food security [5–9].

On the methodology:

Soil Moisture Prediction Using Machine Learning Techniques

Soil moisture is the main factor in agricultural production. Its prediction is essential for ration use and management of water resources. It is difficult to establish an ideal mathematical model for soil moisture prediction. This paper tests linear regression, support vector machine regression, PCA and Naive Bayes for soil moisture prediction 12 - 13 weeks ahead.

The introduction starts by describing the importance of agriculture for the Indian population. Further it describes the importance of soil moisture from serving as a solvent and food nutrients carrier to boost plant growth, to the fact that temperature of soil is regulated by the movement of water in the soil, to its being essential for photosynthesis.

Problem statement of the paper is that too little moisture can lead to plant death. One of the objectives of the research is to predict soil moisture required for the next three months of a year by using the machine learning algorithms. Also, it studies the machine learning approach for detection of the patterns by using both supervised and unsupervised learning algorithms.

Tests are conducted by collecting data from fields across the 13 districts of West Bengal, India. The research found out that machine learning models can be used to predict the soil moisture required in the upcoming years with significant accuracy depending on temperature and humidity content of the atmosphere. Moreover, it shows that accuracies rise when data is converted into a time series supervised learning format.

By using measurements such as F score, the research also suggests that for this use case multivariate linear regression was seen to produce a relatively more accurate model.

Methodology

Software Design

Coding Platform and Selection

The main coding platform for this program is python. Python is selected mainly because the project deals with data collection and manipulation using machine learning. Beside more high-level languages like MatLab and Octave, python is the best language to work on the above specified topics. Python provides useful tools such as numpy and pandas which facilitate working on large chunks of data and matrices.

The database is designed using the MySQL service. The project requires making relationships between several entities such as sampling sites, devices and readings. For this reason a relational database was found to be fit, MySQL being the most used relational database service.

Design

For this semester, the data is collected using a simple serial-read method, connecting the microcontroller with a laptop computer using a serial interface. This is done because the LoRaWAN gateway - the core component for the LoRaWAN architecture implementation - is in delivery from the US. Once the LoRaWAN gateway is received, the serial communication with the end-devices will be totally migrated to the LoRaWAN architecture. This will lead to a continuous stream of data which requires no human supervision.

The received raw data is pre-processed using a pre-processor module developed in python. SInce the data is now being collected using the serial interfacing method with a human supervision on site, a chunk of data is collected at irregular timing. To remove duplicate data and clean it, data collected within an hour interval are averaged. After this pre-processing stage, the new data is stored in a separate database.

The pre-processed data is then fed into a multivariate linear regression machine learning algorithm to produce certain theta values which can be used to predict soil moisture given a set of temperature, pressure and humidity values. The machine learning algorithm writes calculated theta values to a “theta.txt” file.

To make soil moisture predictions, temperature, pressure and humidity forecasts are required. This forecast is supplied by an online service called OpenWeatherMap. The free trial version of this service provides temperature, pressure and humidity values of a selected location for the following five days with a 3-hour step. Collect-Forecast program makes an API call and stores the data into the mysql database.

The Predict program is designed to perform the soil moisture forecast using the machine learning model. The program loads the theta values in the “theta.txt” file and makes a calculation using the temperature, pressure and humidity forecast to find soil moisture for each entry.

The system is described in detail using Data Flow Diagrams in the following section.

Data Flow Diagrams

Level 0 DFD

The entire project is abstracted as follows. The central system that we design – the Soil Moisture Prediction System – takes weather condition values from the environment to generate a machine learning model. That model is used to forecast soil moisture of a specified location 5 days ahead taking weather condition forecast from an online forecast service.

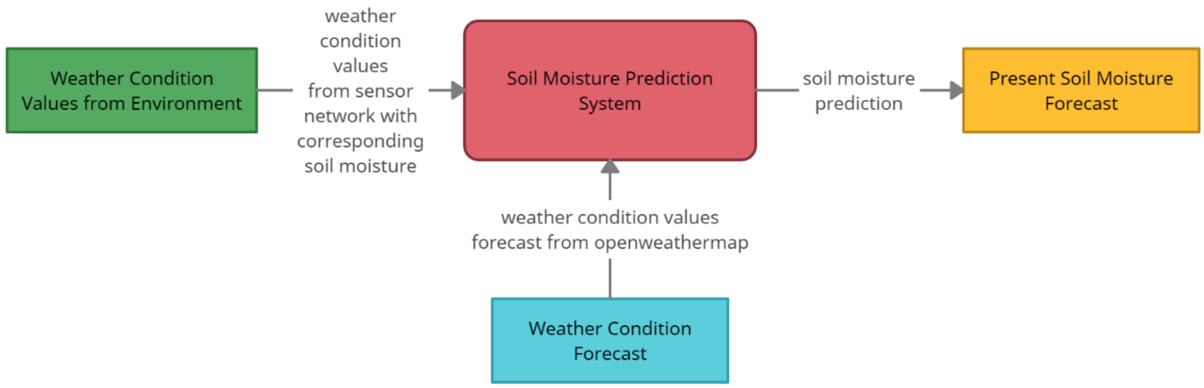


Fig 01. Masa Data Driven Agriculture System DFD

Level 1 DFD

The level 1 DFDs describe each of the blocks in the Level 0 DFD.

The weather conditions are from the environment are sampled and collected using a LoRaWAN sensor network. The collected values are stored in a MySQL database. The raw data collected is stored into two identical tables parallelly. The reason behind this is explained in the Databases section.

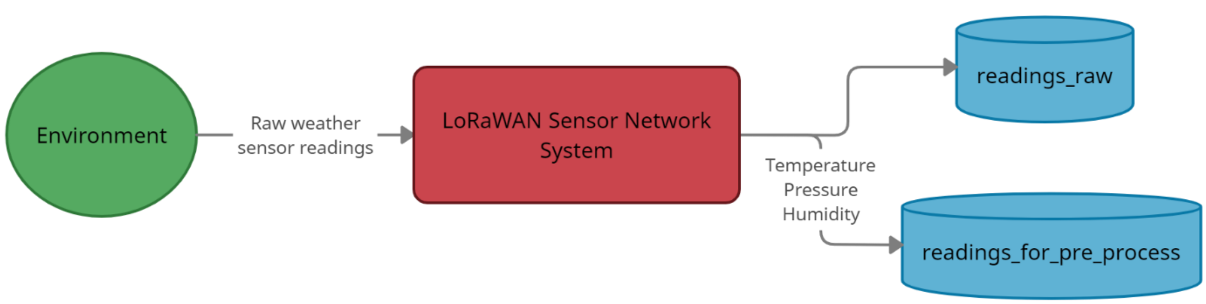


Fig 02. Weather Condition Values from the Environment DFD

The abstraction behind the soil moisture prediction is described below. The raw data is pre-processed and fed into a machine learning algorithm to produce a theta value which hold the constants of the linear equation used to predict the soil moisture. Weather forecast data is then used as input to the equation to output a soil moisture forecast.

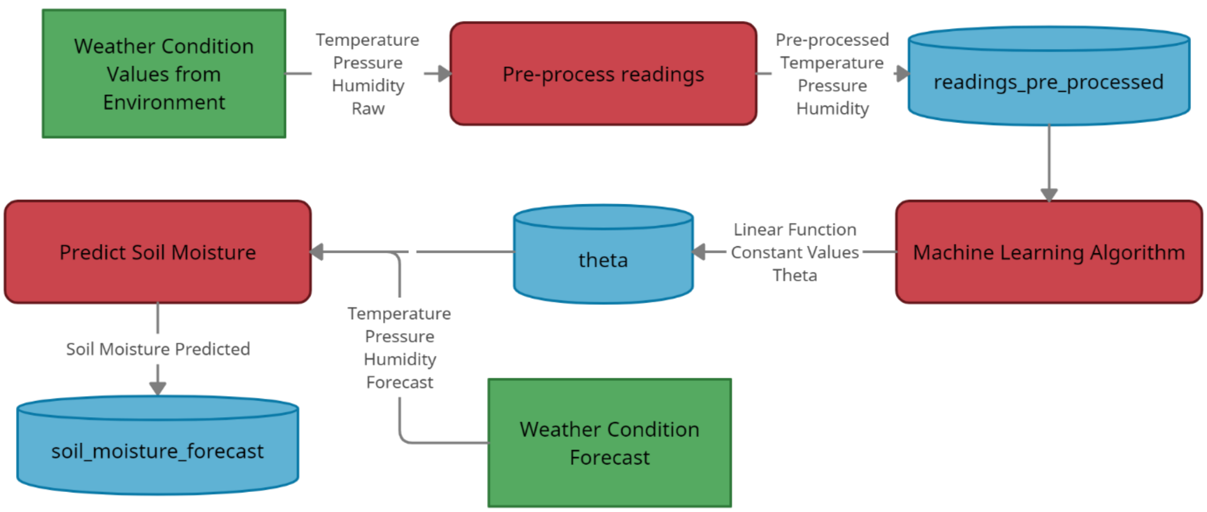


Fig 03. Soil Moisture Prediction System DFD

A 5-day weather forecast is received from an online forecast service called openweathermap. The service provides an API to collect the forecast data. The free trial version only provides forecast for following 5 days with 3-hour difference.

