Design and Prototyping of an Environmental Conservation System Based on Embedded Machine Learning for Precision Farming

By:

Nalwanga Rosemary
Reference Number: 220000300

A dissertation presented to the African Center of Excellence in Internet of Things (IoT), University of Rwanda as partial fulfilment of the requirements for the Master's degree in Internet of Things-Embedded Computing Systems.

Supervised by: Dr Ignace Gatare

Dr Gerard Rushingabigwi

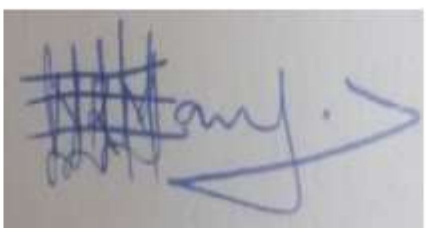
DECLARATION

I NALWANGA ROSEMARY, hereby declare that this dissertation entitled “Design and Prototyping of an Environmental Conservation System Based on Embedded Machine Learning for Precision Farming” is my original work based on a simulation and prototype and has not been submitted for any other degree or professional qualification, except where work that has formed part of jointly-authored publications has been included.

Name of student

NALWANGA ROSEMARY

Signature and Date

……………………………………………………………
…………………………….

BONAFIDE CERTIFICATE

This is to certify that this dissertation entitled "Design and Prototyping of an Environmental Conservation System Based on Embedded Machine Learning for Precision Farming" is a record of the original work done by **Ms NALWANGA Rosemary (Reference Number: 220000300)** a MSc student in Internet of Things Embedded Computing Systems. It has not been submitted for any other degree or professional qualification, except where work that has formed part of jointly-authored publications has been included. The research work has been done under the supervision of Dr Ignace GATARE and Dr Gerard RUSHINGABIGWI.

Main Supervisor:

Dr Ignace GATARE

Co-supervisor

Dr Gerard RUSHINGABIGWI

The Head of Masters and Trainings ACEIoT

Dr James RWIGEMA

ACKNOWLEDGEMENT

First and foremost, I thank and appreciate the Almighty God for enabling me to work through and accomplish this research. I also thank my supervisors and lecturers: Dr Ignace Gatare, Dr Gerard Rushingabigwi, and Dr Jimmy Nsenga for the patience, wisdom, enthusiasm, commitment, and timely feedback that they have rendered throughout this research period. I am so much grateful for the continued support and efforts rendered.

I thank my friends and family for the unwavering support that they have rendered to me throughout this journey I am grateful to each of my friends for their love encouragement and support towards this work. A big thanks to Samson Ooko, for his continued support and guidance throughout this journey. I appreciate Halima so much for the love, care and support rendered to me in various ways to achieve this.

To all individuals that have contributed to the accomplishment of this work academically, spiritually, socially, financially, and other aspects, I appreciate you so much, and may the Almighty God bless you abundantly. My family and friends have been a great source of courage and they have believed in that I can make it. Without them, there would be a great gap.

I also thank the STES Group in Rwanda for giving me a platform to deploy my prototype on their field and dashboard too. Thanks to Toussaint and Wilson plus Mr Nzitonda.

Lastly, I want to thank the IUCEA organization for believing in me and giving me an opportunity of furthering my studies to this level. Thank you so much for your support in every way that was possible.

ABSTRACT

Most of the existing precision agriculture solutions recommend the use of fertilizers as a remedy to poor soil fertility. Such solutions cause environmental degradation in the long run mainly due to the overuse of fertilizers. There is, therefore, a need for a system to ensure that farmers can practice precision farming in terms of a sustainable soil management approach to attain high yields while at the same time conserving the environment. In this research, a design and prototype of an embedded machine learning-based system to predict the best crop to grow with minimal use of fertilizers to conserve the environment is presented. The system senses different real-time soil parameters daily, integrates them with forecast weather information and uses embedded machine learning techniques to determine which crop would grow best under the existing conditions with minimal use of fertilizers. In addition to crop prediction, the system helps farmers to monitor the nutrient evolution of the soil so that action can be done in real time. The results are either displayed on the device or sent to the farmer’s mobile phone. This is a move from the existing solutions that depend on cloud analytics and do not consider the change of soil conditions overtime in making the predictions and decisions. The prototype was tested at STES Group in Rwanda, an innovation and start-ups support hub that provides a commercial smart farming system. The data collected was hosted on a virtual cloud provided by STES so that data can be stored for future use. The implementation of the proposed solution is expected to not only lead to high productivity and reduced costs but also conserve the environment.

Keywords: *Internet of Things, Precision Farming, Embedded Machine learning (ML), Environmental conservation, Deep learning, Crop prediction*

LIST OF ACRONYMS

DL- Deep Learning

AI- Artificial Intelligence

ML- Machine Learning

TinyML- Tiny Machine Learning

NPK- Nitrogen, Phosphorus, Potassium

IDE- Integrated Development Environment

ROM- Read-Only Memory

GSM- Global System for Mobile Communications

IoT – Internet of Things

BLE –Bluetooth Low Energy

LCD- Liquid Crystal Display

RAM- Random Access Memory

STES – Seed Technology Engineering and Science

IIHT - The Indian Institute of Technology Hyderabad

APK – Android Application Package

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER ONE

INTRODUCTION

1.0 Introduction

The demand for agricultural products has mainly been increasing due to the fast growing population and urbanization [1]. To give way to infrastructural development, the increase in population has led to more agricultural land being converted into non-agricultural fields. Adverse changes in climate and wastage of natural resources have also had negative impacts on agriculture [2]. This has led to the application of new technologies to help mitigate the problems leading to the growing popularity of precision agriculture.

Internet of Things (IoT) is used in precision agriculture [3], [4] to help in the optimization of resources, assisting farmers to make informed decisions to achieve high productivity and yields. Digital Farming and Precision Agriculture allow utilization of agricultural inputs like pesticides, water, seeds plus soil fertilizers in right amounts and time for maximizing crop quality, yield and productivity. By deploying sensors for data collection, it enables the farmers to know and understand what to do with their farms in a better way so as to conserve the resources being used and reduce adverse effects on the environment. With solutions that deal with soil parameters, farmers are mostly advised on the nutrients to add, and such solutions may lead to overuse of chemicals and fertilizers; thus causing environmental degradation in the long run. These fertilizers are usually chemicals that, in most cases, kill important microorganisms in the soil that facilitate the conversion of dead animals and plants into organic matter that is rich in nutrients. Synthetic fertilizers that are Nitrogen- and phosphate-based leach into water ways that are underground and make them toxic, leading to water pollution. More to that, chemical fertilizers make the top soil acidic leading to crop burn and hence lower crops yields [5]

Besides, soil fertility is one of the most important elements that determine the growth and ultimate production of crops. Thus, it is one of the primary factors considered in developing precision farming solutions. To begin with, the three main crop nutrients are nitrogen (N), phosphorus (P), and potassium (K) together often referred to as NPK. There are also other important nutrients such as calcium, magnesium, and sulphur, among others, but are needed in small quantities as compared to NPK [6]. Soil Potential of Hydrogen (PH) also affects soil chemical properties and thus fertility. Each and every plant has a preferred level of pH. With the knowledge of what pH level your soil is, it can enhance the ability to determine the particular crops that will thrive greatly and aid in clear decisions about fine tuning the pH level to accommodate specific plants[6].

The convergence of precision farming with artificial intelligence, where farmers can be able to respond to changes in crop growth on time can be a great solution for sustainable food production [6]. Studies and commercial precision farming solutions have considered different parameters in an attempt to determine soil fertility. In a survey of IoT technologies used in the Agricultural environment, different studies emphasize the need to measure soil NPK and soil pH in ensuring a reliable solution [7]. In addition, as also discussed by Lova Raju et al., some studies also include soil moisture, soil temperature, and soil humidity which even though are important to plant growth do not exactly show how fertile a given soil sample is [7]. This narrows down the parameters to be considered for a concrete soil fertility survey to soil pH, soil NPK, and weather conditions.

Most of the crop farmers are based in rural areas that hardly have connectivity, there is no infrastructure to support communication with the cloud which as well comes with an added cost. This comes as a challenge to farmers who may need to practice precision farming through IoT integrated with AI or machine learning in particular to achieve some of the sustainable development goals. Soil being complex and harbouring living organisms constantly evolves in a number of aspects that may be physical, biological and chemical. Testing of soil samples in standard laboratories so as to determine the nutrient content levels is not done frequently because it is expensive and takes time yet nutrient contents vary on short timescales [8].

Existing solutions are either based on offline expensive lab process or use of cloud based architecture in which sensing devices have to collect data and send to the cloud to enjoy machine learning driven intelligence. Due to cost, connectivity, and the need for real-time intelligence, there is a need for a solution that can enable machine learning at the edge to overcome the challenges. This study, therefore, presents an embedded machine learning solution that will also ensure a real time soil fertility management and prediction of crops to be grown thus leading to increased yields while at the same time conserving the environment.

So as to prove the viability of the system, the prototype was deployed at STES group[9]in Rwanda which maintains a smart precision farming system. This provided a virtual cloud that hosted the databases to store the collected data of NPK and soil pH.

1.1 Problem statement

In Africa, agriculture is mainly practiced in rural areas. Most of the time traditional methods of farming are used by small scale farmers. There has been an increasing rural urban migration leading to loss of labour. In addition, this has made rural areas to become less attractive to socioeconomic developments [1]. Moreover, numerous factors have led to constraints in agricultural production systems for smallholders in Africa, these include; excessive waterlogging, low soil fertility, frequent drought and moisture stress, run-off, and soil erosion. This is as a result of using inappropriate farming methods and agricultural inputs, difficulty in accessing the desired markets for inputs and outputs, changing climatic conditions, and weak agricultural extension services. This has led to low productivity, low incomes, increased costs, environmental degradation, labour constraints, and food [12].

Due to the above mentioned challenges, the use of technology is being adopted to enhance precision farming in Africa with many companies offering such solutions. One of the commercially available solutions in Rwanda is provided by STES Group [13]. Even though the commercial and prototyped precision farming solutions have led to an increase in production and reduced costs, on the other hand some have led to environmental degradation. Most of the solutions encourage the use of fertilizers as a remedy to poor soil fertility.

With Synthetic fertilizers, the crops are given a quick boost but do little in stimulating the soil life, improving the soil texture, or improving soil's long-term fertility. This has long-term effects on the soil, leading to pollution of the environment as some of the chemicals find their way into waters sources. It also leads to the burn of crops chemically, an increase in air pollution, acidification, and mineral depletion of the soil. There is, therefore, a need for a solution to minimize the use of fertilizers and still allow the use of the IoT-based precision farming systems but having to ensure the environment is conserved at the same time.

In addition to that, most farmers are based in rural areas where the connectivity is poor and there is no infrastructure to support cloud based solutions to help with their activities. With the urge to incorporate digital farming with AI, there is a need to bring AI close to the farmers that is, pulling it from the cloud to the edge devices.

1.2 Objectives

1.2.1 General Objective

The study aimed to design and prototype an Embedded AI driven System that can predict the best crop to grow with minimal use of fertilizers thereby ensuring conservation of the environment.

1.2.2 Specific Objectives

1. To find out the soil nutrient requirements for crops grown in Rwanda in terms of soil NPK, soil pH, and weather (rainfall, temperature, Humidity).
2. To review existing IoT-enabled precision agriculture solutions.
3. To investigate different AI technologies that can be used in precision farming systems
4. To prototype an environmentally friendly system for precision farming using Internet of Things (IoT) that helps farmers to predict which crop can thrive in the already existing soil nutrients and content.

The study was guided by the following questions

1. Are there environmental challenges that come up due to the excessive use of synthetic fertilizers to the soil?
2. Can we incorporate IoT and Artificial Intelligence to help farmers take an informed decision about the crop to be grown on a fertilizer free basis?
3. What should be considered in developing an environmental friendly precision framing solution through crop prediction?

1.3 Hypothesis

The hypothesis was that IoT technologies can be merged with Artificial Intelligence to enhance environmental conservation through sustainable soil management and crop prediction.

1.4 Study Scope

The study was focused on using ML to predict the best crop to be grown with the existing soil conditions with minimal use of fertilizers. Our prototype will however not recommend the amount of fertilizer to be added. In addition, only 5 crops were considered for prototype design, and these include; beans, maize, lentil, peas, and watermelon. The monitored soil parameters that were considered in the prediction were NPK, pH, and weather information of Temperature, Humidity, and Rainfall

1.5 Significance of the study

This research is expected to contribute towards the sustainable development goals of conserving the environment through guided precision farming in terms of best crop prediction and sustainable soil management. The study gives a farmer a prediction based on a fertilizer free context and thus minimize the overuse of fertilizers which may end up damaging the soil

1.6 Organization of the document

This chapter gave an introduction to the research study and the next chapter which is chapter 2 gives a review of the related literature is presented and the gaps identified. Chapter 3 presents the methodology applied in this research study and Chapter 4 presents the system model and design, simulation models, and simulation parameters. The 5th chapter discusses the results and findings analysed from the research study carried out and lastly Chapter 6 outlines the conclusions and recommendations.

1.7 Summary

This chapter presented an introduction to the study, from the problem statement it is evident that the current precision farming solutions have led to environmental degradation and thus the need for the study. On implementation of the system, the study will among other benefits contribute towards the attaining some of the sustainable development goals in Agriculture, farming and the Environment at large.

CHAPTER TWO
LITERATURE REVIEW

In this chapter, a review of related literature is presented. Literature on digitization of soil nutrient data is first presented followed by precision farming solutions based on soil fertility. The use of artificial intelligence in soil fertility management is presented and lastly, a review of data sets is given and a conclusion drawn.

2.1 Digitizing soil nutrient data

There exist different techniques used to evaluate the soil fertility of a particular field. To begin with, while measuring the soil fertility, the basic method soil samples are mixed with water and the Nitrogen, Phosphate, and potassium as nitrates, phosphates, and potassium chemically extracting. A comparison is done to a colour chart to determine the amount of N, P, and K found in the soil sample [10]. The use of commercial soil NPK and Soil pH sensors has also been proposed in recent studies and solutions. A system is proposed in which the soil nutrients (N, P, K) are measured for rice crops using colour sensor TCS3200 [11]. The use of customized in-house sensors that measure soil chemical properties is proposed [12]. An optical transducer is developed and used to detect and measure the presence of different soil nutrients (N,P,K) in that soil [13]. This sensor helps in deciding how much the needed additional contents of the detected nutrients can be added to the soil to increase soil fertility to the desired value. The N, P, and K values of the sample are determined by light absorption of each nutrient.

2.2 Precision farming using IoT

A framework for precision agriculture [14] uses environmental sensors that are of low cost, with an Arduino board and two wireless transceivers, and an actuating circuit. This provides automated monitoring and irrigation of crops. Solutions using the same concepts of irrigation using IoT are recommended [15] – [16]. Other precision farming solutions based on IoT are proposed [4]; [17]-[18]. Such solutions show the increasing use of Internet of Things in agriculture but do not put into consideration the plant nutrient requirement, a focus of the proposed study, and will also not be applicable in areas where irrigation is not practiced which is the case with many farms in Rwanda. A system based on energy conservation and low cost for smart agriculture is proposed by Kumar et al. This system monitors the soil moisture content and using a sensor developed in-house. The Indian Institute of Technology Hyderabad (IITH) mote is used in the proposed network as both a sensor node and a sink node which provides low-power communication [12]. This however considers just soil moisture to give advice on irrigation. It does not put the aspect of soil content in terms of nutrients into consideration.