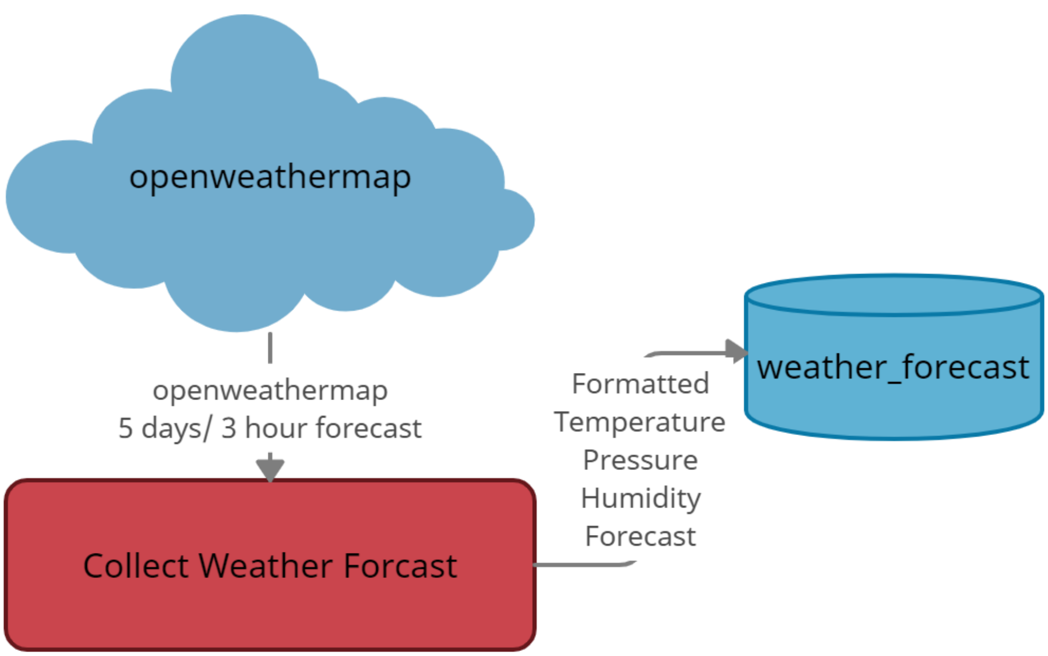


Fig 04. Weather Condition Forecast DFD

The forecasted soil moisture data is displayed to the end-used on a Grafana panel. The panel will show the data on a world map. This gives the user a harmonious experience.

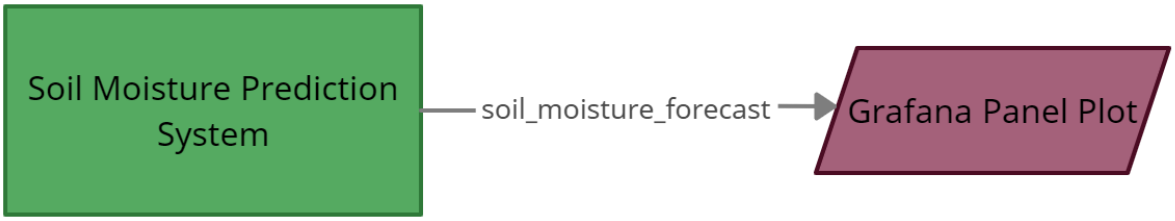


Fig 05. Present Soil Moisture Forecast DFD

Level 2 DFD

The level 2 DFDs further explain processed with in the level 1 DFDs.

A LoRaWAN architecture comprises of LoRaWAN end-devices, LoRaWAN gateway and LoRaWAN Network Server. End-devices are deployed throughout a farming field in selected sites. LoRaWAN gateways are deployed at an optimum distance from the deployed end-devices. The gateways should be at a distance where they can make a connection with the end-devices. The gateway then forwards the data from the end-devices to the network server. The network server is common to the entire system.

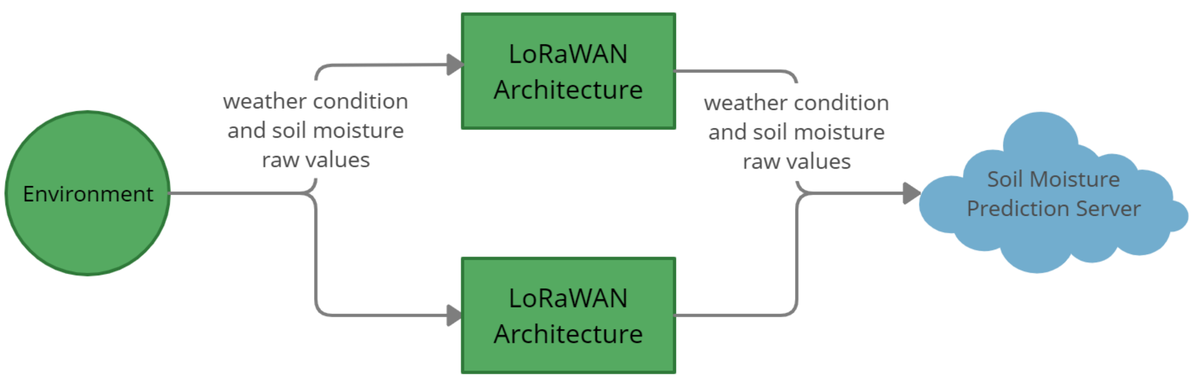


Fig 06. LoRaWAN Sensor Network System DFD

In the multivariate liner regression (machine learning) module processed the raw data in the following was. The raw sensor readings are passed into a pre-processor module and is prepared for the machine learning algorithm. An initial set of theta values (normally initialized at 0) as well as the pre-processed data is passed into the gradient descent module. The gradient descent module, after a specified number of iterations, returns the new set of constant values theta.

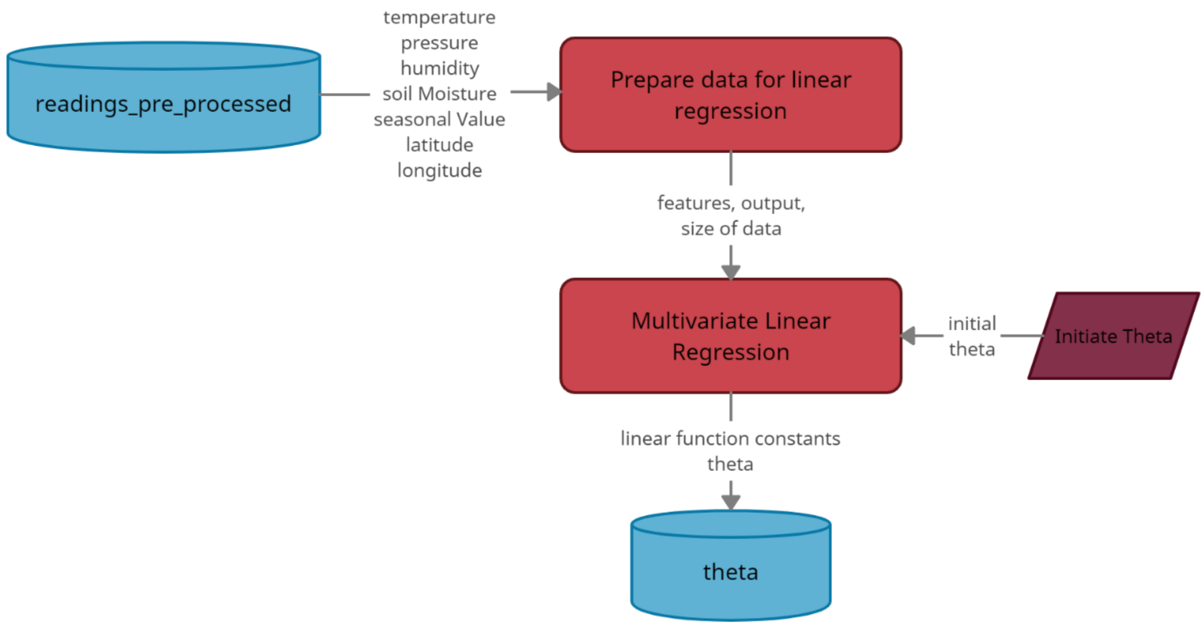


Fig 07. Multivariate Linear Regression DFD

To predict the soil moisture for the following five days, the weather forecast and theta values are used as inputs to the linear equation. The output is then stored in the *soil_moisture_forecast* database.