2.3 Precision farming solution based on soil fertility

Waddington et al. presents a system that measures the soil nutrients, i.e. NPK, for rice crops using a colour sensor is proposed. It allows the farmer to view the soil fertility status at their convenience on a web application and also suggests which fertilizer they can add to get a better yield [19]. A system is proposed by Pravallika that uses data values of moisture, values of soil pH, values of Temperature, and values of Humidity value from the soil and analyses the soil status [20]. It thereafter helps the farmer to make a thorough analysis of the soil fertility of their field and plant a crop accordingly to increase on crop yield and productivity. This system uses a coding algorithm (data-driven approach) to analyse and predict the soil fertility and suitable crop. Such studies and other related studies show that monitoring the soil NPK and pH is essential when developing solutions that monitor soil fertility. They also support the argument that the fertility of the soil changes overtime.

2.4 Artificial Intelligence in soil fertility management

Artificial intelligence centered with deep learning provides several algorithms that can help in monitoring the health of the soil before planting and during the growth process also. Soil deficiencies can be analysed to ensure smooth crop growth [21]. With soil weakness comes several crop defects and low production; so assessment of nutrient levels in the soil is relevant.

Incorporating the nutrient cycling models that exist with embedded AI approaches of crop productivity can help in optimization of uptake, targets, nutrient capture, delivery, and long-term impacts on soil microbial communities that combine functionality profiles and optimal safety [21]. This can as well help farmers to take timely actions on time to the changes in plant growth based on the soil nutrient status.

According to Vijayabaskaret al. [22], a system that uses predictive analysis to suggest the fertilizer which has to be added to the soil to increase crop productivity is designed. The prediction is done based on a Bayesian algorithm at the cloud to give farmers information after a certain period.

A logistic regression ML algorithm by Ghanshala et al. is used at the cloud in order to analyse data that is being sent from the field [23]. The collected data is based on NPK sensors and, after analysis, information is sent to the farmers to know the status of their farms. A web portal is also created which gives information about the fertilizer(s) required for their crops [24]. Milija Bajčeta et al. developed an IoT-based private cloud platform that is used in ecological monitoring and agriculture [25]. In this paper, IoT nodes are used and they communicate to the server in a cloud gateway or directly. The server serves the purpose of hosting analysis for data, it hosts data integration and remote visualizations, plus smart application development and deployment.

Spandana et al. proposed an application of IoT for soil quality. In the solution, eight different sensors are used to analyse the soil type, soil moisture levels, and soil quality with weather aspects including wind, temperature, and humidity [26]. A node MCU is used with data being sent to the cloud through Wi-Fi technology. A related commercial solution BAZAFARM [9] is used in Rwanda. It works in a network of sensor nodes that collect the soil moisture, temperature, and nutrient content and send data to a master node which then pushes data to the Internet. The collected data can be accessed by users to do the needful in terms of irrigation. The collected data is also sent to a virtual cloud for storage and future use. The decision to irrigate is made based on threshold values of the soil state.

2.5 Datasets

Sources of data include; actual data collection, open-source datasets and synthetic data generation among others. Open datasets from various studies can easily be explored from readily online datasets which provide links to many different data sites. Due to issues of privacy and security concerns, identifying open datasets in some areas of study may be limited. Data collected from African settings are also limited. This pushes the need for the exploration of synthetic data generation to complement the small datasets. Third-party datasets such as weather datasets can also be explored from service providers who either provide them for free or based on subscription fees and can be readily accessed through custom APIs.

There are AI-powered synthetic data solutions that take original data and transform it into privacy-compliant synthetic copies. Synthetic data comes as a solution to the lack of enough datasets that are needed to build strong and accurate machine learning models to aid in prediction systems [27].

Table 1: N, P, K range requirements of each crop [33]

Crop	Beans	Maize	Lentil	Peas	watermelon
NPK (mg/kg)	20, 65, 25	74, 50, 18	20, 70, 19	40, 70, 77	99, 20, 50

2.6 Summary

In most of the aforementioned systems, the need to reduce the application of external chemical fertilizers to the fields is not emphasized yet this has been proven to harm the environment in the long run. Furthermore, some of the systems only present to the farmer the state of their field in terms of soil nutrient content and this leaves them to make uninformed decisions on what to do with the data. In addition, the solutions that recommend the use of predictive algorithms depend on a cloud-based architecture that does not apply to the African setting where connectivity is a challenge. Thus the need for an edge based solution.

CHAPTER THREE
RESEARCH METHODOLOGY

This chapter describes the selected system development methods and the machine learning, data collection tools, synthetic data generation tools, and programming tools.

3.1 Software Development Method

3.1.1 Waterfall Model

The waterfall model was selected as the system development method. This model uses the software development cycle for creating of the system in a sequential and linear approach. This model systematically develops phase after phase following a download fashion [28]. The model is divided into several phases with the output of one phase being the input in the next phase. With this model, phases cannot overlap and one phase has to be completed before moving to the next phase until the process is completed. Figure 3.1 shows the phases in the waterfall model.

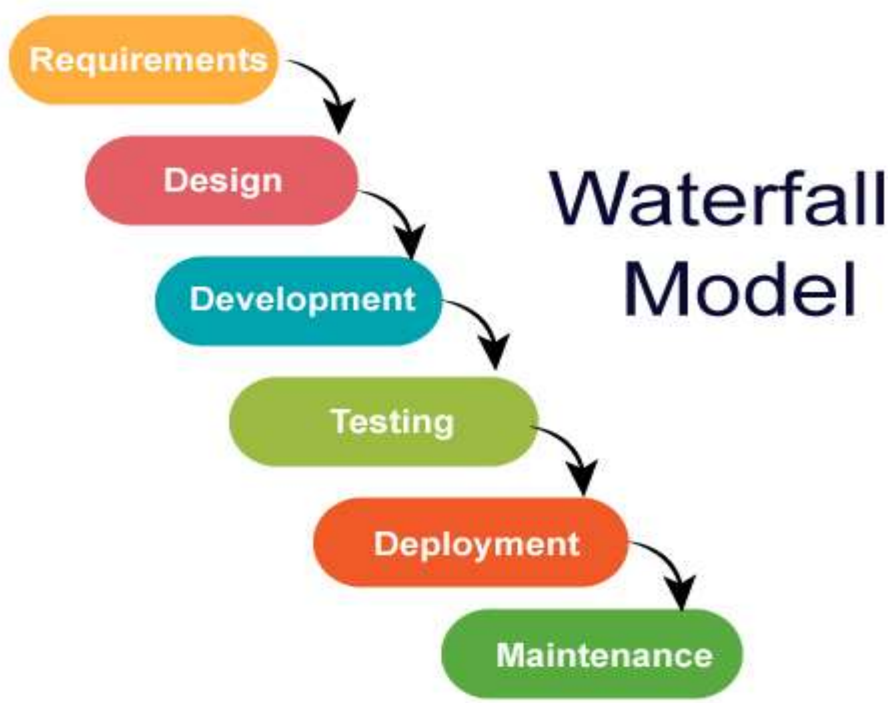


Figure 3.1: The waterfall model

3.1.2 Reasons for using waterfall

The waterfall model was selected due to the following reasons:

- The system requirements for the precision farming system were clearly well documented and not expected to change
- The definition of the proposed product was stable.
- The technologies to be applied were well understood and were not dynamic.
- There were no ambiguous requirements for the system.
- The researchers had the required expertise and were available to support the product development process
- The proposed project was short and had to be completed within a fixed time frame

3.1.3 System development steps

The sequential phases of the waterfall model in software development were followed

i) Requirements Gathering and Analysis

In this phase, the requirements were gathered based on the problem statement and consultation of stakeholders and other service providers and review of related solutions and literature. The requirement was analysed and the system requirements documented.

ii) System Design

The researchers developed the system-level design including the system architecture, block diagram, use case diagram, simulation setup, and prototype design.

iii) Implementation

The researcher worked on the coding for both the simulation and prototype and refined accordingly. The machine learning model was also coded and developed at this phase.

iv) Testing

The model was first tested in the cloud platform before a model to be implemented in the device was simulated and tested. The system was then developed and tested to ensure it meets the user requirements.

v) Deployment and maintenance

The deployment and maintenance phases will be implemented in future works

3.1.4 Advantages of the waterfall model

Some of the advantages for the waterfall model were;

- It was simple and easy to use and understand
- Managing the model was easy given the fact that each phase had a specific deliverable
- Each of the phases was completed one after the other
- The requirements were well understood so the model worked well for our case
- This model is more efficient for smaller projects where there is a clear understanding of the requirements.
- The tasks were easy to arrange
- The results of each process were well documented.

3.2 Embedded ML Process flow

The embedded ML process starts the training of datasets as shown in figure 2, through a synthetic data platform to generate synthetically enhanced data. The datasets were subjected to Mostly AI platform to generate synthetically enhanced data with a relatively bigger volume compared to the original raw dataset. The dataset was collected with the same sensors as per the system design. These included soil NPK, values soil pH, weather parameters of temperature, rainfall, and humidity. The synthetic data forms the input to the Machine Learning process and is used to train a model that can run on edge embedded devices that are resource constrained. An Embedded Machine Learning package is then generated for compilation and simulation. The model is thereafter being subjected to implementation on an embedded device. For our simulation context, test data is used to test the model in the cloud and results compared to when the same is used in the edge simulation environment. Synthetic data generation steps are shown in figure 3.2.

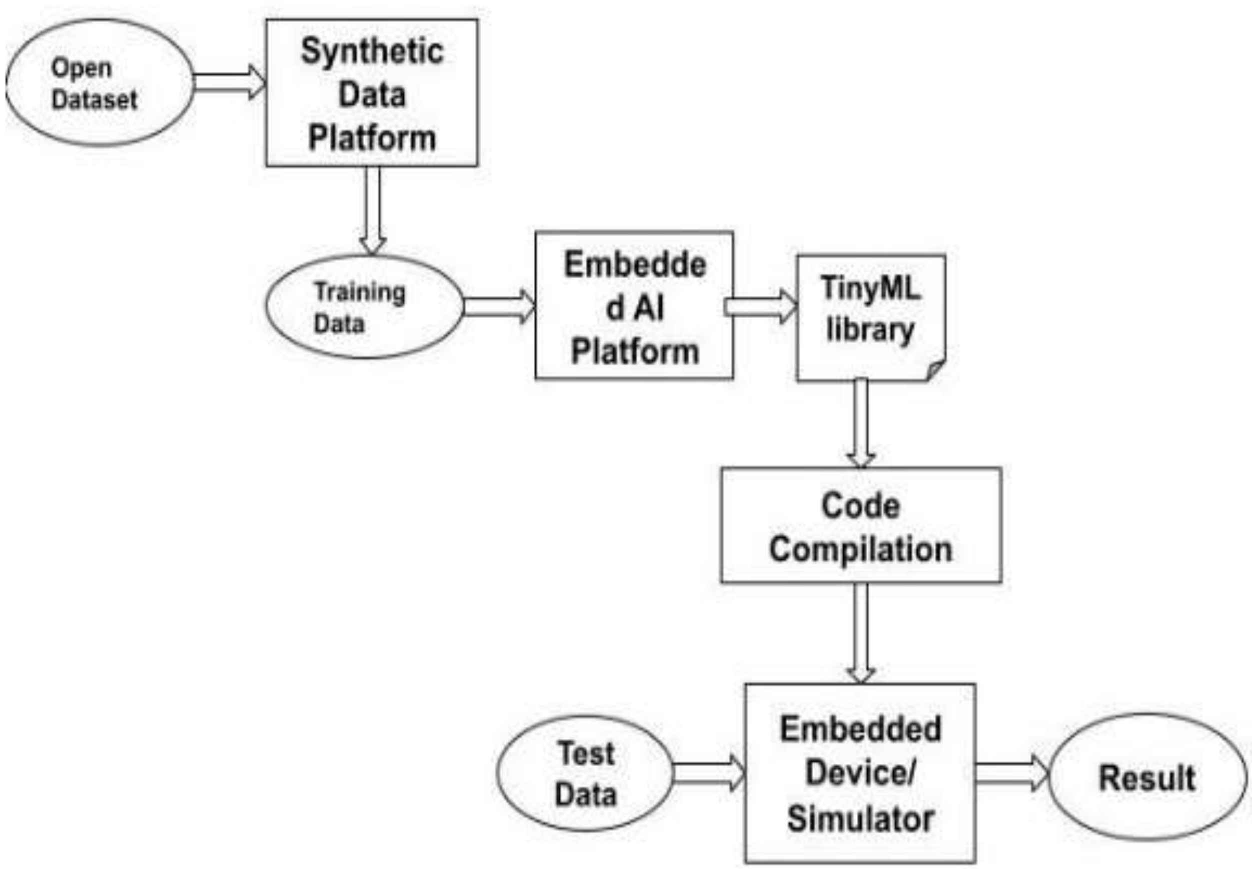


Figure 3.2: The Embedded ML Process

3.2.1 Open Dataset

Due to limited time, we chose to use an open dataset from Kaggle, an online platform which dataset formed a basis for creating the synthetic data used in this study. Data augmentation was used to create the dataset. The augmented data of fertilizer data, climate, and rainfall was collected from India.

Attributes information:

- N - Nitrogen
- P - Phosphorous
- K - Potassium
- pH
- Weather (Temperature, Humidity, Rainfall - Rainfall in mm)

3.2.2 Synthetic Data Generation

Data collected from African settings are limited. This pushes the need for the exploration of synthetic data generation to complement the small datasets. There are AI-powered synthetic data solutions that take original data and transform it into privacy-compliant synthetic copies. Synthetic data comes as a solution to the lack of enough datasets that are needed to build strong and accurate machine learning models to aid in prediction systems. In this case we used Mostly AI which is an open source platform and it trains a model to and allows it retain the granularity and statistical distribution of the data. It retains the data patterns plus correlations and dependencies on time. It then uses its model to generate a synthetic version of the data.

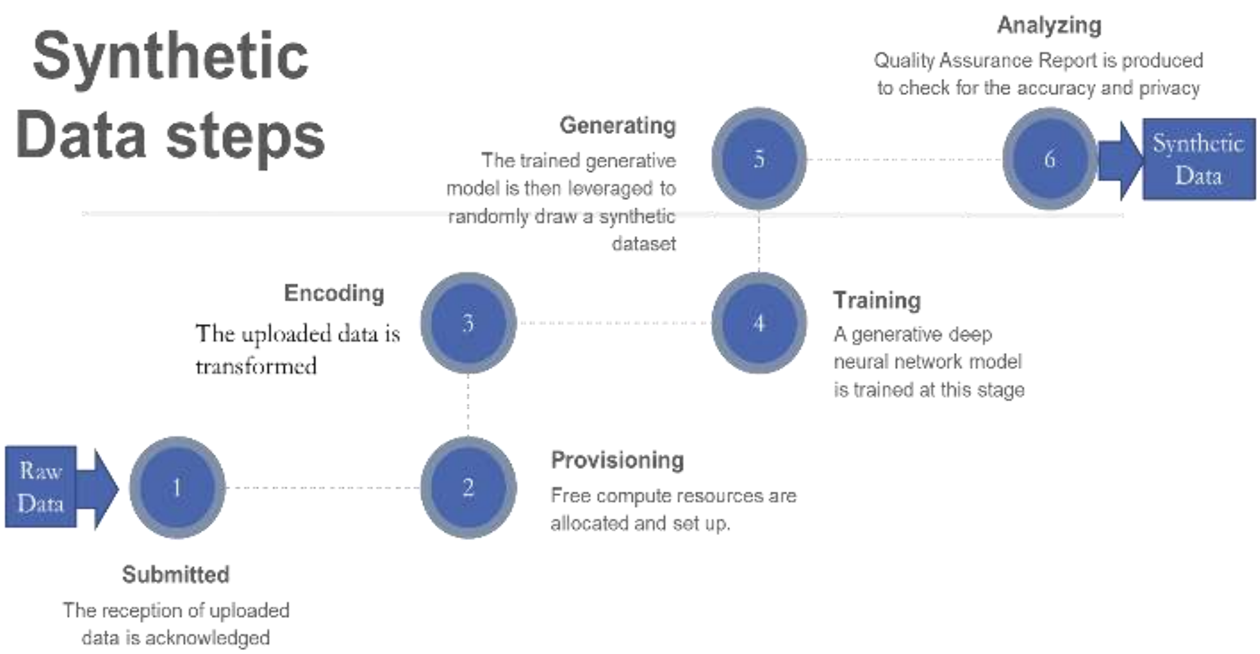


Figure 3.3: Synthetic Data generation steps

3.3 Software Tools

3.3.1 Embedded ML

Embedded ML allows the use of AI in resource constrained smart devices. It is a type of machine learning that enables the shrinking of deep learning networks to fit tiny hardware. Tensor Flow Lite, a machine learning framework for embedded devices created by Google is used in Embedded ML. The framework makes deep learning smaller and faster for implementation in embedded devices.