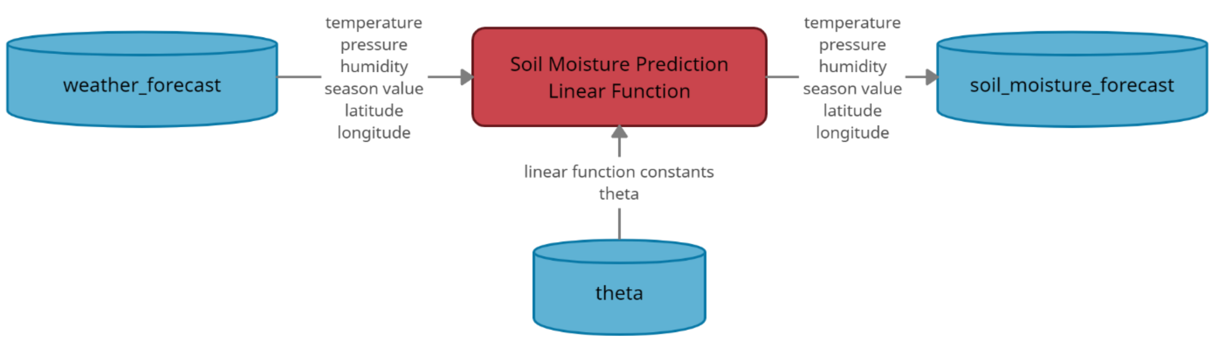


Fig 08. Predict Soil Moisture DFD

Flow Charts

The end-devices sample temperature, pressure, humidity and soil moisture values of their respective sites and send the collected data to the LoRaWAN gateway. Here, the software that the end-device runs is described.

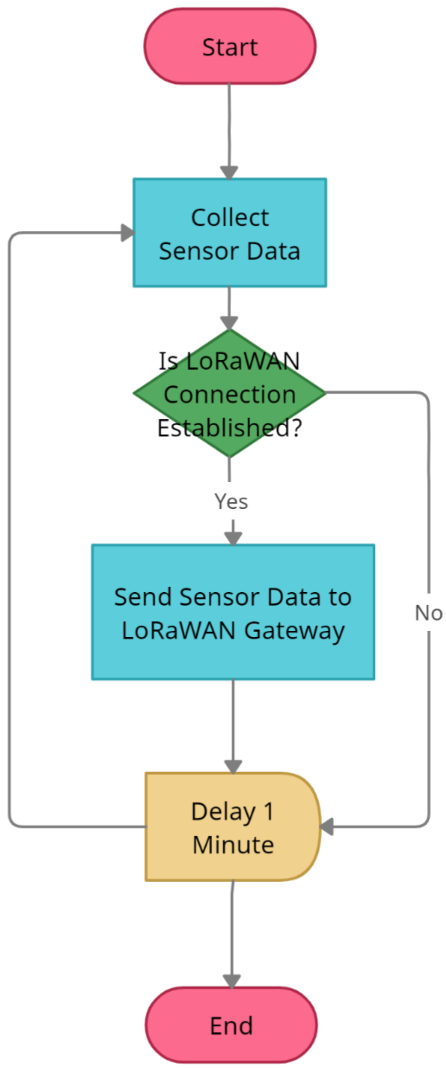


Fig 09. End-device software flow chart

Data pre-processing is an essential part of the machine learning system. The pre-process is done to clean the raw data so as to make the machine learning model accurate. For this project, the data pre-processing focused on compressing the large sum of data collected. Raw data collected within an hour difference is averaged, reducing the size of the data significantly while maintaining the quality. Averaging data collected within an hour is considered because, there is insignificant difference between readings collected in this duration. Also, it can correct any false readings caused by various factors. The working mechanism is described below.

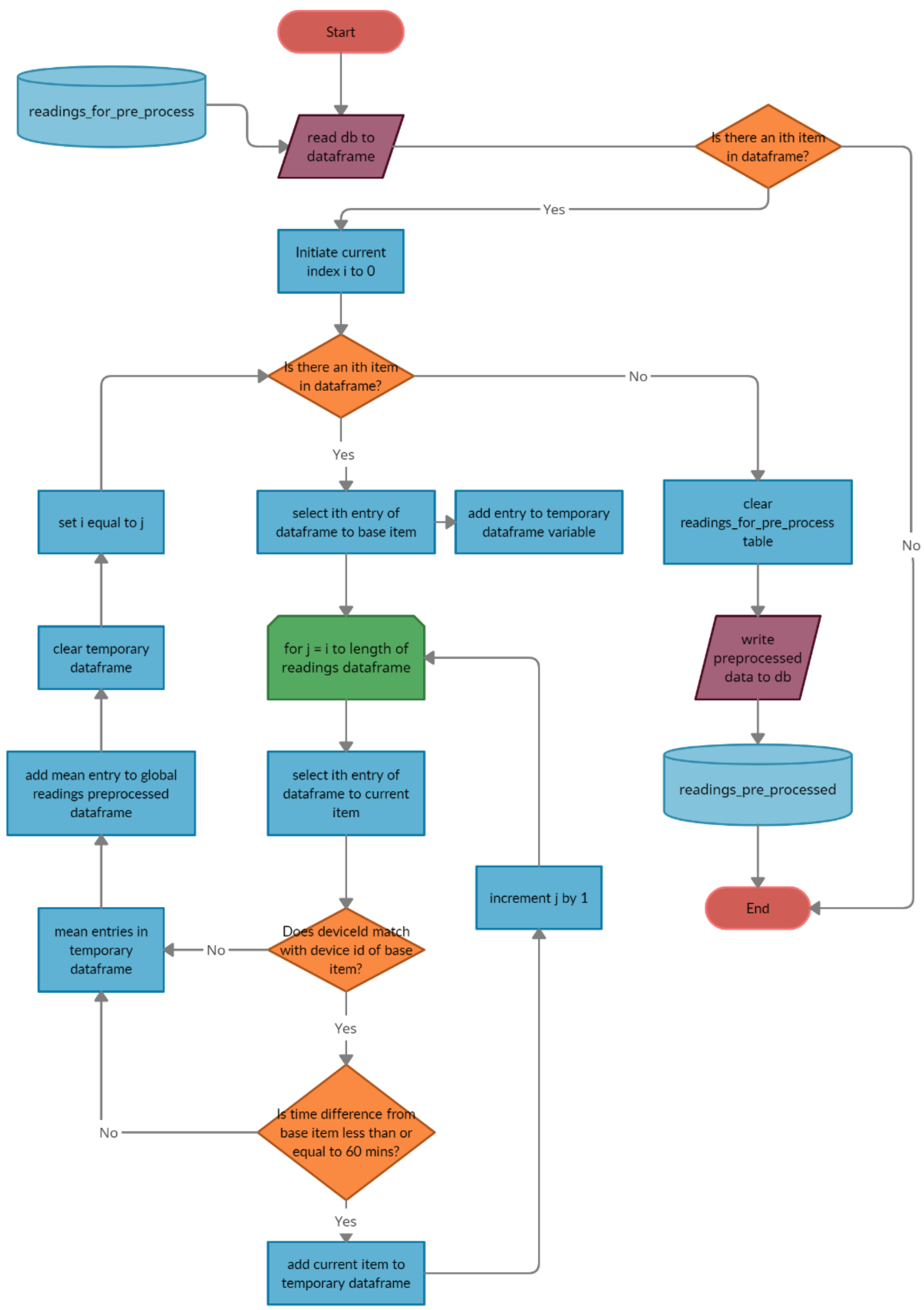


Fig 10. Pre-process readings flow chart

Database Design

The *masadb* database has a total of 8 database tables. Two tables hold the collected reading values. One of the two tables stores the readings values permanently, while the other one is only used as a temporary storage prior to pre-processing.

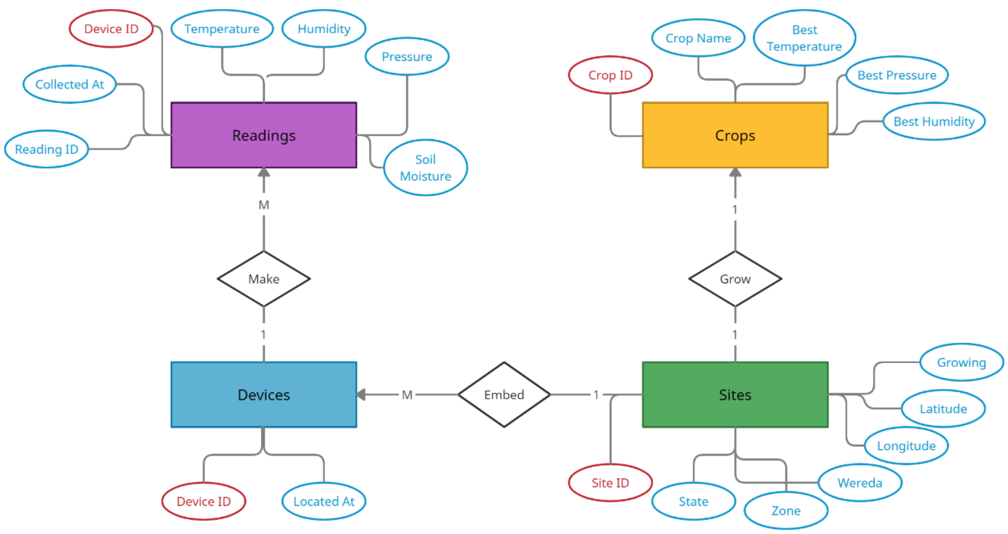


Fig 11. Database ER diagram

The *readings_pre_processed* table holds readings which are prepared for the machine learning processing. *Crops* table holds crop specific data to suggest which crops are best planted where. *Devices* table holds the devices deployed, while *sites* table holds the list of sites which house devices.

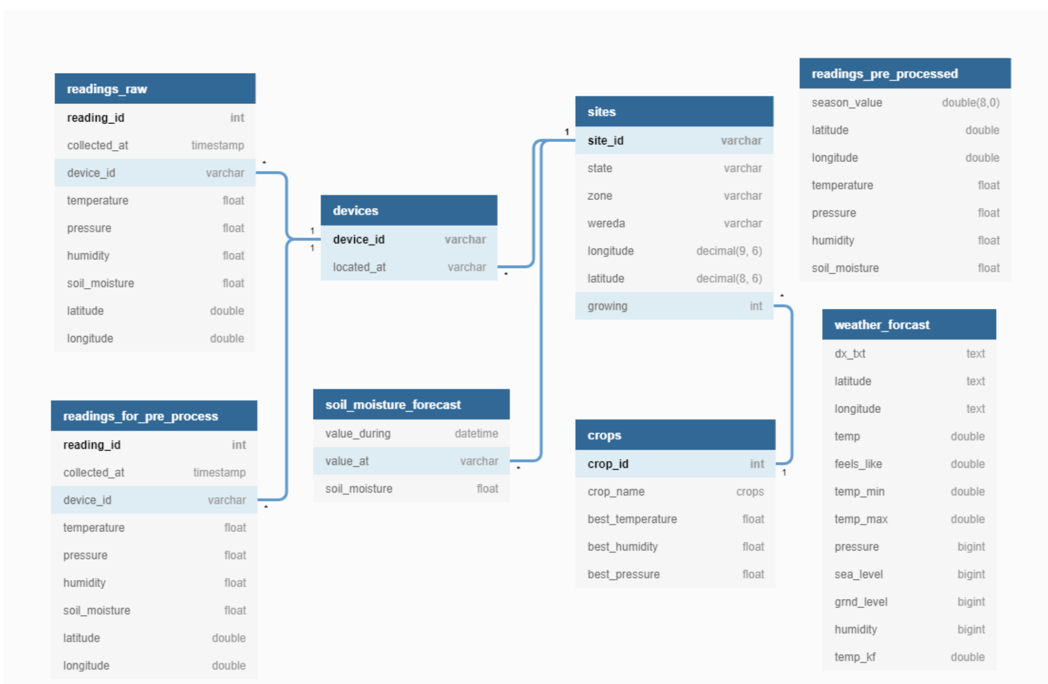


Fig 12. Database Class diagram

Hardware Design

Component Description

LoRaWAN End-device

The LoRa end-device is part of the IoT architecture which samples and sends temperature, pressure, humidity and soil moisture values from a selected farm. Hence, it incorporates sensors to sample the previously mentioned environmental variables on a selected microcontroller unit.

The MCU selected for the end-device is an Arduino Uno. This specific MCU is selected because the end-devices that shield the micro controller require integration with a LoRa module. The LoRa module enables the end-devices to send the collected weather and soil moisture data to the LoRaWAN gateway using the LoRaWAN protocol.

The LoRa module used in this project is a Dragino SX127x Shield. The dragino is designed as an arduino shield. is a Arduino Shield featuring LoRa® technology and based on Open source library.This Shield allows the user to send data and reach extremely long ranges at low data-rates.It provides ultra-long range spread spectrum communication and high interference immunity whilst minimising current consumption.

SX127x Shield is based on Semtech SX1276/SX1278 chip, it targets professional wireless sensor network applications such as irrigation systems, smart metering, smart cities, smartphone detection, building automation, and so on*[9]*. The shield selected for this project sends LoRa packets in 868MHz frequency.

BME280

For this project the BME280 sensor is selected to collect temperature, pressure and humidity data. Alternatively, a DHT sensor was used to sense temperature and humidity values and compare them to data collected using the BME280 to validate the data. Following the test, the BME280 sensor was found to be reliable. The BME280 sensor was integrated with the Arduino Uno board.

The BME280 is integrated simply by using the Adafruit BME280 library. There was a challenge when trying to read the weather condition values using the BME280 sensor. After properly integrated, the serial monitor was not showing the sensor readings. However, this was solved by passing “0x76” as an argument to the *bme.begin* function. Finally, after correctly reading the temperature, humidity and barometric pressure values, these values were passed to the CayenneLPP library functions to be compressed and added into the uplink data buffer.