3.3.2 STM32CubeIDE

STM32CubeIDE in a development platform from ST electronics that can enable code generation, configuration of peripherals, compilation and debugging for STM32 microcontrollers [29]. These microprocessors have the capability to accommodate TinyML models for edge artificial intelligence inferencing.

3.3.3 Proteus Design Suite

The Proteus Design Suite has got an easy to use interface and a powerful feature set that can enable one to rapidly design a system, test the system and come up with a PCB [30]

3.3.4 Mostly Ai

This is an open source platform that was used in the generation of synthetically enhanced data. It trains a model to retain the data's granularity, statistical distributions, patterns, correlations, and time dependencies. It then uses its model to generate a synthetic version of the data.

3.3.5 Edge Impulse

Edge Impulse enables you to finally use Tensor Flow on Microcontrollers. The end-to-end deep learning pipeline enables you to finally create Tensor Flow models.

CHAPTER FOUR

SYSTEM DESIGN AND ANALYSIS

In this chapter, the system design, system design analysis, and system simulation setup are presented.

4.1 System Design

Soil nutrient sensors and soil pH sensors integrated with communication modules and microcontrollers form part of the sensing unit of the system. Different sensor nodes are deployed in a farm. Each node collects data daily for one month and forwards it to the sink node via Bluetooth Low Energy (BLE). The collected data is then aggregated and integrated with forecast weather information at the sink node. An embedded AI model is then used to predict the best crop to grow based on the observed soil parameters with notifications being shown on the device and also sent by SMS to the farmer's mobile phone. Figure 4.1 gives the system architecture.

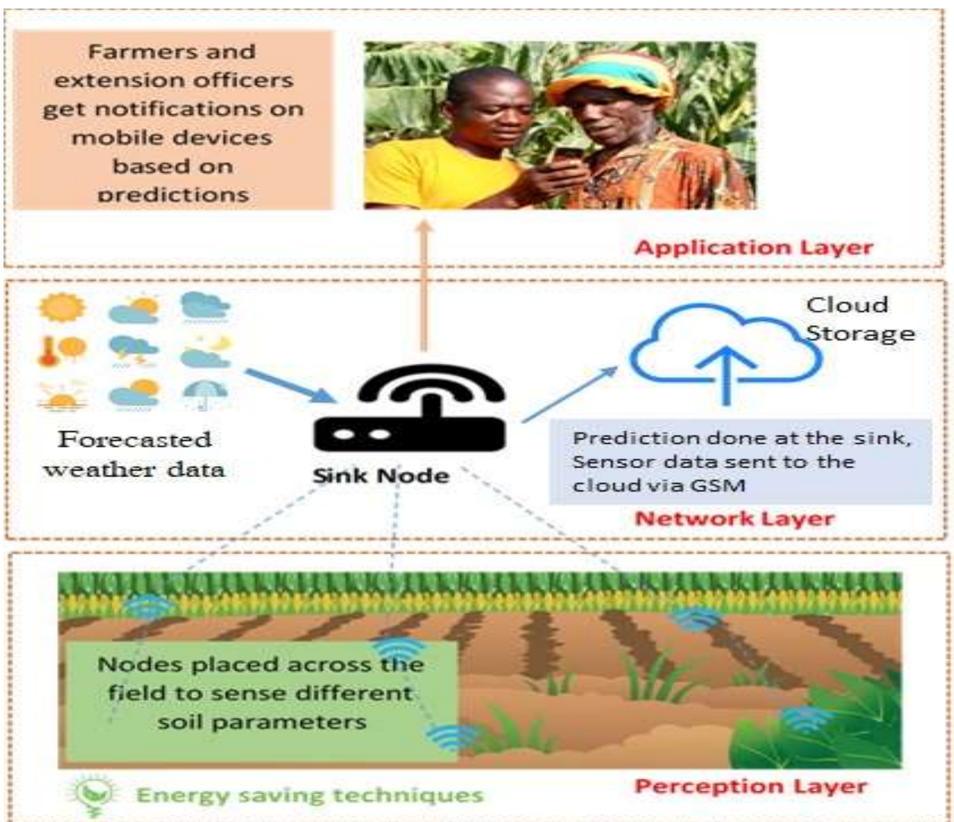


Figure 4.1: High Level system architecture

The data aggregated by the sink node and the prediction results are sent to the cloud through cellular networks. Data will be stored on the ThingSpeak cloud platform with the information being made available through web and mobile based dashboards for future analysis and data sharing. Energy harvesting for the system using solar radiation from the sun is used to power the system.

4.1.1 Embedded System Level Design

The sink node constitutes of an NPK sensor, soil pH sensor, a GSM for connectivity in case a message (SMS) is to be sent to the farmer's mobile phone, an Arduino Nano 33BLE sense with ARM Cortex M4 microprocessor to support the TinyML models, an LCD for display and a solar power harvesting module for powering the device. The other sensor nodes constitute the sensors and the Arduino Nano 33BLE sense. These collect data and send it to the sink node via Bluetooth Low Energy. Figure 4.2 shows the sink node block diagram

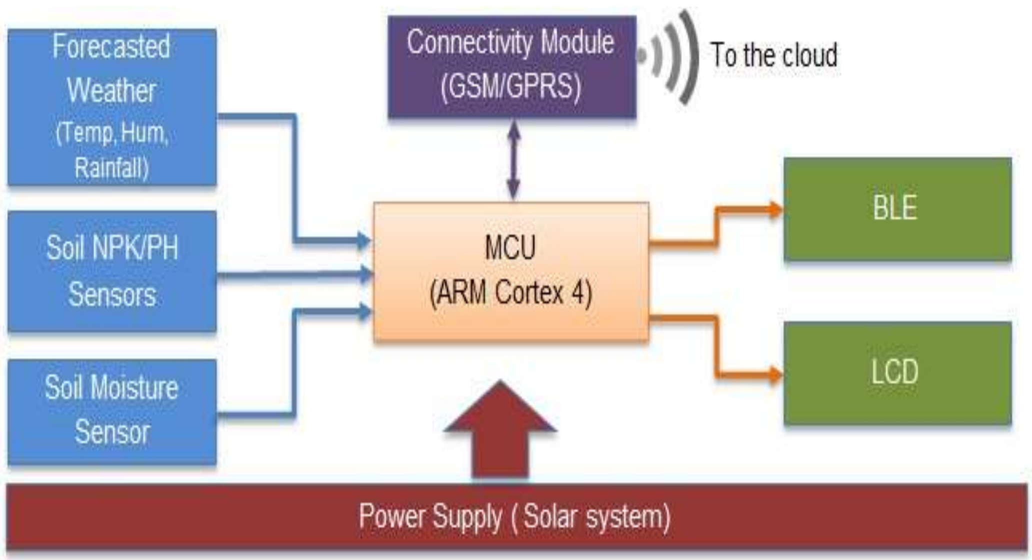


Figure 4.2: Sink node block diagram

4.2 Hardware Components

The proposed design constitutes the following hardware components; Soil NPK Sensor - the RS485 soil nutrient fertilizer detector meter which is suitable for detecting the content of nitrogen, phosphorus, and potassium in the soil. Soil PH Sensor- the RS485 Arduino Soil pH Sensor for agriculture was used. The sensor is widely used and reliable in soil PH testing and other occasions that need pH monitoring. An Arduino Nano 33BLE sense that has an ARM

Cortex 4 processor which supports TinyML models and a GSM module for sending an SMS to the user's mobile phone. It consists of an LCD for the display of results on the device and a MAX485 TTL to RS-485 Interface Module for interfacing with the two sensors NPK and soil pH sensor.

4.2.1 Arduino Nano BLE Sense

The Arduino Nano 33 BLE sense is built upon the nRF52840 microcontroller. It has the ARM Cortex 4 processor making it appropriate for solutions that involve embedded machine learning. In addition, it has an integrated Bluetooth low energy module that enables communication with other devices and a variety of sensors including temperature and humidity sensors. Figure 4.3 shows the pin layout for the board. It has been designed to offer a power savvy and cost effective solution.

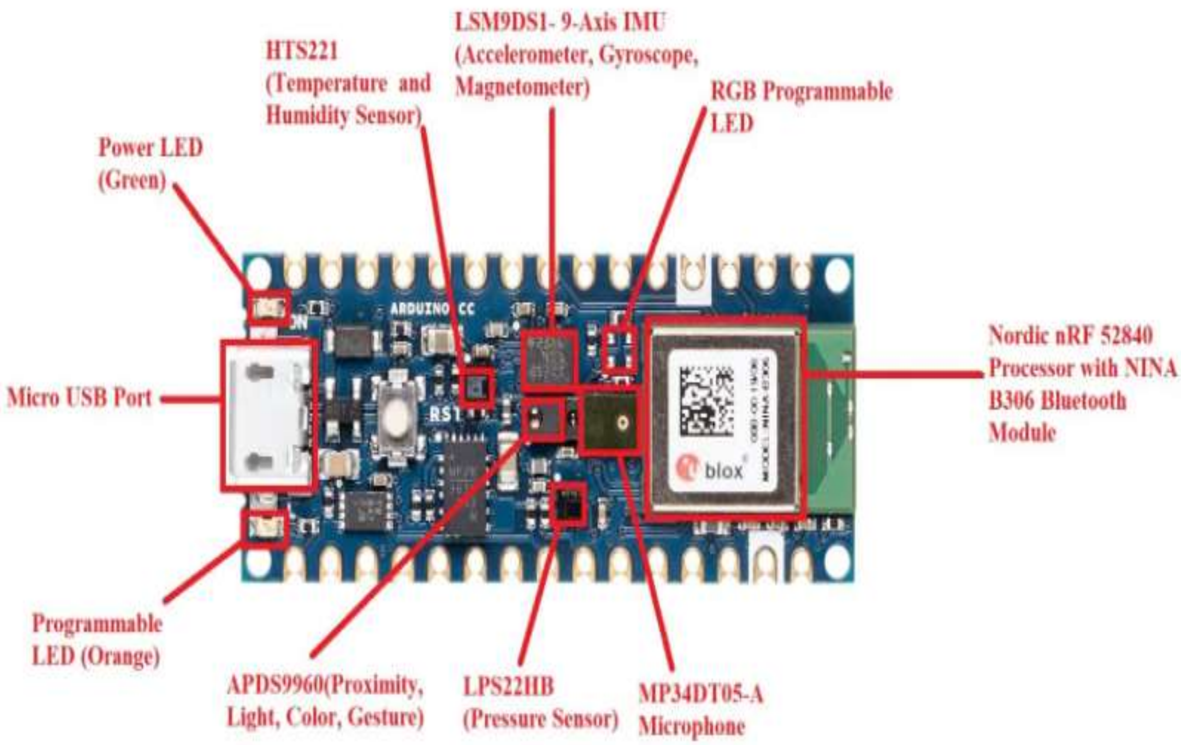


Figure 4.3: Arduino Nano 33BLE Sense

This compact and reliable Nano board is built around the NINA B306 module for BLE and Bluetooth 5 communication; the module is based on Nordic nRF 52840 processor that contains a powerful Cortex M4F and the board has a rich set of sensors that allow the creation of innovative and highly interactive designs. Its architecture, fully compatible with Arduino IDE Online and Offline.

4.2.2 Soil NPK Sensor

This is suitable for detecting the content of Nitrogen (N), Phosphorus (P), and Potassium (K) in the soil. The JCXT soil NPK sensor that was used for this system is a low cost, quick responsive, high precision and portable Sensor that works with Modbus RS485. The advantage of this sensor over a traditional detection method is that it gives very fast measurement and data are highly accurate. It has a high-quality probe**,** rust resistance**,** electrolytic resistance, and alkali corrosion resistance to ensure the long-term operation of the probe part and is thus suitable for all kinds of soil.

Figure 4.4: The NPK sensor

It operates on a power range of 9V-24V and has a measuring range of 0-1999 mg/kg. The Operating temperature is 5-45 °C, its resolution is1mg/kg, and precision of ±2% with a baud rate of 2400/4800/9600. The principle of an optical NPK sensor is based on the interaction between incident light and soil surface properties, such that the characteristics of the reflected light vary due to the soil's physical and chemical properties.

The Pin layout of the connections with the Modbus RS485 module

The NPK Sensor has 4 wires. The brown one is VCC which needs a 9V-24V Power Supply. The GND pin is black in colour and is connected to the GND of Arduino. The Blue wire which is the B pin is connected to the B pin of MAX485 and the Yellow Wire which is the A pin is connected to the A pin of MAX485.

4.2.3 Soil pH Sensor

This is a Soil pH Sensor that can measure the Soil pH value from 3 to 9 with high accuracy up to ±0.3PH. It is water and dustproof and has an IP68 protective case and is sealed with High-density epoxy resin which can prevent moisture from entering the body interior part. The sensor works perfectly with Modbus RS485 and the result is highly impressive [31]

Figure 4.5: The soil pH sensor

Pin lay out connection

The soil pH sensor uses software serial Modbus as a communication protocol to interface with the Arduino board. VCC is the brown wire that needs a power supply between 5V-30V thereafter connects to the 5V of Arduino. The GND pin which is black in colour needs to be connected to the GND of Arduino. The Blue wire which is the B pin is connected to the B pin of MAX485 and the Yellow Wire, which is the A pin, is connected to the A pin of MAX485.

Figure 4.6: Pin connection of soil pH sensor to the interface module

4.2.4 MAX485 TTL to RS-485 Interface Module

This interfaces both the soil NPK and soil pH sensors with Arduino board. It allows us to use the RS-485 differential signalling for robust long-distance serial communications up to 1200 meters or in electrically noisy environments and is commonly used in industrial environments. It supports up to 2.5MBit/Sec data rates, but as distance goes up, the maximum data rate that can be supported comes down.

Figure 4.7:MAX485 TTL to RS-485 Interface Module

Pin lay out of the module

There are 4-pin headers on the assembly module namely;

1 x 4 Header (Data side)

RO = Receiver Output. Connects to a serial RX pin on the microcontroller

RE = Receiver Enable. This is for Active LOW and Connects to a digital output pin on a microcontroller.