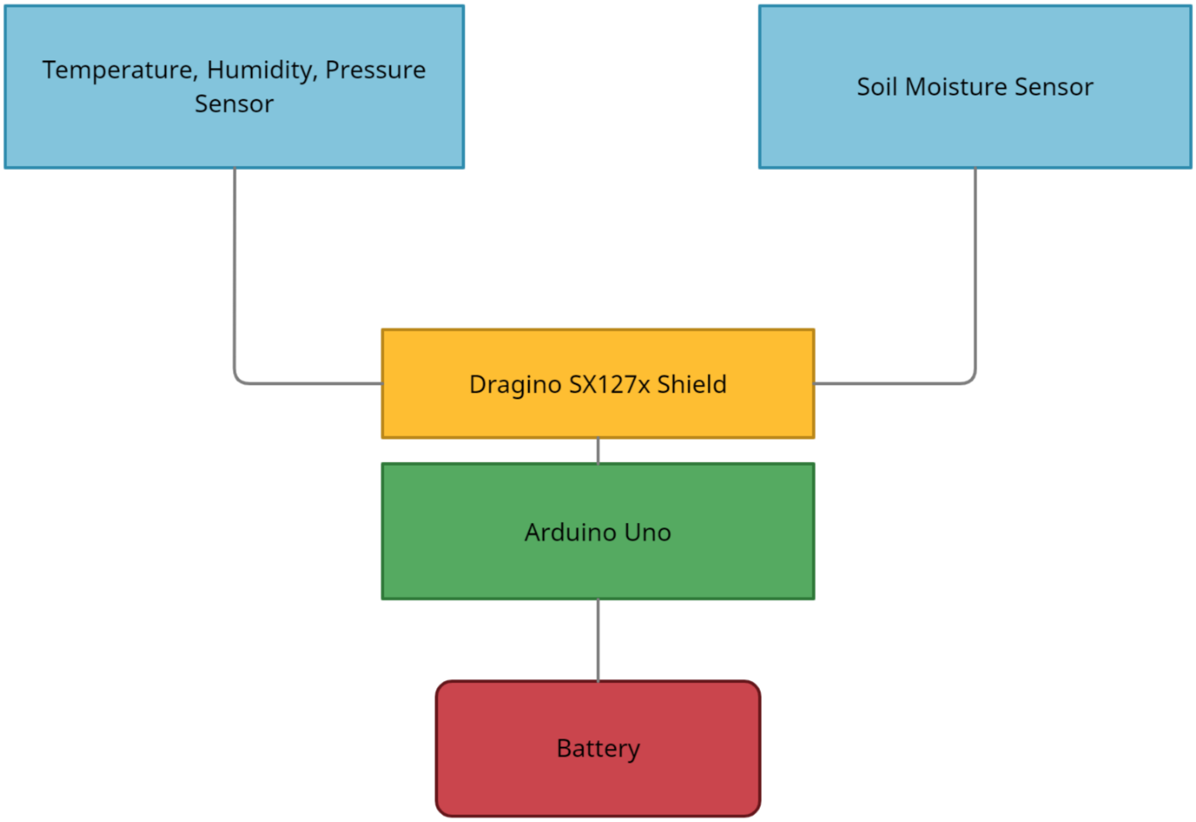


Fig 13. End-device Block Diagram

Item Name	**Number of Item**	**IO Pin Per Item**	**Arduino Pin Con**
Temperature, Pressure, Humidity Sensor	1	4	5V, GND, A5, A4
Soil Moisture Sensor	1	3	3.3V, GND, A0
Dragino LoRa Shield	1		

Fig 14. Used Items

Circuit Layout

The following diagram was sketched using the circuit designing tool provided on circuito.io

The diagram shows the interconnection of the sensor with the MCU. In this diagram, the Dragino LoRa Module is not presented because of limitation in the available block diagrams list. However, even with the Dragino shield the interconnection with the BME280 and soil moisture sensors is as follows.

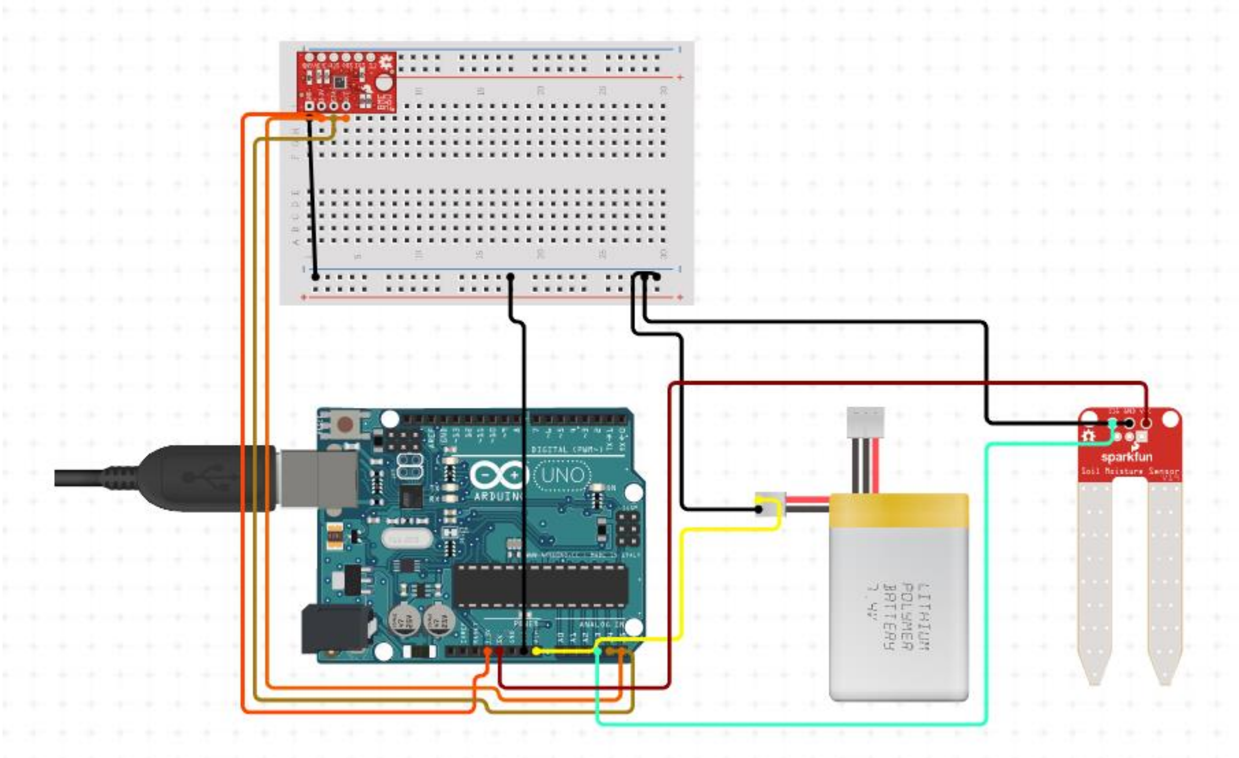


Fig 15. End-device circuit design

LoRaWAN Gateway

The LoRaWAN gateway is a central hub which collects LoRaWAN packets sent by LoRaWAN end-devices. The gateway decrypts the incoming packets and transfers it to an integrated database. The gateway uses a Raspberry Pi 4 as a base machine with a RAK2245 Pi Hat packet concentrator plugged into the Raspberry Pi.

The RAK2245 is based on the Semtech® 1301 and dual SX1257/58 front-end chips. It is a full-fledged 8 channel LoRa® concentrator with a number of improvements over the last generation (RAK831) of this product. [10]

The gateway works using an open-source network server stack called Chirpstack. The ChirpStack open-source LoRaWAN Network Server stack provides open-source components for LoRaWAN networks. Together they form a ready-to-use solution including a user-friendly web-interface for device management and APIs for integration. The modular architecture makes it possible to integrate within existing infrastructures. All components are licensed under the MIT license and can be used for commercial purposes. [11]

The Chirpstack network server stack has three components. The first component, the gateway-bridge, converts received LoRa packets to a JSON data format. The second component, the network server, performs several tasks. These tasks range from authentication to de-duplication of received frames by gateways connected to the LoRaWAN architecture. It also communicates the data to the next component of the network stack, the application server. The application server is responsible for the device "inventory" part of a LoRaWAN infrastructure, handling of join-request and the handling and encryption of application payloads. [12]

The end-device is added to an application created in the Chirpstack application server. An “application” is a container for end-devices which perform related tasks. End-devices are added to the application along with network session key, application session key and device address. These keys are used to allow communication between end-devices and the gateway and to decrypt received data.

Results

Previously, the data to train the machine learning model was collected through a serial communication between the end-device and the server. After configuration of the LoRaWAN gateway, the IoT architecture started functioning well and data was collected using the LoRaWAN protocol. The end-device sends an uplink message containing the collected sensor values every 10 minutes.

Fig 16. Data collection on site-001

Data Collection

Using the serial interface data was collected from two sites. Here is the “sites” table in the database.

site_id	state	zone	wereda	longitude	latitude	growing
site-001	oromia	finfine-special	legetafo	38.886231	9.074487	1
site-002	addis-ababa	arada	01	38.763338	9.040695	1

Fig 17. sites table in database

In each site an end-device is deployed. The end-devices are mapped to the sites they are deployed at by the “located at” parameter in their entry.

device_id	located_at
dragino-end-device-001	site-001
dragino-end-device-002	site-002

Fig 18. devices table in database

Every time a set of temperature, pressure, humidity and soil moisture sensor values are read, the system inserts the data to both the *readings_raw* and the *readings_for_pre_process* tables. The raw data is first pre-processed before the machine learning algorithm is applied to it. To prevent data duplication in the pre-processed data table, as soon as readings are pre-processed they are removed from the *readings_for_pre_process* table. However, to keep a record of all readings, all the readings are duplicated in the *readings_raw* table. A sample of what the *readings_raw* and *readings_for_pre_process* hold is presented in the following image.

reading_id	collected_at	device_id	temperature	pressure	humidity	soil_moisture	latitude	longitude
46	2021-06-01 16:02:35	dragino-end-device-001	22.04	757.66	43.22	38	9.040695	38.763338
47	2021-06-01 16:02:52	dragino-end-device-001	21.3	757.7	43.46	38	9.040695	38.763338
48	2021-06-01 16:03:09	dragino-end-device-001	21.08	757.66	42.68	38	9.040695	38.763338
49	2021-06-01 16:03:26	dragino-end-device-001	21.2	757.64	42.94	38	9.040695	38.763338
50	2021-06-01 16:03:43	dragino-end-device-001	21.13	757.67	44.49	38	9.040695	38.763338
51	2021-06-01 16:04:00	dragino-end-device-001	20.94	757.68	43.9	38	9.040695	38.763338
52	2021-06-01 16:04:17	dragino-end-device-001	20.84	757.66	46.32	38	9.040695	38.763338
53	2021-06-01 16:04:35	dragino-end-device-001	20.58	757.67	45.73	38	9.040695	38.763338
54	2021-06-01 16:04:52	dragino-end-device-001	20.62	757.68	44.65	38	9.040695	38.763338
55	2021-06-01 16:05:09	dragino-end-device-001	20.57	757.7	44.53	38	9.040695	38.763338
56	2021-06-01 16:05:26	dragino-end-device-001	20.51	757.69	46.29	38	9.040695	38.763338
57	2021-06-01 16:05:43	dragino-end-device-001	20.3	757.67	45.88	38	9.040695	38.763338
58	2021-06-01 16:06:00	dragino-end-device-001	20.21	757.7	48.27	38	9.040695	38.763338
59	2021-06-01 16:06:17	dragino-end-device-001	20.07	757.71	46.89	38	9.040695	38.763338
60	2021-06-01 16:06:35	dragino-end-device-001	20.11	757.72	46.87	38	9.040695	38.763338
61	2021-06-01 16:06:51	dragino-end-device-001	20.13	757.75	48.98	38	9.040695	38.763338
62	2021-06-01 16:07:09	dragino-end-device-001	20.09	757.69	49.01	38	9.040695	38.763338
63	2021-06-02 08:14:55	dragino-end-device-001	22.78	759.78	49.18	37	9.040695	38.763338
64	2021-06-02 08:15:12	dragino-end-device-001	22.86	759.77	51.48	37	9.040695	38.763338
65	2021-06-02 08:15:29	dragino-end-device-001	23.03	759.77	49.77	36	9.040695	38.763338
66	2021-06-02 08:15:47	dragino-end-device-001	22.94	759.75	50.33	36	9.040695	38.763338
67	2021-06-02 08:16:04	dragino-end-device-001	22.91	759.74	48.56	36	9.040695	38.763338
68	2021-06-02 08:16:21	dragino-end-device-001	23.09	759.72	53.39	37	9.040695	38.763338
69	2021-06-02 08:16:38	dragino-end-device-001	23.16	759.71	48.44	37	9.040695	38.763338
70	2021-06-02 08:16:55	dragino-end-device-001	23.36	759.71	51.7	37	9.040695	38.763338
71	2021-06-02 08:17:12	dragino-end-device-001	23.34	759.7	50.84	38	9.040695	38.763338
72	2021-06-02 08:17:29	dragino-end-device-001	23.09	759.67	49.13	37	9.040695	38.763338
73	2021-06-02 08:17:46	dragino-end-device-001	22.78	759.69	49.92	37	9.040695	38.763338
74	2021-06-02 08:18:03	dragino-end-device-001	22.82	759.69	52.47	37	9.040695	38.763338
75	2021-06-02 08:18:21	dragino-end-device-001	23.04	759.66	52.96	37	9.040695	38.763338
76	2021-06-02 08:18:38	dragino-end-device-001	22.43	759.66	53.07	36	9.040695	38.763338
77	2021-06-02 08:18:55	dragino-end-device-001	22.26	759.64	52.57	37	9.040695	38.763338
78	2021-06-02 08:19:12	dragino-end-device-001	22.21	759.63	54.66	37	9.040695	38.763338
79	2021-06-03 10:54:27	dragino-end-device-001	19.91	759.97	60.59	40	9.040695	38.763338
80	2021-06-03 10:54:43	dragino-end-device-001	19.67	760.02	56.66	40	9.040695	38.763338

Fig 19. readings_raw table in database

Data Pre-processing

The collected raw readings are prepared before being passed on to the machine learning algorithm. Once the raw data is passed through the pre-processor module, the following *readings_prepared* data is stored.

season_value	latitude	longitude	temperature	pressure	humidity	soil_moisture
13104292	9.040695	38.763338	20.6894	757.685	45.5359	38
13162624	9.040695	38.763338	22.8813	759.706	51.1544	36.8125
13258595	9.040695	38.763338	19.555	759.996	59.2725	39.625
13373461	9.074487	38.886231	16.7412	760.215	50.1963	34
13406671	9.074487	38.886231	14.0631	759.554	59.4831	34
13696523	9.040695	38.763338	26.0243	758.786	29.8262	48.381

Fig 20. readings_prepared table in database

The pre-processed data is then fed into the machine learning algorithm.

Machine Learning Model Validation

One goal of this project is to benchmark several machine learning algorithms and see which one produces the best prediction of soil moisture values. To achieve these goals, multi-variate linear regression and random forest algorithms were used.

For producing the machine learning model and validation, the raw data was divided into two. 80% of the raw data was allocated for pre-processing and machine learning model generation. The remaining 20% was used for validation.

The multi-variate linear regression algorithm validation showed a 30 - 35% percentage difference between the actual soil moisture values.

```
Percentage difference between predicted and actual soil moisture reading values
0      34.701532
1      34.620486
2      33.181999
3      33.133092
4      33.035961
5      33.000544
6      32.938148
7      31.505832
8      31.385677
9      31.357163
10     31.364519
11     31.279822
12     31.286629
13     31.265069
14     31.250464
15     31.322467
16     31.428692
17     29.975558
18     30.230153
19     30.359945
```