DE = Driver Enable. This is for Active HIGH and it is typically jumpered to RE Pin.

DI = Driver Input. This Connects to serial TX (Transmission) pin on the microcontroller

1 x 4 Header (Output side)

VCC = 5V

B = Data ‘B’ Inverted Line. Common with the B

A = Data ‘A’ Non-Inverted Line. Connects to A on far end module

GND = Ground

4.2.5 Liquid Crystal Display (LCD)

LCD: provides visual status information about the system status on the device. When the sensors collect data from the soil, the N, P, and K values are displayed plus the soil pH value too on the LCD. It also eventually displays the crop predicted after data aggregation and inference are done.

The LCD that was used has 8 data pins. An LCD screen is an electronic display module that uses liquid crystals to produce a visible image. The 16×2 translates to a display of 16 characters per line in 2 such lines.

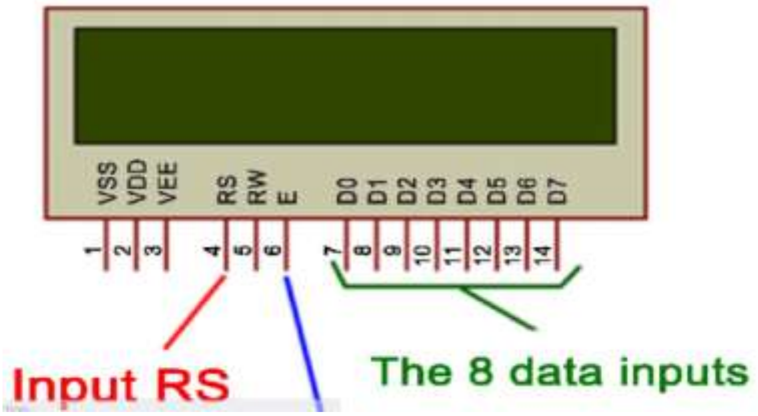


Figure 4.8: The 16x2 Liquid Crystal Display

4.2.6 The GSM Module

A GSM SIM800L modem was used as a communication module to enable the farmer to receive the prediction information on their mobile phone through an SMS. GSM is a wireless modem that works with a GSM wireless network. This modem sends& receives data through radio waves. It requires a SIM card from a wireless carrier in order to operate a GSM module[32]

This has an on-board LED indicator which blinks once every two to three seconds when it has completely registered the SIM to a network. When the LED indicator is blinking every second, this means that the SIM800L is still searching for a network to register onto. If the LED indicator does not blink, recheck the power supply to ensure that it provides plentiful current and precise output voltage. The SIM800L module requires voltage in range of 3.4 to 4.4 V. If proper voltage is not provided, the module will give under- and overvoltage warnings.

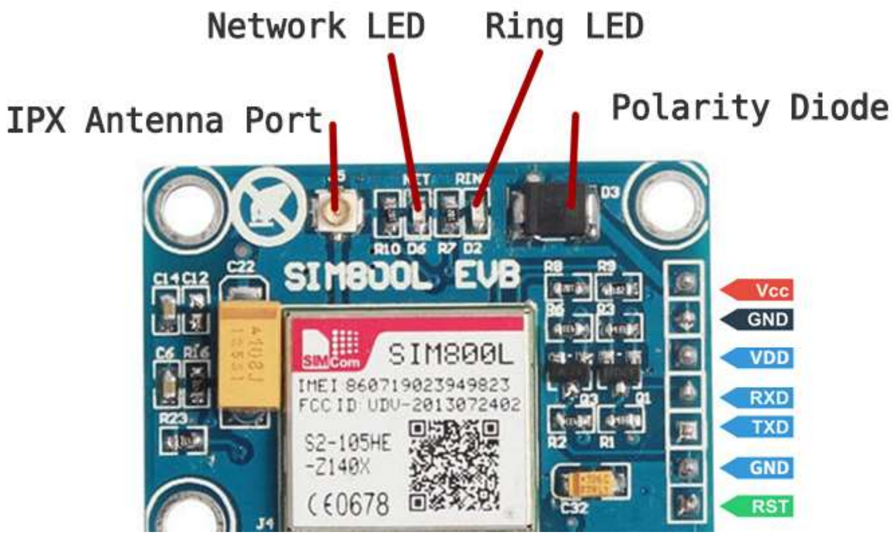


Figure 4.9: The SIM800L GSM Module

4.3 System Analysis

4.3.1 System functional requirements

The end goal of a project is to deliver a high quality product. Functional requirements are the primary ways that the requirements are communicated to a project team. Functional requirements help to keep the project team going in the right direction. They include product features or functions that developers must implement to enable users to accomplish their tasks. The functional requirements for the proposed system were as follows:

- The system should be able to sense the existing soil parameters in a farm namely; the nutrient (N, P, and K) and soil pH.
- The system should be able to collect the forecasted weather (Temperature, Humidity and Rainfall).
- The system should be able to predict the best crop to grow with the current soil conditions.
- The system should be able to send an alert to a farmer on the predicted crop to be grown on the device and farmer's mobile
- The system should be able to send the collected data and the prediction to an open source IoT cloud platform for storage.
- The farmers and extension officers should be able to view a dashboard with a variety of visualization tools of data stored in the cloud.

- The system should be powered through a renewable energy source and use the minimal energy possible.

4.3.2 Non-functional requirements

Non-functional requirements are quality attributes that describe the ways the system should behave. They include the following:

- Availability: the system's functionality and services should be available for use with all operations 99.99% of the time.
- Usability: the system should be easy to use by the farmers, the extension officers, and the administrator.
- Reliability: The system should work without failure for at least 10 years
- Scalability: The system must grow without negative influence on its performance.
- Power consumption: It should be in a position to consume as low power as possible to conserve energy and the environment. Ultra-low power devices should be used in the implementation.
- Data Integrity: the system should be in a position to secure access to confidential data for the users.
- Performance: the system should ensure optimal responsiveness to various user interactions with it at all times
- Recoverability: In case of failure, the system should have a self-recovery backup procedure
- Flexibility: Flexible service based architecture will be highly desirable for future extension
- Security: ensure that the software is protected from unauthorized access to the system and its stored data.
- Size: the system should be designed using miniaturized devices for portability
- Regulatory requirements: the system should confirm the traffic regulatory requirements

4.4 The Flow chart for data collection

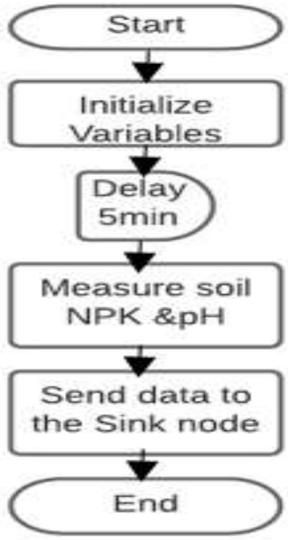


Figure 4.10: How the data collection process flows

4.5 The system flow chart

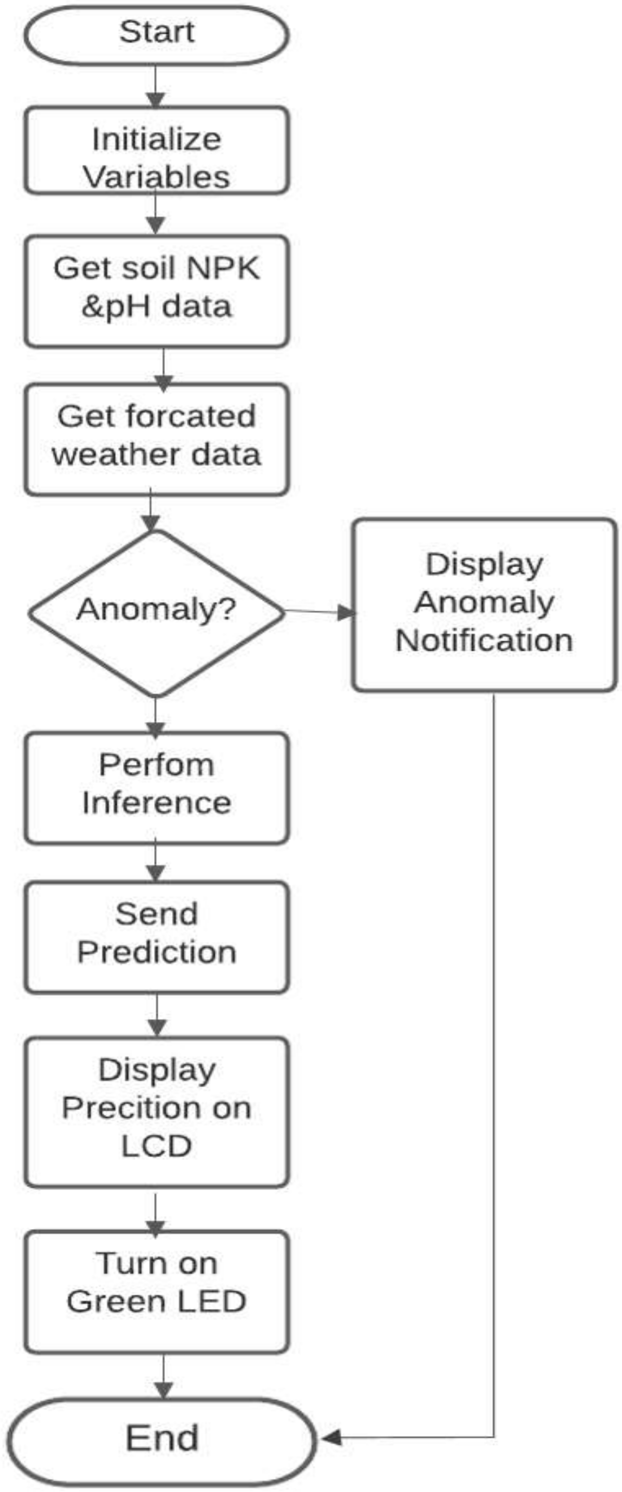


Figure 4.11: The flow of the system process

4.4 Simulation design of the Embedded Kit

Figure 4.11 presents the high-level simulation context of the proposed system. The Embedded AI model executable is deployed in an STM32F401CC board on proteus design suite. Input data of readings from sensors and weather information is given in the form of a file from an SD card. This data includes values of Nitrogen, Phosphorus, and Potassium (NPK), values of soil pH, and the values of weather in terms of rainfall, temperature, and humidity. The inference results being shown on the serial terminal or also on the Liquid Crystal Display (LCD)

The STM32F401xC devices are based on the high-performance Arm® Cortex® -M4 32-bit RISC core operating at a frequency of up to 84MHz. These can support the TinyML model that is being used for inference and prediction at the edge device.

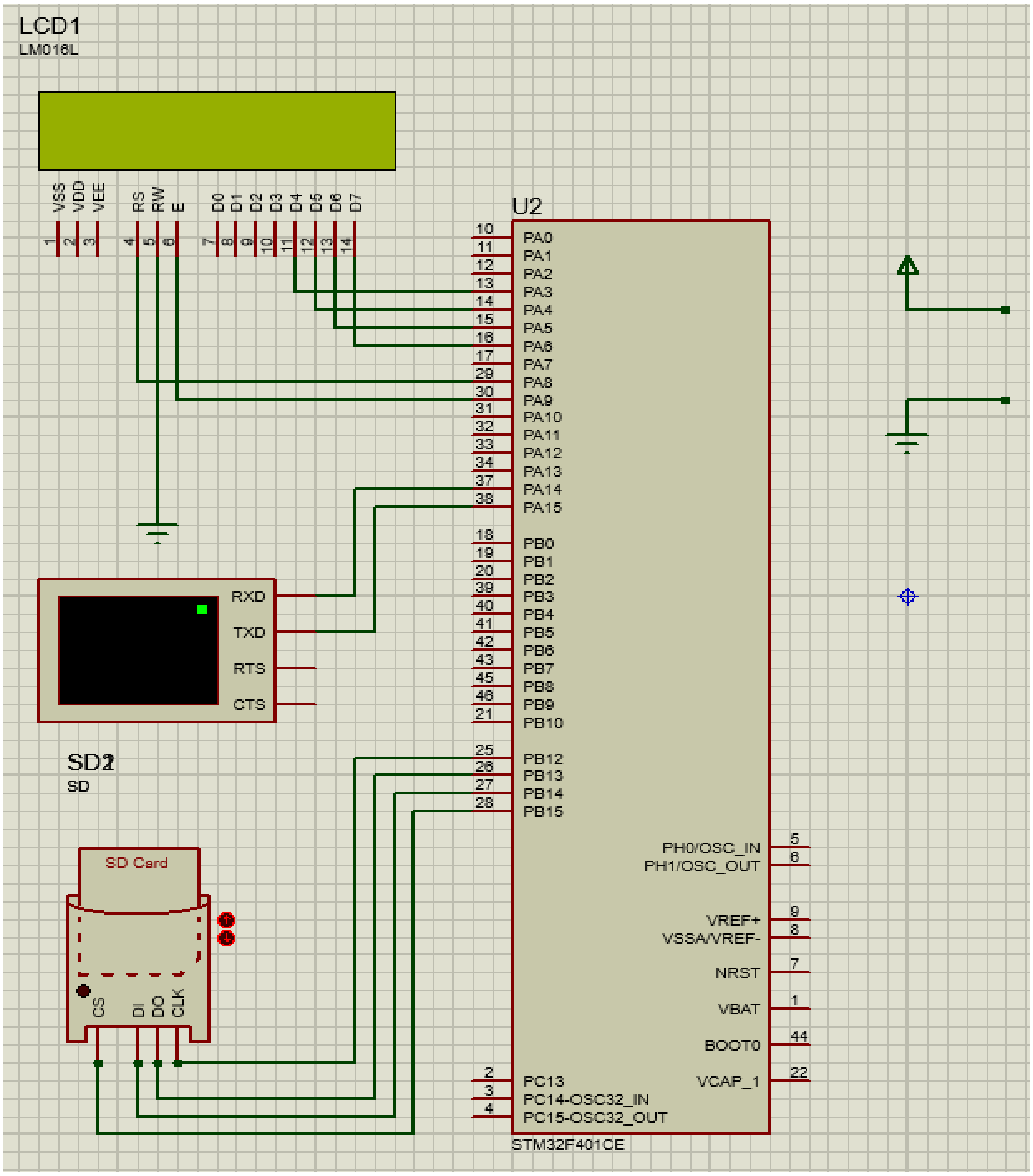


Figure 4.12: Proteus simulation layout of the system

CHAPTER FIVE

RESULTS AND ANALYSIS

This section presents the results of the research study, the analysis of the results and system plus the explanations of the results.

5.1 Evaluation of embedded AI for predicting best crop

From the Open data sets, synthetic data is generated using Mostly AI platform in this case. Then, the synthetically enhanced data is fed into Edge impulse which is a Machine learning platform. A TinyML model is generated and deployed on STM32 Cube with an Edge AI application IDE, from which the model is simulated and inference got on a serial monitor.