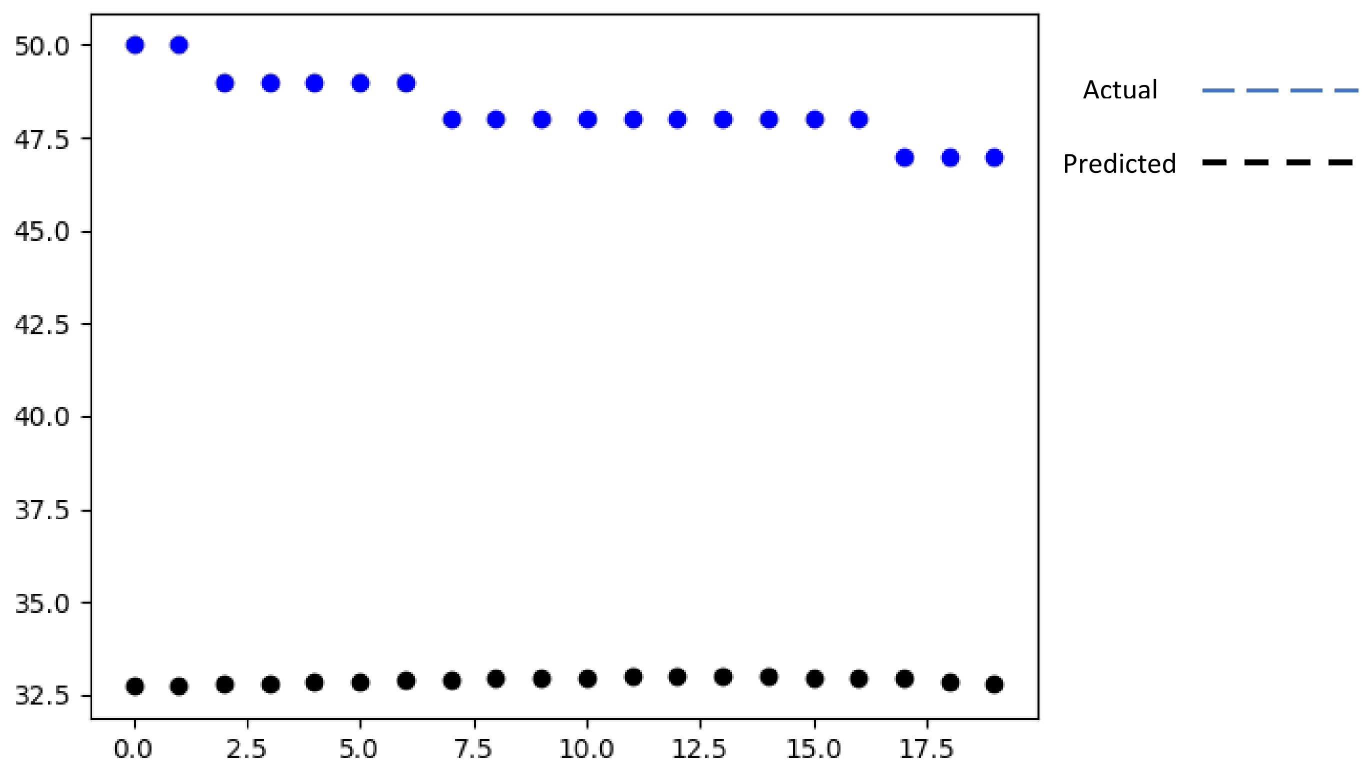
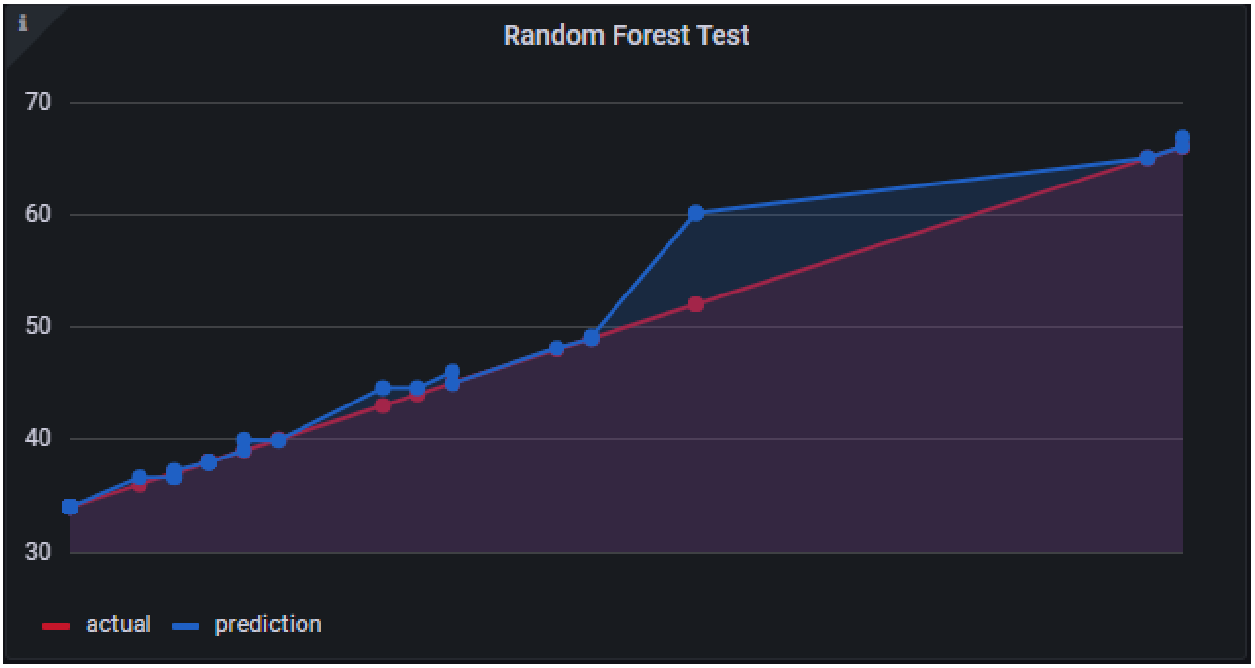


Fig 21. Multivariate linear regression test

Using random forests, a higher percentage of accuracy in prediction was recorded.




```
[38. 38. 38. 38. 38. 38. 38. 38. 38. 38. 38. 38. 38. 38. 38. 38. 38. 37.
 37. 36. 36. 36. 37. 37. 37. 38. 37. 37. 37. 37. 36. 37. 37. 40. 40. 40.
 40. 40. 40. 40. 40. 40. 39. 40. 39. 39. 39. 39. 39. 34. 34. 34. 34. 34.
 34. 34. 34. 34. 34. 34. 34. 34. 34. 34. 34. 34. 34. 34. 34. 34. 34. 34.
 34. 34. 34. 34. 34. 34. 34. 34. 34. 50. 50. 50. 49. 49. 49. 49. 49. 48.
 48. 48. 48. 48. 48. 48. 48. 48. 48. 47. 47. 47. 67. 67. 52. 52. 48. 46.
 45. 45. 44. 44. 44. 44. 44. 44. 44. 43. 43. 44. 44. 68. 69. 67. 67. 66.
 66. 66. 66. 66. 66. 66. 65. 65. 65. 65. 65. 65.]
The actual values:
[48. 52. 38. 37. 34. 34. 34. 43. 45. 34. 49. 39. 34. 34. 37. 39. 37. 36.
 49. 34. 45. 38. 44. 40. 52. 65. 38. 37. 38. 66. 34. 38. 66. 34. 34.]
The predicted values:
[48.118 60.13  37.906 37.227 34.    34.    34.    44.569 46.012 34.
 49.159 39.044 34.    34.    36.62  39.973 37.009 36.615 48.979 34.
 44.959 38.    44.571 39.919 60.13  65.024 37.924 36.629 38.    66.809
 34.    38.002 66.028 34.    34.   ]
Mean Absolute Error: 0.67 degrees.
Accuracy: 98.63 %.
```

Fig 22. Random forests test

Soil Moisture Forecast

To predict the soil moisture of a specific time in the future, the function requires a value which describes the season of the selected time, the geolocation of the area for which the soil moisture is to be predicted for, as well as the temperature, pressure and humidity values. This data is found using the free version of an online service called “openweathermap”. In the free trial version, the service providers weather condition values of a specified location 5 days ahead with 3-hour difference. When called, it provides 40 forecast values. These values were stored on a table called *weather_forecast*. The values in this table are saved as follows.