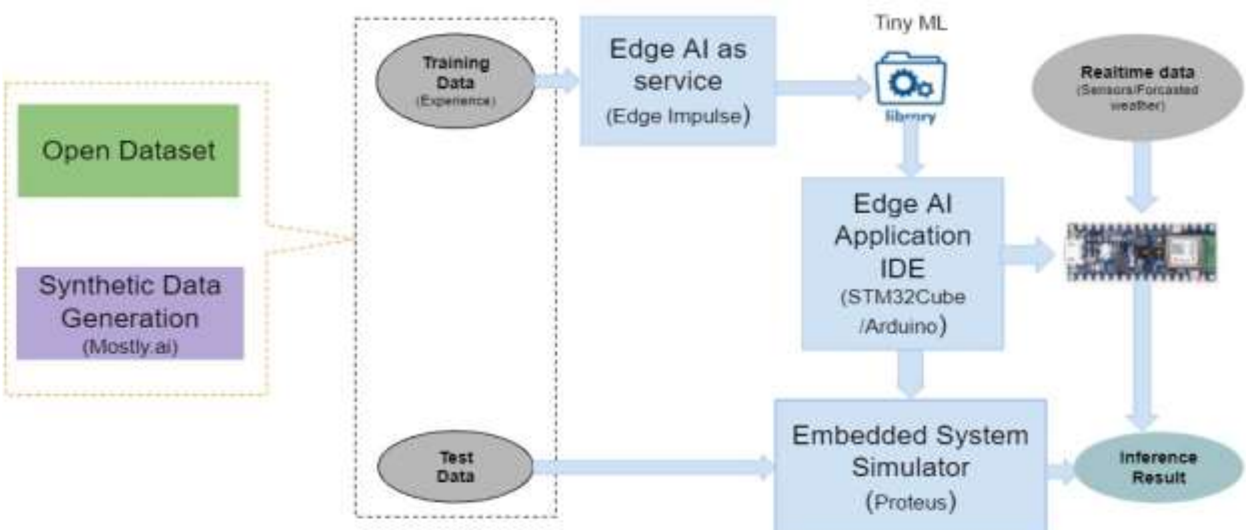


Figure 5.1: Summary of the steps for system development

5.1.1 Input: Synthetic Dataset

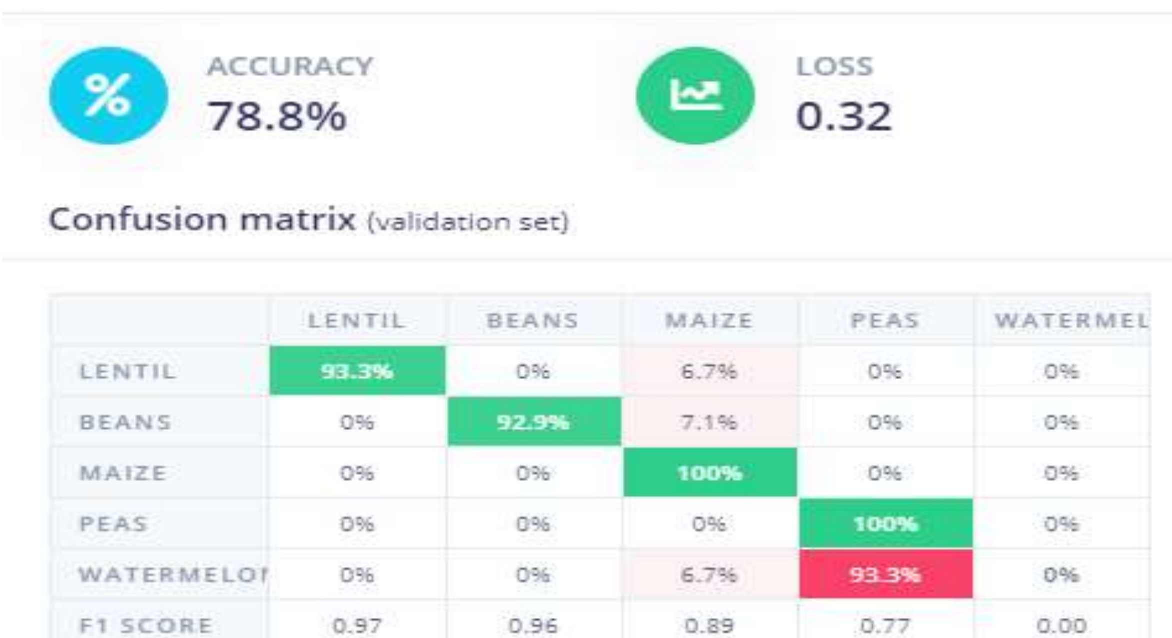
In this study, an open data set collected from a farm found was used as input to the synthetic data generation platform [33]. The reason for using synthetic data is to increase the volume of data to enable a better deep learning model. The free version of a commercial synthetic data platform, Mostly AI, was used to generate more data [27]. The dataset includes information on N, P, K, pH, temperature, humidity, and rainfall collected everyday over some time with labels of the best crops that did well under the specified conditions. The data contained 20 different crops and the best parameters for maximum yields. Data relating to five food crops were selected for our study. Before use, the generated synthetic data was tested and the performance compared to using the original data set and was 99% accurate. Synthetic data generation reports also show that all the required thresholds including privacy tests were met.

5.1.2 Embedded AI model generation

The training model was developed using an open source embedded ML platform called edge impulse. The data was preformatted and JSON files were created for upload into the platform. Five files each with data about the five selected crops were uploaded and data automatically separated into the training, validation, and test sets using the holdout method. The raw data were classified using a Neural Network classifier. The model had 7 inputs, from sensor and weather data and 5 outputs being the selected crops which were Maize, Beans, Lentil, Peas, and watermelon. The window size used was 1000ms with a sampling rate of 1000ms for the data.

5.1.3 Prediction Model with non-synthetic datasets

With the use of small volumes of data that were raw from the open source, a model accuracy of 78.8% was achieved with a loss of 0.32% from the validation set. This implies that there is a 78.8 percent probability of predicting the best crop rightly



	LENTIL	BEANS	MAIZE	PEAS	WATERMEL
LENTIL	93.3%	0%	6.7%	0%	0%
BEANS	0%	92.9%	7.1%	0%	0%
MAIZE	0%	0%	100%	0%	0%
PEAS	0%	0%	0%	100%	0%
WATERMELOI	0%	0%	6.7%	93.3%	0%
F1 SCORE	0.97	0.96	0.89	0.77	0.00

Figure 5.2: Model confusion matrix with non-synthetic data

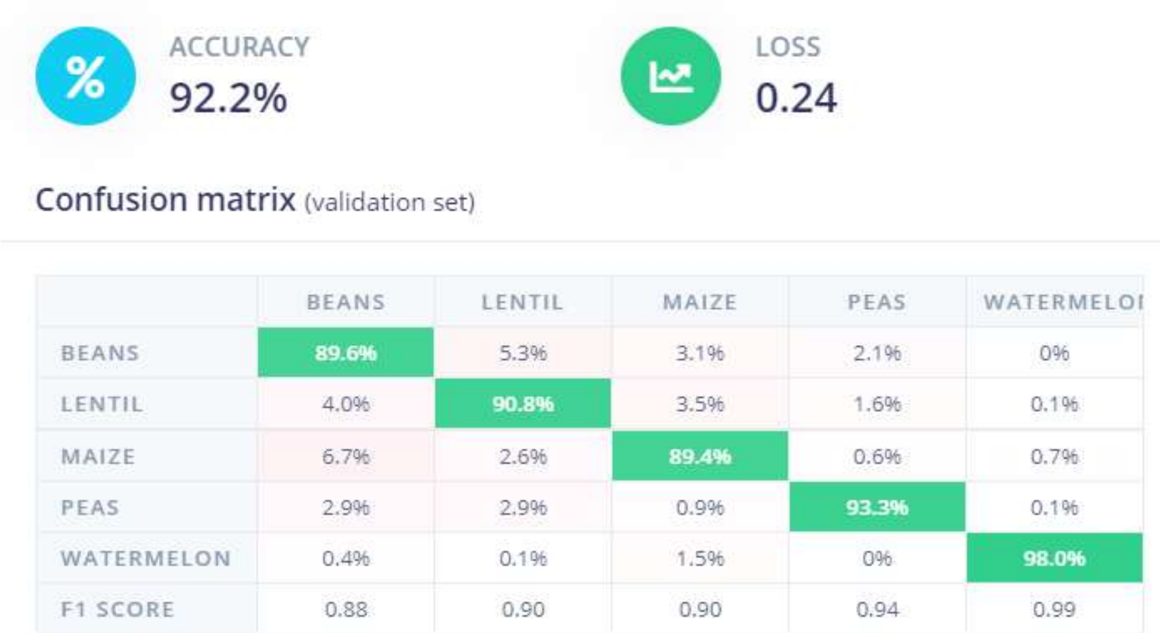
With this model, there is a probability of 0.97 of lentil being right, 0.96 probability of beans being right, 0.89 probability of maize being right, 0.77 probability of peas being right and 0.0 probability of watermelon being right as shown in the F1 score row. This is because of the limited dataset samples that the model is being based on for training. Figure 5.2 shows the confusion matrix for the validation set.

Figure 5.3: On-device performance for non-synthetic data

With non-synthetic data, the on-device performance gave an inference speed of 1ms, it utilizes a RAM amount of 1.5Kb and a flash memory of 15.8Kb. This implies that our model can well fit and function on embedded devices that have limited memory space.

5.1.4 Prediction Model with Synthetic datasets

With the use of large volumes of data that were synthetically generated, a model accuracy of 92.2% was achieved with a loss of 0.24% from the validation set. From the confusion matrix below, the probability of predicting beans accurately is 0.88, the probability of predicting lentil accurately is 0.90, the probability of predicting maize accurately is 0.90, the probability of predicting peas accurately is 0.94 and the probability of predicting watermelon accurately is 0.99. This shows an increase in accuracy of the model compared to when raw small datasets are used. Figure 5.4 shows the confusion matrix for the validation set.



	BEANS	LENTIL	MAIZE	PEAS	WATERMELOI
BEANS	89.6%	5.3%	3.1%	2.1%	0%
LENTIL	4.0%	90.8%	3.5%	1.6%	0.1%
MAIZE	6.7%	2.6%	89.4%	0.6%	0.7%
PEAS	2.9%	2.9%	0.9%	93.3%	0.1%
WATERMELON	0.4%	0.1%	1.5%	0%	98.0%
F1 SCORE	0.88	0.90	0.90	0.94	0.99

Figure 5.4: Model confusion matrix with synthetic data

This implies that with large volumes of data, a better model inference accuracy can be achieved and this is through the use of the small volumes of data available to generate synthetic data with bigger volumes but maintaining the statistical distribution of the data. Synthetic data use in the generation of datasets for AI models assures the privacy of the data and this is key in the data science world. It also saves costs, it's more accurate and faster.

Figure 5.5: On-device performance for synthetic data

With the use of synthetic data, the on-device performance achieved is shown above. The inference took 1ms, the amount of RAM used is 1.7Kb and the flash memory used is 17.4Kb. These can all be accommodated on edge devices with constrained memory and computation capacity.

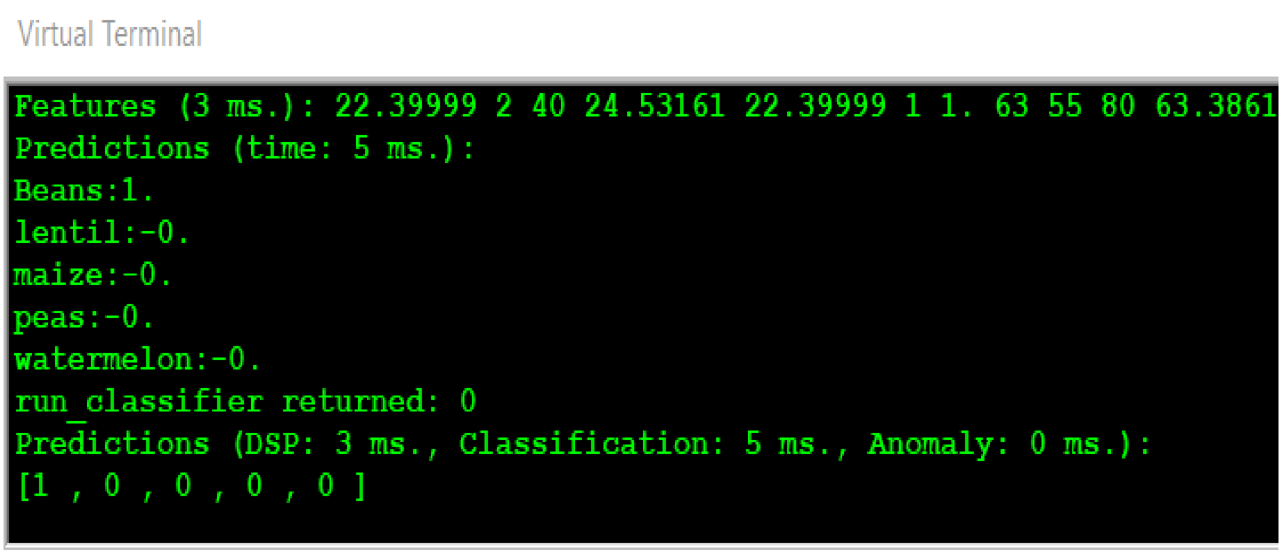
K means anomaly detection learning block was added to enable the model to identify any anomalies that may have been captured by the sensors. The cluster count was set at 32 with a minimum score before tagging an anomaly of 0.30. After testing a CMSIS-PACK for STM32 boards was generated for integration in an IDE and deployment

5.1.4 Model Validation

To validate the model, test data from the real original open datasets were used to find out how accurately the best crop to be grown can be predicted. When the test data was applied on both the cloud and embedded device, the model predicted the crop to be grown with 99.9% accuracy. This shows that the model is effective in predicting the best crop to be grown considering the real-time conditions of the soil. In addition, this confirms that synthetically enhanced data has minimal effects on the performance of the resulting models. The synthetically enhanced data actually proved to give a more concrete model as compared to when small datasets were used.

5.1.5 Inference Simulation

A C++ project was created in STM32CubeIDE and the CMSIS-PACK was integrated into the project. The project was then compiled and debugged and an executable HEX file was created for simulation on proteus platform. Figure 5.6 gives a sample output from the simulation on the Proteus design suite. This is the same result as when compared to classification in the cloud using the same data as shown in figure 5.7.



```
Virtual Terminal
Features (3 ms.): 22.39999 2 40 24.53161 22.39999 1 1. 63 55 80 63.3861
Predictions (time: 5 ms.):
Beans:1.
lentil:-0.
maize:-0.
peas:-0.
watermelon:-0.
run_classifier returned: 0
Predictions (DSP: 3 ms., Classification: 5 ms., Anomaly: 0 ms.):
[1 , 0 , 0 , 0 , 0 ]
```

Figure 5.6: proteus inference result

From the virtual terminal display result, the crop with value 1 indicates the predicted crop to be planted based on the real time condition of the soil, then value 0 shows the crops not suitable for the soil in that state at the moment. The classification of the data took 5ms and the digital signal processing (DSP) of the data took 3ms. The model performs anomaly detection on the data to rule out the ambiguity of results.

When a model was trained using a similar dataset and deployed at the cloud to test if the inference would be the same as the inference at the edge device, the result was satisfactory that the embedded AI result is still accurate. It implies that the embedded Machine Learning model is perfect and accurate and gives the same result as would be done in the cloud. This is the result of cloud analysis of the parameters for crop prediction.

Detailed result

Show only unknowns

TIMESTAMP	BEANS	LENTIL	MAIZE	PEAS	WATERMELON
0	1.00	0	0	0	0

Figure 5.7: Cloud inference output

The results give a prediction of the best crop to be grown indicated by value 1 without the use of fertilizers as the conditions on the validation dataset are appropriate for the recommend crop. This supports the objectives of the study on limiting the use of fertilizers and thus environmental conservation.