dt_txt	latitude	longitude	temp	feels_like	temp_min	temp_max	pressure	sea_level	grnd_level	humidity	temp_kf
2021-06-14 12:00:00	9.040547129103139	38.76335981254135	20.03	19.66	20.03	21.44	1024	1024	764	60	-1.41
2021-06-14 15:00:00	9.040547129103139	38.76335981254135	18.95	18.55	16.8	18.95	1020	1020	763	63	2.15
2021-06-14 18:00:00	9.040547129103139	38.76335981254135	17.48	16.95	16.2	17.48	1018	1018	764	64	1.28
2021-06-14 21:00:00	9.040547129103139	38.76335981254135	14.93	14.33	14.93	14.93	1016	1016	764	71	0
2021-06-15 00:00:00	9.040547129103139	38.76335981254135	14.89	14.26	14.89	14.89	1015	1015	763	70	0
2021-06-15 03:00:00	9.040547129103139	38.76335981254135	13.25	12.67	13.25	13.25	1016	1016	763	78	0
2021-06-15 06:00:00	9.040547129103139	38.76335981254135	17.13	16.52	17.13	17.13	1016	1016	765	62	0
2021-06-15 09:00:00	9.040547129103139	38.76335981254135	22.02	21.27	22.02	22.02	1012	1012	766	38	0
2021-06-15 12:00:00	9.040547129103139	38.76335981254135	18.74	18	18.74	18.74	1012	1012	764	51	0
2021-06-15 15:00:00	9.040547129103139	38.76335981254135	19.25	18.43	19.25	19.25	1012	1012	764	46	0
2021-06-15 18:00:00	9.040547129103139	38.76335981254135	16.97	16.18	16.97	16.97	1015	1015	764	56	0
2021-06-15 21:00:00	9.040547129103139	38.76335981254135	14.57	13.86	14.57	14.57	1016	1016	763	68	0
2021-06-16 00:00:00	9.040547129103139	38.76335981254135	13.53	12.84	13.53	13.53	1015	1015	762	73	0
2021-06-16 03:00:00	9.040547129103139	38.76335981254135	12.86	12.08	12.86	12.86	1017	1017	763	72	0
2021-06-16 06:00:00	9.040547129103139	38.76335981254135	18.4	17.58	18.4	18.4	1016	1016	767	49	0
2021-06-16 09:00:00	9.040547129103139	38.76335981254135	22.72	21.86	22.72	22.72	1013	1013	767	31	0
2021-06-16 12:00:00	9.040547129103139	38.76335981254135	23.94	22.99	23.94	23.94	1011	1011	766	23	0
2021-06-16 15:00:00	9.040547129103139	38.76335981254135	21.81	20.75	21.81	21.81	1012	1012	766	27	0
2021-06-16 18:00:00	9.040547129103139	38.76335981254135	16.72	15.6	16.72	16.72	1016	1016	765	44	0
2021-06-16 21:00:00	9.040547129103139	38.76335981254135	15.1	14.08	15.1	15.1	1017	1017	764	54	0
2021-06-17 00:00:00	9.040547129103139	38.76335981254135	14.39	13.43	14.39	14.39	1016	1016	763	59	0
2021-06-17 03:00:00	9.040547129103139	38.76335981254135	13.42	12.49	13.42	13.42	1018	1018	764	64	0
2021-06-17 06:00:00	9.040547129103139	38.76335981254135	19.47	18.52	19.47	19.47	1017	1017	768	40	0
2021-06-17 09:00:00	9.040547129103139	38.76335981254135	23.84	23.01	23.84	23.84	1014	1014	768	28	0
2021-06-17 12:00:00	9.040547129103139	38.76335981254135	24.59	23.76	24.59	24.59	1011	1011	767	25	0
2021-06-17 15:00:00	9.040547129103139	38.76335981254135	22.38	21.35	22.38	22.38	1013	1013	767	26	0
2021-06-17 18:00:00	9.040547129103139	38.76335981254135	17.04	16	17.04	17.04	1017	1017	766	46	0
2021-06-17 21:00:00	9.040547129103139	38.76335981254135	15.9	14.88	15.9	15.9	1017	1017	765	51	0
2021-06-18 00:00:00	9.040547129103139	38.76335981254135	14.87	13.8	14.87	14.87	1016	1016	764	53	0
2021-06-18 03:00:00	9.040547129103139	38.76335981254135	13.88	12.79	13.88	13.88	1018	1018	764	56	0
2021-06-18 06:00:00	9.040547129103139	38.76335981254135	19.26	18.31	19.26	19.26	1017	1017	768	41	0
2021-06-18 09:00:00	9.040547129103139	38.76335981254135	24.49	23.62	24.49	24.49	1013	1013	768	24	0
2021-06-18 12:00:00	9.040547129103139	38.76335981254135	26.15	26.15	26.15	26.15	1010	1010	767	17	0
2021-06-18 15:00:00	9.040547129103139	38.76335981254135	23.05	22.04	23.05	23.05	1012	1012	767	24	0
2021-06-18 18:00:00	9.040547129103139	38.76335981254135	17.62	16.69	17.62	17.62	1016	1016	766	48	0
2021-06-18 21:00:00	9.040547129103139	38.76335981254135	15.54	14.82	15.54	15.54	1017	1017	765	64	0

Fig 23. weather_forecast table in database

Soil Moisture Forecast Presentation

The soil moisture is predicted using the weather condition forecast values from openweathermap API as a variable input. The selected machine learning algorithm then uses these inputs and produces soil moisture predictions.

After the soil moisture values are predicted with respect to their longitude and latitude values, they are mapped to a Grafana panel.

This final product aids the end-user - Agricultural Transformation Agency - visualize forecasted soil moisture values.

Fig 24. Soil moisture forecast on Grafana panel

Conclusion

MasaFarms is a project which uses LoRaWAN communication protocol to collect weather and soil moisture values, and use these values to make data-driven forecast of soil moisture. The project generally has three phases. The first phase deals with collecting raw data using an IoT infrastructure. Following data collection, the second phase deals with using the collected raw data to build a machine learning model which can accurately predict soil moisture of a given geolocation given weather condition forecast values. The final phase uses this model to predict soil moisture for up to 5 days in advance and present this forecast for the end-user.

In the first phase, an IoT infrastructure following the LoRaWAN model was implemented. The LoRaWAN end-device sends temperature, pressure, humidity and soil-moisture values collected from farm fields. These values are collected by a RAK2245 LoRaWAN gateway. The collected values are stored in MYSQL database ready to be processed further. The LoRaWAN architecture shows great promise in the area of data-driven agriculture. The LoRaWAN end-devices use a relatively smaller battery consumption. This is particularly useful because in data-driven agriculture applications, sensors are deployed very remotely throughout farm fields.

The second phase takes the stored raw sensor data and builds a machine learning model which uses temperature, pressure, humidity, latitude, longitude and seasonal value as features and soil-moisture as an output. To see which machine learning algorithms is best to predict the soil moisture, multivariate linear regression and random forest algorithms are tested. The collected raw data was divided into two groups for validation. The multivariate linear regression algorithm predicted soil moisture values showed greater variance from the actual values. Random forest showed a better performance. Comparing the actual soil moisture values with the predicted ones showed a mean absolute error of 0.67 degrees and an accuracy of 98.63%. Hence, random forest in a more suitable algorithm for this specific problem as compared to multi-variate linear regression.

The final phase first uses an opensource API to collect weather forecast for 5 days in advance. It then uses this forecast as an input to predict the soil moisture for the upcoming 5 days. Finally, the predicted soil moisture values are presented on a Grafana panel. Grafana provides a user-friendly interface which makes it easy for end users to refer to the provided soil moisture forecast.

In conclusion, MasaFarms provides an end-to-end data-driven agriculture solution to forecast soil moisture of selected farms. This system can be scaled to include more farms, which will make the prediction more robust. The system is designed so that it uses newly collected sensor values to strengthen the machine learning model.

In the future, MasaFarms can be extended so that it provides soil moisture forecast to farmers directly. More machine learning algorithms can be tested to see if there is a more suitable algorithm to produce soil moisture forecast than random forest.

References

[1] Bulti Merga, Abdulatif Ahmed, "A Review on Agricultural Problems and Their Management in Ethiopia" *Turkish Journal of Agriculture - Food Science and Technology, 2019*

[2] Urgessa Tilahun Bekabil, "Review of Challenges and Prospects of Agricultural Production and Productivity in Ethiopia" *Turkish Journal of Agriculture - Food Science and Technology, Vol.4, No.18, 2014*

[3] Chantal Saad Hajjar, Celine Hajjar, Michel Esta and Yolla Ghorra Chamoun " Machine learning methods for soil moisture prediction in vineyards using digital images" *ICESD 2020*

[4] Hamed Adab, Renato Morbidelli, Carla Saltalippi, Mahmoud Moradian and Gholam Abbas Fallah Ghalhari "Machine Learning to Estimate Surface Soil Moisture from Remote Sensing Data " MDPI, 17 November 2020

[5] D. Sermpanos, Marylin Wolf, Internet of Things Systems: Architectures, Algorithms and Methodologies ,Springer, 2018

[6] LoRa Alliance, "What is LoRaWAN specification?", 2021. Available: https://lora-alliance.org/about-lorawan/. [Accessed Jan. 2, 2021]

[7] ATA, Agri Hub project overview, December 2020

[8] Sagarika Paul and Satwinder Singh. 2020. Soil Moisture Prediction Using Machine Learning Techniques. In 2020 The 3rd International Conference on Computational Intelligence and Intelligent Systems (CIIS 2020), November 13–15, 2020, Tokyo, Japan.

[9] Dragino. https://www.dragino.com/products/lora/item/102-lora-shield.html/

[10] RAKWireless. https://store.rakwireless.com/products/rak2245-pi-hat

[11] Chirpstack. https://www.chirpstack.io/

[12] Chirpstack. https://www.chirpstack.io/application-server/

[13] Mapping Soil moisture in Ethiopia using remote sensing techniques https://www.gislounge.com/mapping-soil-moisture-ethiopia-using-remote-sensing-techniques/

[14] Yu Cai, Wengang Zheng, Xin Zhang, Lili Zhangzhong, Xuzhang Xue. 2019. Research on soil moisture prediction model based on deep learning.

[15] Arduino Project Hub, https://create.arduino.cc/projecthub/atlas-scientific/make-your-own-ph-and-salinity-monitoring-system-fb14c1

[16] Grafana Labs, https://grafana.com/docs/grafana/latest/panels/