5.2 PROTOTYPE RESULT ANALYSIS

5.2.1 Prototype Implementation

Different components of the embedded device were connected as shown in figure 5.8. The prototype was set up in the STES group Lab and with sample soils being collected from STES group farm where they practice smart irrigation. Given the fact that their system did not monitor the soil NPK and PH as done in the study, the solution was also integrated into their system so as to evaluate its ability to improve the existing system.

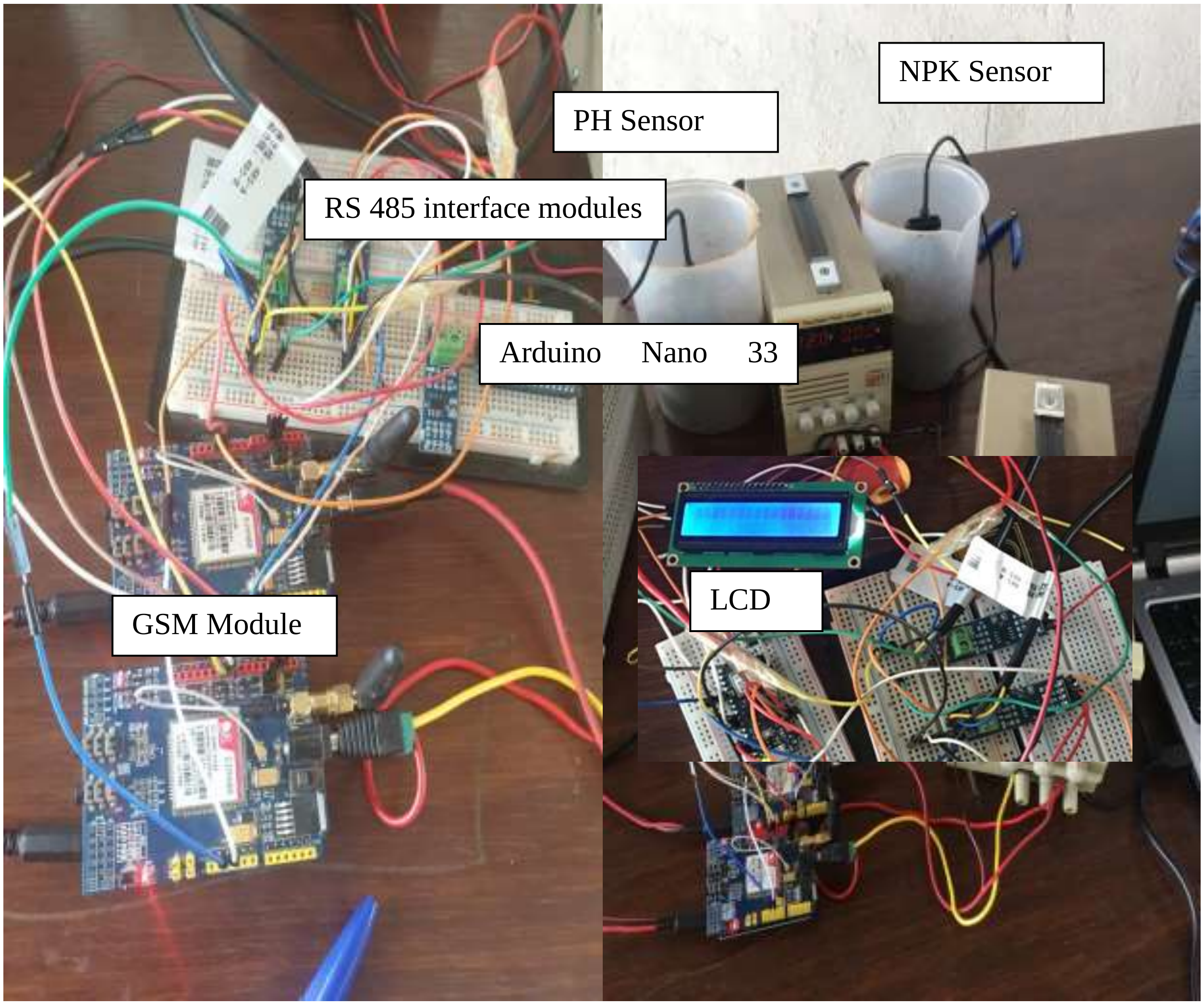


Figure 5.8: The System prototype

5.2.2 Sensor Readings

Figures 5.9 shows the readings from the sensor on a serial monitor during the testing of their functionalities. Different soil samples were taken and the sensors adequately gave different readings for each sample.

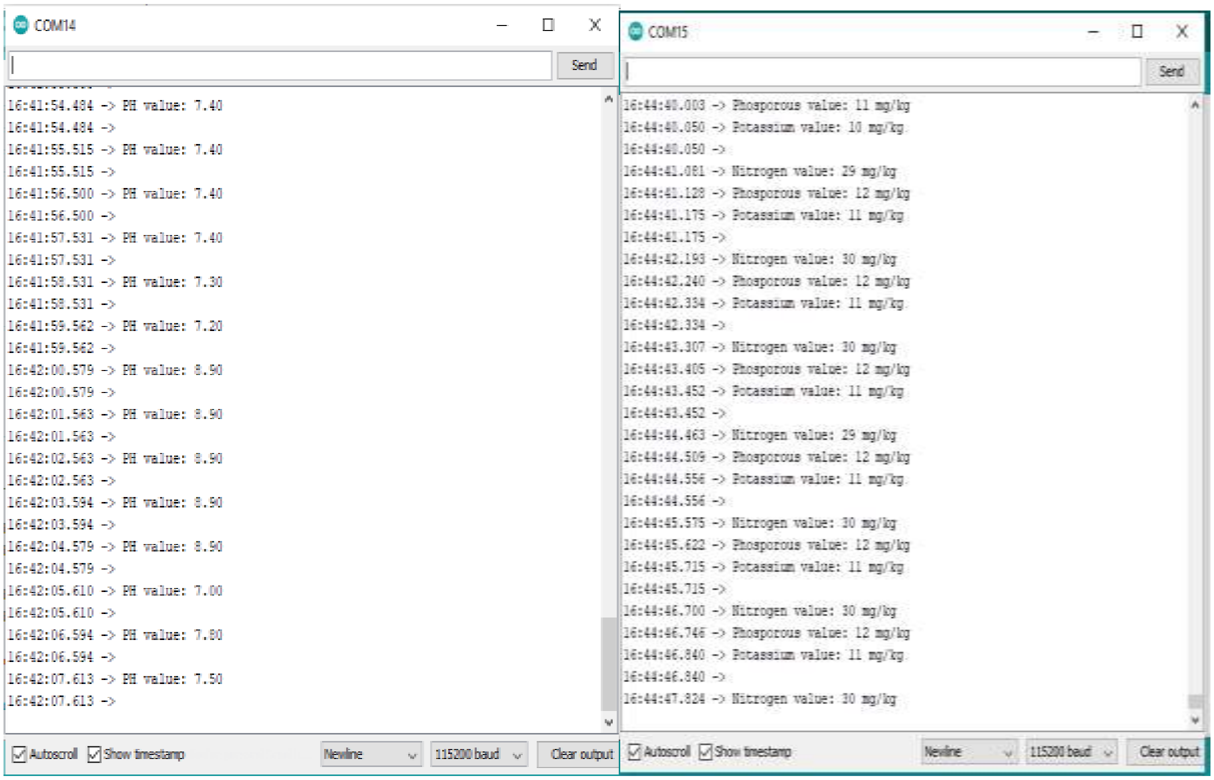


Figure 5.9: Sample sensor readings

5.2.3 Cloud Storage

A private cloud storage platform used by STES group was used for storage of the soil condition readings overtime. Figure 5.10 shows the dashboard for the cloud storage platform. Separate tables were created for each of the targeted sample values and data collected over a period of 5 days. Figure 5.11 shows sample collected data for the sensors. Data was sent to the virtual cloud via GSM modules after every 5 minutes. The database created and hosted on Bazafarm virtual cloud stores the sensor readings of Nitrogen, Phosphorus, Potassium and soil pH plus their time stamps. The reason for storage of this data is for analysis, future use and research because in this work, we faced a big challenge of lack of enough datasets for concrete AI model training.

The dashboard displays the number of devices connected, the number of collected data entries, the number of parameters being collected and the time stamps for the data.

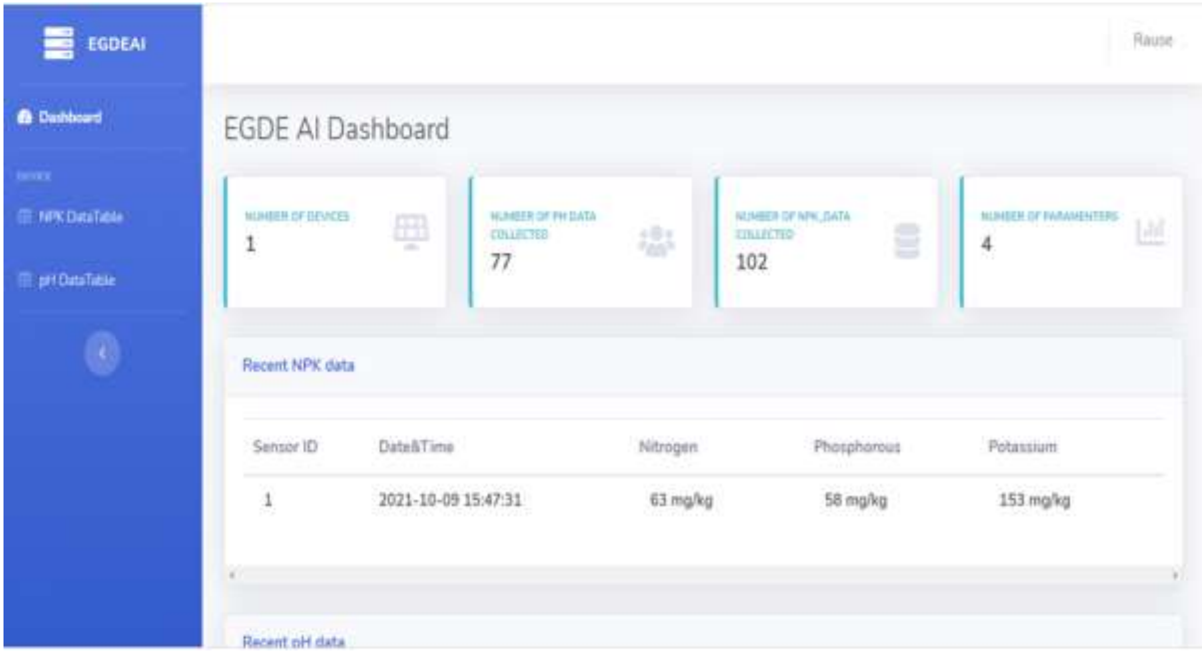


Figure 5.10: The system cloud dashboard

1	83	76	204	2021-10-09 13:11:43
1	83	76	204	2021-10-09 13:17:05
1	83	76	204	2021-10-09 13:22:29
1	83	76	202	2021-10-09 13:27:49
1	83	76	202	2021-10-09 13:33:12
1	82	75	201	2021-10-09 13:38:33
1	79	72	194	2021-10-09 13:43:58
1	79	72	194	2021-10-09 13:49:19
1	79	72	194	2021-10-09 13:54:42
1	79	72	194	2021-10-09 14:00:05
1	78	72	192	2021-10-09 14:05:26

Figure 5.11: Sample NPK sensor collected data

5.2.4 Crop Prediction

The system was able to predict the best crop to be grown based on the soil NPK, pH and forecasted weather data pulled by an API integrated in the application. Figure 5.12 gives a sample prediction for watermelon at 0.99. The prediction is given in terms of a probability of the crop that would do best under the existing soil conditions.

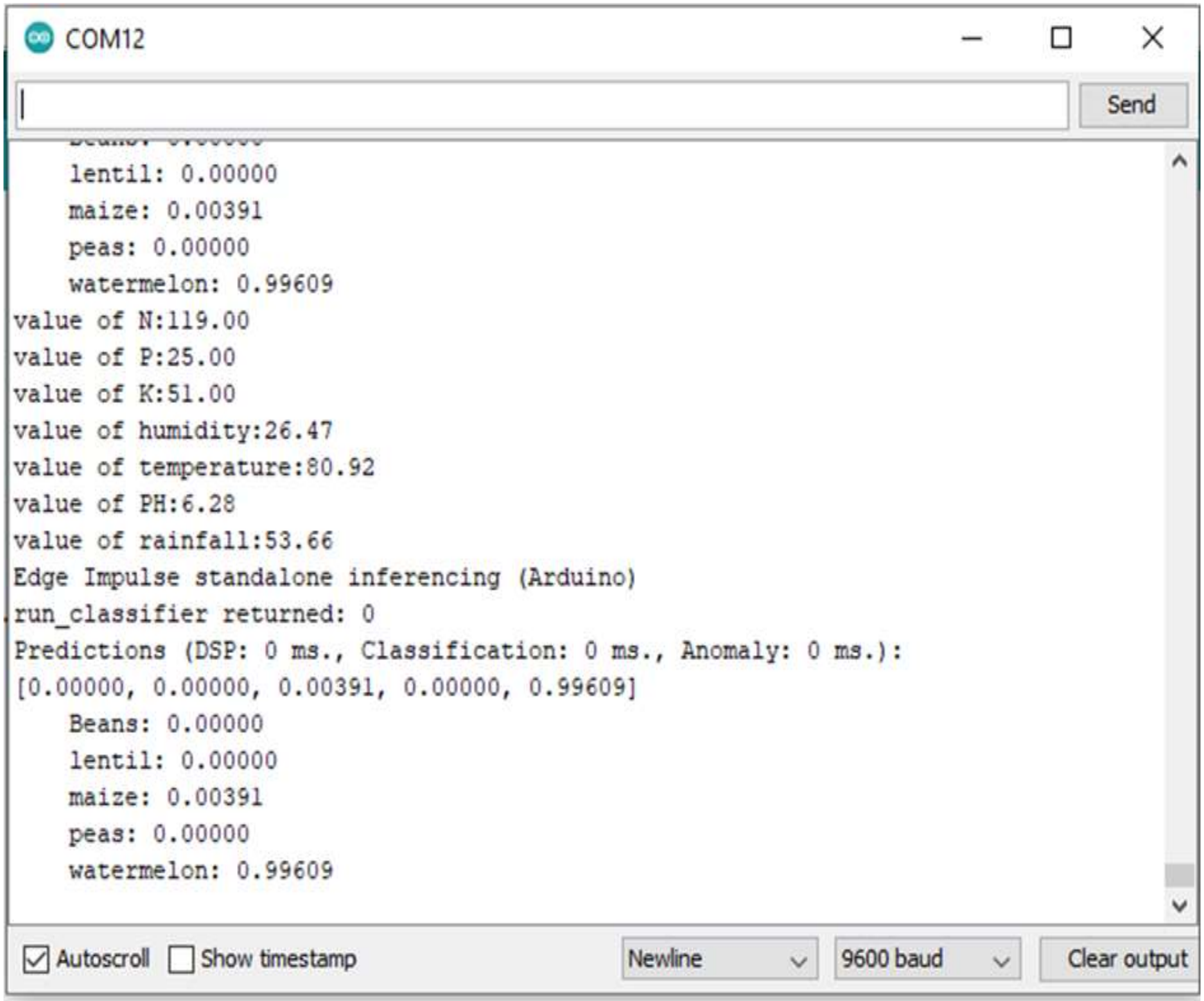


Figure 5.12: crop prediction results

The probability of beans growing well in the sampled soil without having to complement the soil with NPK fertilisers is 0.00, for Lentil it is 0.00, for maize it is 0.00391 and for peas its 0.00. With this prediction, in case the farmer chooses to plant otherwise, He will need to apply NPK fertilizers to support his crop planted

The figure 5.12 shows the inference display on LCD that is attached to the embedded device

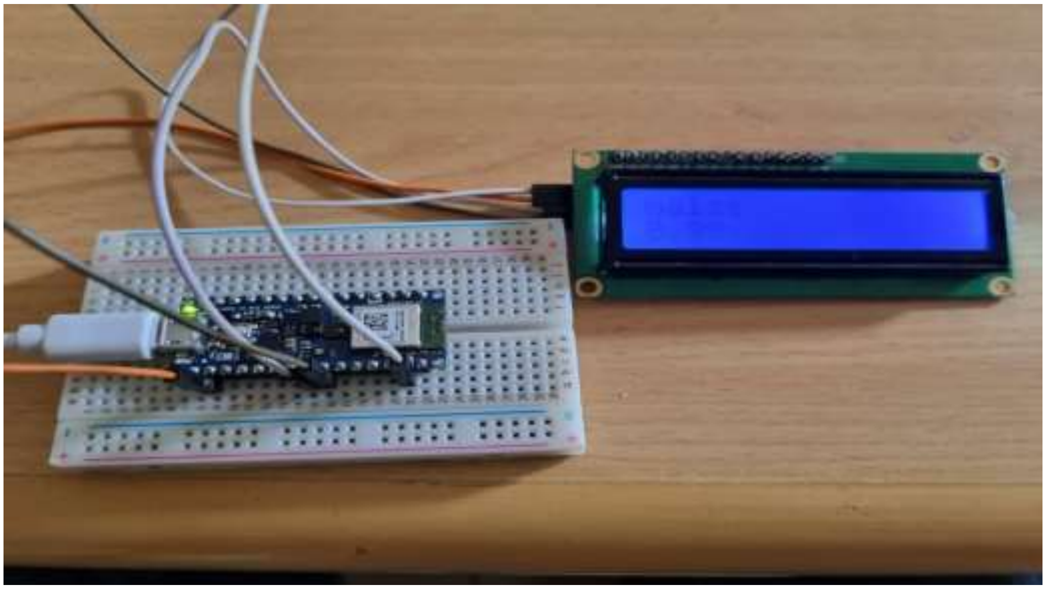


Figure 5.13: Inference display on LCD

5.3. Analysis and interpretation of results

5.3.1. Device Performance

The required device resources by the model were analysed to determine if the model could run on an embedded device as was intended. The estimated on-device performance by the model on an embedded device from the cloud training platform is 1.7Kb peak RAM usage, 17.4Kb ROM usage, and an inference time of 1ms. The results show that the required resources are still minimal. This shows that the model can be used on many commercially available embedded devices that have the required ARM Cortex M4 core.

5.3.2. Model parameter Significance

Different parameters were omitted during the model training to verify the significance of each parameter in predicting the best crop to be grown. Figure 5.14 performs each of the three classes of parameters used namely soil nutrient, Soil PH, and weather (temperature, humidity, and rainfall). This was done using the same settings for the neural network. The graph below shows how relevant each parameter is in providing a concrete prediction inference for the system. NPK is most relevant with a percentage of 78.5% followed by soil humidity or moisture with 68% that comes under the forecasted weather APK. These are followed by rainfall and pH with 52%.

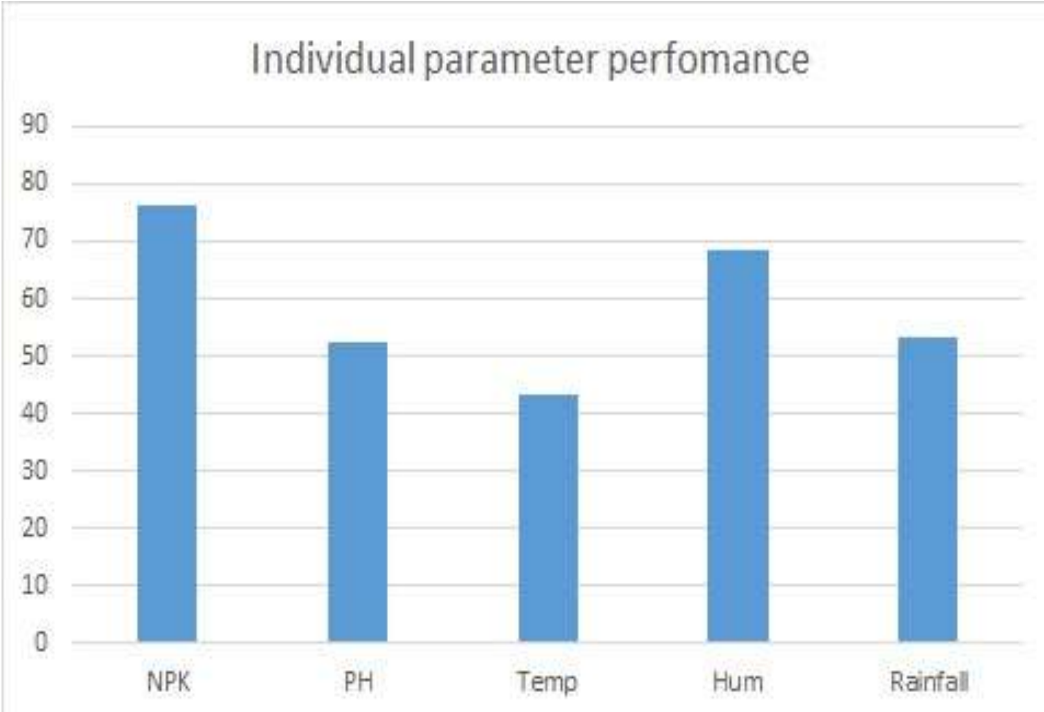


Figure 5.14: Individual sensor significance

5.3.3 Sensor Significance

In this experiment, sensors were used and an APK (Android Application Package) for forecasted weather data. The graph below in figure 5.15 shows the significance of each sensor and weather APK in the prediction analysis. From the graph it shows that weather APK and soil NPK sensors are the major considerations that are mandatory for predicting the best crop to grow as compared to pH. The soil pH is dependent on the soil nutrient content hence the pH sensor can be replaced by the soil NPK sensor. In addition to that, there exists a soil integrated sensor that measures seven (7) parameters in one that is; soil Nitrogen, Phosphorus and Potassium content (NPK), soil pH, soil conductivity, soil moisture and soil temperature. Using this can help optimise resources at a reduced cost. Looking at the resulting graph, the weather APK has a significance of 95% followed by the soil NPK sensor with a significance of 80% and lastly the soil pH sensor with a significance of 55%.

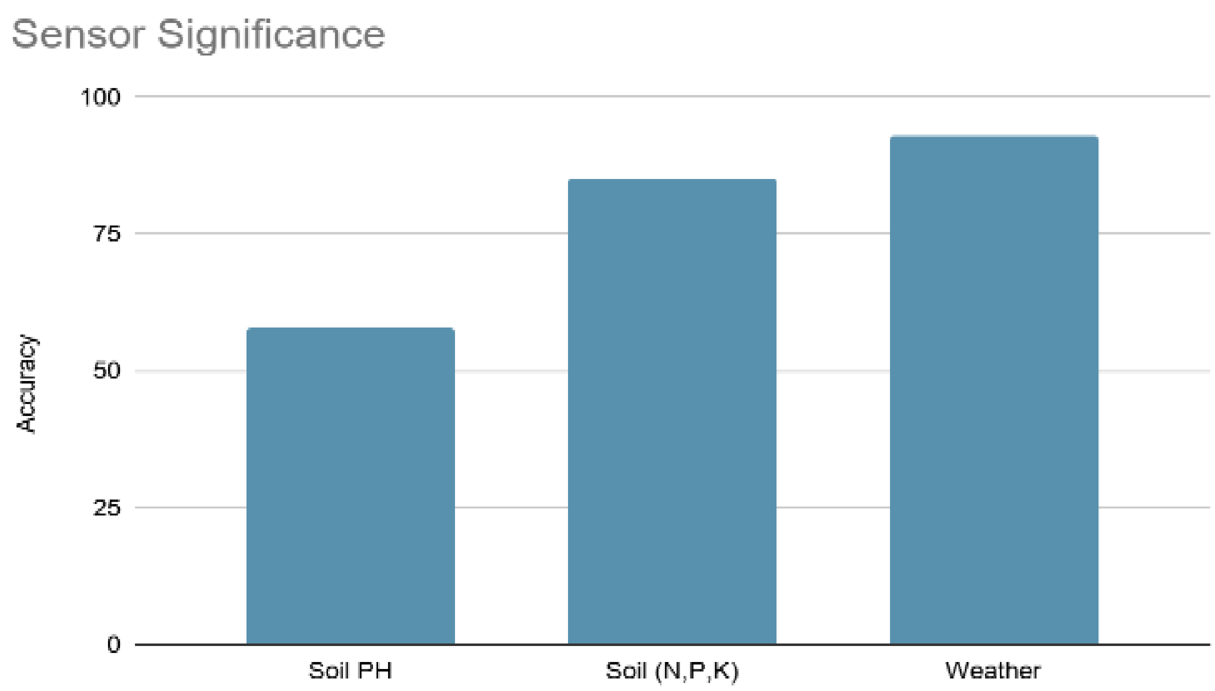


Figure 5.15: Individual sensor significance

5.3.5 Effect of Soil Moisture on soil NPK and PH

During the testing of the working of the prototype it was noted that soil moisture affects the values for both NPK and pH. This means that it is important to also monitor the soil moisture so as to get a more accurate prediction.

When soils were dry, the pH values tended to hike beyond 7 and the NPK values also tended to lower down. This is one of the reasons why rainfall is very key in crop growth. It moisturizes the soil and helps the NPK values to increase and stabilise. When the soil is too acidic, lower than the recommended level of 5.5 for good crop yield, there will be a decrease in the crop yield instead.

CHAPTER SIX

CONCLUSION, RECOMMENDATIONS, AND FUTURE WORKS

6.1 CONCLUSIONS

This study proposes the use of Embedded AI in precision agriculture for the prediction of the best crop to grow with the existing soil conditions to conserve the environment. This is a move from the existing solutions that mostly use cloud-based solutions. The use of Embedded AI helps overcome connectivity challenges in Africa and ensures real-time responses for precision solutions.

Our model was tested in both the cloud and embedded devices with the results giving the same accuracy. This supports the use of Embedded ML for precision farming solutions and is scalable to other use cases so long as the data for model training is available. The study also shows that synthetic data can also be applied in smart agriculture in cases of limited data for machine learning. Our experiment shows that the use of synthetic data does not degrade the performance of an AI model so long as the right methods are applied. From the evaluation of the sensors, we note that soil nutrient and weather information are vital when deciding on which crop to the best plant. Since pH is related to the underlying soil nutrient levels its effect on the model performance is minimal.

Considering that this is ongoing research, the next step is to implement this solution, which will lead to the conservation of the environment by ensuring that farmers minimize the use of fertilizers that have a lasting effect on the environment. The use of embedded AI will also ensure that costs are reduced and real-time actions are taken to enhance productivity.

6.1.1 NULYFYING THE HYPOTHESIS

The hypothesis was that IoT technologies can be merged with Artificial Intelligence to enhance environmental conservation through sustainable soil management and crop prediction.

From our results, it is proven that through merging of Artificial Intelligence and embedded Machine Learning in particular with Internet of things technology, a concrete crop prediction can be made on a fertilizer free basis. When the crop is grown with the current soil contents based on the prediction, fertilisers won't have to be added and thus conserving the environment through avoidance of the effects of the excessive use of synthetic fertilizers.

6.2 RECOMMENDATIONS

Based on the findings and experiences during the study we would like to recommend the following for better results for such a solution:

- Soil moisture should be part of the inputs to the AI model as it directly influences the readings for the soil quality parameters.

A soil integrated sensor suitable for reading values of soil moisture, soil pH, Nitrogen, Potassium, Phosphorus (NPK), soil temperature and total soil salt (soil conductivity) can be used instead of the 2 sensors that were used in this research. This sensor is soil comprehensive and creatively measures the 7 parameters which greatly optimises resources and facilitates a concrete soil assessment with one sensor.

6.3 PERSPECTIVE AND FUTURE WORKS

This work introduced the use of embedded AI in Agriculture and proposes and evaluates a model for the prediction of the best crop to grow under the existing conditions with minimal use of fertilizers. In addition, a prototype on a real embedded development board was developed and conducted tests with soil from different farms to further evaluate the performance of the proposed solution. Future works will involve implementation of the solution and further evaluation on a deployment setting.

REFERENCES

[1] F. F. Nchuchuwe and K. D. Adejuwon, “The Challenges of Agriculture and Rural Development in Africa : The Case of Nigeria,” *Int. J. Acad. Res. Progress. Educ. Dev.*, vol. 1, no. 3, pp. 45–61, 2012.

[2] M. Dholu and K. A. Ghodinde, “Internet of Things (IoT) for Precision Agriculture Application,” *Proc. 2nd Int. Conf. Trends Electron. Informatics, ICOEI 2018*, no. Icoei, pp. 339–342, 2018.

[3] A. Khattab, A. Abdelgawad, and K. Yelmarthi, “Design and implementation of a cloud-based IoT scheme for precision agriculture,” *Proc. Int. Conf. Microelectron. ICM*, vol. 0, no. October 2017, pp. 201–204, 2016.

[4] P. Sanjeevi, S. Prasanna, B. Siva Kumar, G. Gunasekaran, I. Alagiri, and R. Vijay Anand, “Precision agriculture and farming using Internet of Things based on wireless sensor network,” *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 12, pp. 1–14, 2020, doi: 10.1002/ett.3978.

[5] “A. K Sabry, "Synthetic Fertilizers; Role and Hazards," Fertilizer Technology,vol. 1, pp. 110-133, 2015..

[6] “Plant nutrients in the soil.” https://www.dpi.nsw.gov.au/agriculture/soils/soil-testing-and-analysis/plant-nutrients (accessed Sep. 18, 2021).

[7] K. Lova Raju and V. Vijayaraghavan, *IoT Technologies in Agricultural Environment: A Survey*, vol. 113, no. 4. Springer US.

[8] M. Grell *et al.*, “Determining and Predicting Soil Chemistry with a Point-of-Use Sensor Toolkit and Machine Learning Model,” *bioRxiv*, p. 2020.10.08.331371, Oct. 2020.

[9] STES Group Ltd, "Bazafarm Technology Products” [Online] https://www.stesgroup.rw/bazafarm-technology/. (Last accessed Sep. 18, 2021).

[10] “What are the 4Rs | Nutrient Stewardship.” https://nutrientstewardship.org/4rs/ (last accessed Nov. 05, 2021).

[11] A. More, A. Mouria, N. Panchal, and K. Bathe, “Soil Analysis Using Iot,” *SSRN Electron. J.*, Apr. 2019, doi: 10.2139/SSRN.3366756.

[12] S. Heble, A. Kumar, K. V. V. D. Prasad, S. Samirana, P. Rajalakshmi, and U. B. Desai, “A low power IoT network for smart agriculture,” *IEEE World Forum Internet Things, WF-IoT 2018 - Proc.*, vol. 2018-Janua, pp. 609–614, 2018.

[13] M. Masrie, M. S. A. Rosman, R. Sam, and Z. Janin, “Detection of nitrogen, phosphorus, and potassium (NPK) nutrients of soil using optical transducer,” *2017 IEEE Int. Conf. Smart Instrumentation, Meas. Appl. ICSIMA 2017*, vol. 2017-November, pp. 1–4, Mar. 2018.

[14] R. K. Math and N. V. Dharwadkar, “A wireless sensor network based low cost and energy efficient frame work for precision agriculture,” *2017 Int. Conf. Nascent Technol. Eng. ICNTE 2017 - Proc.*, 2017.

[15] A. N. Arvindan and D. Keerthika, “Experimental investigation of remote control via Android smart phone of arduino-based automated irrigation system using moisture sensor,” *2016 3rd Int. Conf. Electr. Energy Syst. ICEES 2016*, pp. 168–175, Jul. 2016.

[16] S. Vaishali, S. Suraj, G. Vignesh, S. Dhivya, and S. Udhayakumar, “Mobile integrated smart irrigation management and monitoring system using IOT,” *Proc. 2017 IEEE Int. Conf. Commun. Signal Process. ICCSP 2017*, vol. 2018-Janua, pp. 2164–2167, 2018.

[17] P. Srivastava, M. Bajaj, and A. S. Rana, “Overview of ESP8266 Wi-Fi module based smart irrigation system using IOT,” *Proc. 4th IEEE Int. Conf. Adv. Electr. Electron. Information, Commun. Bio-Informatics, AEEICB 2018*, Oct. 2018.

[18] M. Swapnali, B. Pawar, P. Rajput, and A. Shaikh, “Smart Irrigation System Using IOT And Raspberry Pi,” *Int. Res. J. Eng. Technol.*, p. 1163, 2008, Accessed: Oct. 04, 2021. [Online]. Available: www.irjet.net.

[19] S. R. Waddington, H. K. Murwira, J. D. T. Kumwenda, D. Hikwa, and F. Tagwira, “Soil fertility research for maize-based farming systems in Malawi and Zimbabwe,” 1998, Accessed: Nov. 05, 2021. [Online]. Available: https://repository.cimmyt.org/handle/10883/539

[20] G. S. Pravallika, L. Kundana, K. S. Thanvi, G. Sirisha, and C. Rupa, “Proficient Smart Soil based IoT System for Crop Prediction,” *Proc. 2nd Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2020*, pp. 752–757, Jul. 2020, doi: 10.1109/ICIRCA48905.2020.9183054.

[21] “Artificial Intelligence in Agriculture - APRO Software.” https://apro-software.com/artificial-intelligence-in-agriculture/ (accessed Sep. 18, 2021).

[22] P. S. Vijayabaskar, R. Sreemathi, and E. Keertanaa, “Crop prediction using predictive analytics,” *6th Int. Conf. Comput. Power, Energy, Inf. Commun. ICCPEIC 2017*, vol. 2018-January, pp. 370–373, Feb. 2018, doi: 10.1109/ICCPEIC.2017.8290395.

[23] K. K. Ghanshala, R. Chauhan, and R. C. Joshi, “A Novel Framework for Smart Crop Monitoring Using Internet of Things (IOT),” *ICSCCC 2018 - 1st Int. Conf. Secur. Cyber Comput. Commun.*, pp. 62–67, Jul. 2018, doi: 10.1109/ICSCCC.2018.8703366.

[24] S. Kapse, S. Kale, S. Bhongade, S. Sangamnerkar, and Y. Gotmare, “IOT Enable Soil Testing & NPK Nutrient Detection.”

[25] M. Bajceta, P. Sekulić, B. Krstajic, S. Đukanović, and T. Popovic, “A Private IoT Cloud Platform for Precision Agriculture and Ecological Monitoring,” *Proc. 3rd Int. Conf. Electr. Electron. Comput. Eng.*, no. June, 2016, [Online]. Available: https://www.researchgate.net/publication/303940620_A_Private_IoT_Cloud_Platform_for_Precision_Agriculture_and_Ecological_Monitoring.

[26] K. Spandana and S. Pabboju, “ICICCT 2019 – System Reliability, Quality Control, Safety, Maintenance and Management,” *ICICCT 2019 – System Reliability, Quality Control, Safety, Maintenance and Management*, 2020. http://dx.doi.org/10.1007/978-981-13-8461-5_31.

[27] “Home - AI-generated Synthetic Data.” https://mostly.ai/ (accessed Sep. 17, 2021).

[28] “JIRA Waterfall Model - Javatpoint.” https://www.javatpoint.com/jira-waterfall-model (accessed Oct. 19, 2021).

[29] Cortex, “STM32F401xB STM32F401xC (ADCS Datasheet),” no. September, 2019, [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f401cb.pdf.

[30] “Labcenter Electronics, "PCB Design & Simulation Made Easy" [Online] https://www.google.com/search?q=https%3A%2F%2Fwww.labcenter.com+%2F&rlz=1C1CHBD_enRW910RW910&oq=https%3A%2F%2Fwww.labcenter (Last accessed Sep. 20, 2021). (accessed Sep. 20, 2021).

[31] “Soil NPK Sensor Probe RS485 – Electronics Devices| Microcontrollers| Sensors | IoT Components and Solutions.” https://homchip.com/product/rs485-soil-npk-ph-sensor-probe-npk-sensors-detector/ (accessed Oct. 19, 2021).

[32] “Buy SIM800A GSM Module with RS232 Interface Online at Lowest Price.” https://robu.in/product/sim800a-quad-band-gsmgprs-module-rs232-interface/ (accessed Oct. 15, 2021).

[33] GitHub, "Crop_recommendation. [Online] https://github.com/gabbygab1233/Crop-Recommender/blob/main/Crop_recommendation.csv (last accessed Sep. 27, 2021).

APPENDICES

Appendix 1: Notification of paper acceptance for Publication

SCOReD Notification for Paper 1570766195 Inbox x

tanminkeng=u...@edas.info
to me, Jimmy, Gerard, Ignace

Oct 26, 2021, 6:00 AM (13 days ago)

Dear Rosemary Nalwanga, Jimmy Nsenga, Gerard Rushingabigwi and Ignace Gatare,

We are pleased to inform you that your paper

Paper ID: 1570766195
Paper title: Design of an Embedded Machine Learning Based System for an Environmental-Friendly Crop Prediction Using a Sustainable Soil Fertility Management

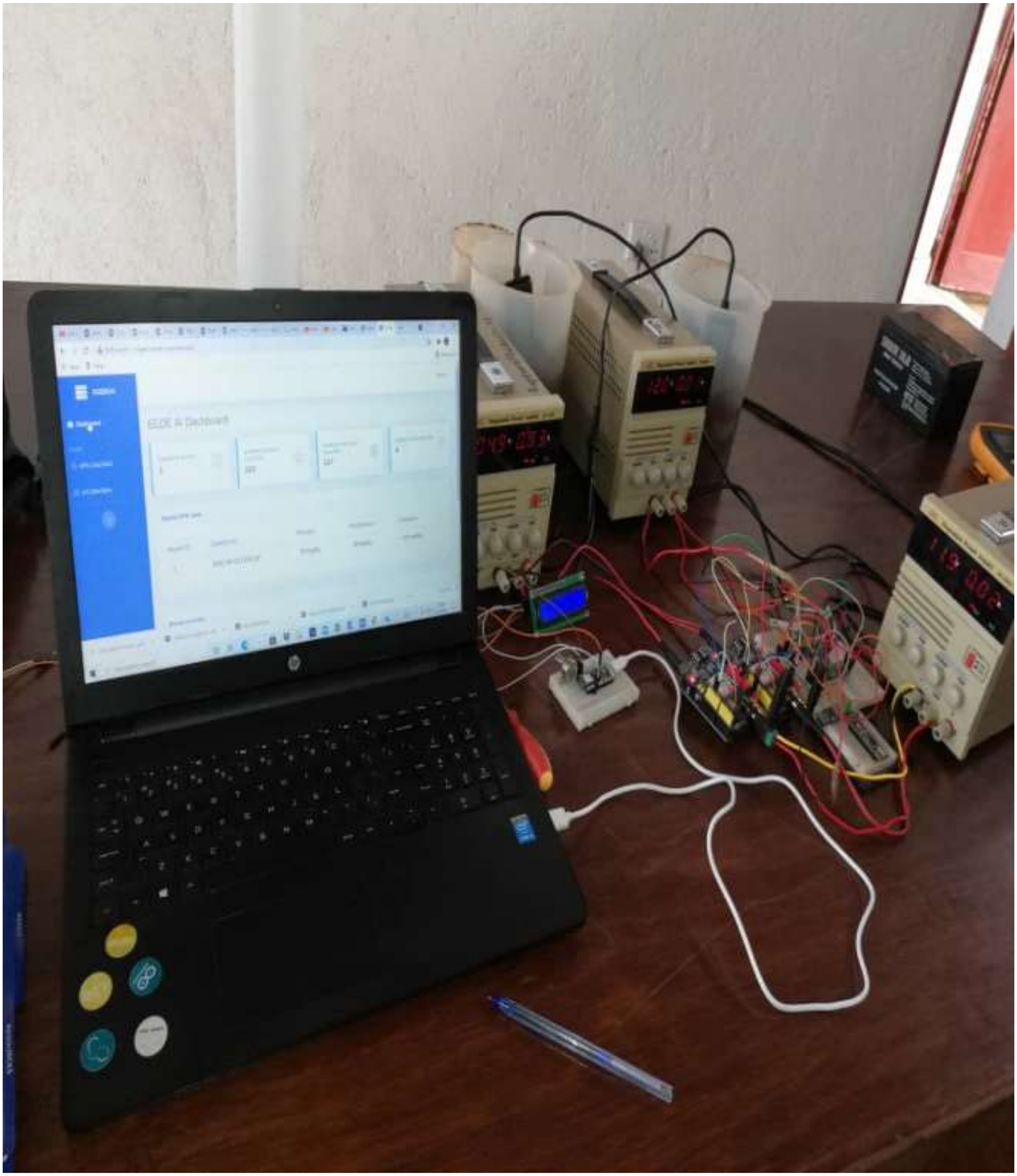
has been ACCEPTED for the 19th IEEE Student Conference on Research and Development (SCOReD 2021), subjected to corrections made based on the reviewers' comments. Your paper may still be rejected if this is not fully addressed.

The template for summary of correction is available in shorturl.at/arQ49. Please submit the summary of correction along with the camera-ready paper. The deadline for camera-ready paper is on 1 November 2021.

Please ensure that the similarity score for the final manuscript does not exceed the allowable 30 % and each individual source is not more than 10 %, as to comply with the IEEE regulation. If this is not followed, your paper may still be rejected.

This notification email serves as our formal acceptance letter for your paper and invitation to present the paper at the 19th IEEE Student Conference on Research and Development (SCOReD 2021). The acceptance of your paper is made with the understanding that at least one author will register and attend the conference to present the paper. Failure to do so, the paper will not be included for publication.

Appendix 2: The system prototype setup



Appendix 3: The Edge AI Crop Prediction Code

```
#include <newmodel_inferencing.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define THRESHHOLD 0.6

// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x3F, 16, 2);
staticconst float features[] = {61.0000, 44.0000, 17.0000, 26.1002, 71.5748, 6.9318, 102.2662
    // copy raw features here (for example from the 'Live classification' page)
    // see https://docs.edgeimpulse.com/docs/running-your-impulse-arduino
};
/**
 * @brief      Copy raw feature data in out_ptr
 *             Function called by inference library
 *
 * @param[in]  offset   The offset
 * @param[in]  length   The length
 * @paramout_ptr  The out pointer
 *
 * @return     0
 */
intraw_feature_get_data(size_t offset, size_t length, float *out_ptr) {
memcpy(out_ptr, features + offset, length * sizeof(float));
return 0;
}
```

```
void setup()
{
Serial.begin(115200);
Serial.println("Edge Impulse Inferencing Demo");
}
void loop()
{
lcd.clear();
float N=features[0];
float P=features[1];
float K=features[2];
float humidity=features[3];
float temp=features[4];
floatph=features[5];
float rainfall=features[6];
ei_printf("value of N:");
Serial.println(N);

lcd.setCursor (0,1); // go to start of 2nd line
lcd.print("value of N:");
lcd.setCursor (1,1);
lcd.print(N);
lcd.setCursor (0,4); // go to start of 2nd line
lcd.print("value of P:");
lcd.setCursor (1,4);
lcd.print(P);
lcd.setCursor (0,8); // go to start of 2nd line
lcd.print("value of K:");
lcd.setCursor (1,8);
lcd.print(K);
```

```
ei_printf("value of P:");
Serial.println(P);
ei_printf("value of K:");
Serial.println(K);
ei_printf("value of humidity:");
Serial.println(humidity);
ei_printf("value of temperature:");
Serial.println(temp);
ei_printf("value of PH:");
Serial.println(ph);
ei_printf("value of rainfall:");
Serial.println(rainfall);
ei_printf("Edge Impulse standalone inferencing (Arduino)\n");
if (sizeof(features) / sizeof(float) != EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE) {
ei_printf("The size of your 'features' array is not correct. Expected %lu items, but had %lu\n",
        EI_CLASSIFIER_DSP_INPUT_FRAME_SIZE, sizeof(features) / sizeof(float));
delay(1000);
return;
   }
ei_impulse_result_t result = { 0 };
   // the features are stored into flash, and we don't want to load everything into RAM
signal_tfeatures_signal;
features_signal.total_length = sizeof(features) / sizeof(features[0]);
features_signal.get_data = &raw_feature_get_data;
   // invoke the impulse
   EI_IMPULSE_ERROR res = run_classifier(&features_signal, &result, false /* debug */);
ei_printf("run_classifier returned: %d\n", res);
if (res != 0) return;
lcd.clear();
```

```
    // print the predictions
ei_printf("Predictions ");
lcd.print("Predictions ");

ei_printf("(DSP: %d ms., Classification: %d ms., Anomaly: %d ms.)",
        result.timing.dsp, result.timing.classification, result.timing.anomaly);
ei_printf(": \n");
ei_printf("[");
for (size_t ix = 0; ix < EI_CLASSIFIER_LABEL_COUNT; ix++) {
ei_printf("%.5f", result.classification[ix].value);
#if EI_CLASSIFIER_HAS_ANOMALY == 1
ei_printf(", ");
#else
if (ix != EI_CLASSIFIER_LABEL_COUNT - 1) {
ei_printf(", ");
        }
#endif
    }
#if EI_CLASSIFIER_HAS_ANOMALY == 1
ei_printf("%.3f", result.anomaly);
#endif
ei_printf("]\n");
    // human-readable predictions
for (size_t ix = 0; ix < EI_CLASSIFIER_LABEL_COUNT; ix++) {
ei_printf("    %s: %.5f\n", result.classification[ix].label, result.classification[ix].value);
if (result.classification[ix].value>=THRESHHOLD)
        {
lcd.clear();
lcd.setCursor(0,0);
lcd.print(result.classification[ix].label)
```

```
lcd.setCursor(0,1);
lcd.print(result.classification[ix].value);
        }
    }
#if EI_CLASSIFIER_HAS_ANOMALY == 1
ei_printf("    anomaly score: %.3f\n", result.anomaly);
lcd.print(result.classification[ix].label, result.classification[ix].value);
#endif
delay(1000);
}
* @brief      Printf function uses vsnprintf and output using Arduino Serial
* @param[in]  format     Variable argument list
voidei_printf(const char *format, ...) {
static char print_buf[1024] = { 0 };
va_listargs;
va_start(args, format);
int r = vsnprintf(print_buf, sizeof(print_buf), format, args);
va_end(args);
if (r > 0) {
Serial.write(print_buf);
    }
}
```