TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Minor Project Report
On
IoT Device for Crop Recommendation using AI

Submitted By:

Aashish Pant (31253)

Kaustub Niraula (31262)

Rajendra Baskota (31271)

Rijan Ghimire (31272)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

March, 2023

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Minor Project Report
On
IoT Device for Crop Recommendation using AI

Submitted By:

Aashish Pant (31253)

Kaustub Niraula (31262)

Rajendra Baskota (31271)

Rijan Ghimire (31272)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

In partial fulfilment for the award of the Bachelor's Degree in Electronics, Communication and Information Engineering.

Under the Supervision of

Er. Saroj Shakya

March, 2023

DECLARATION

We hereby declare that the report of the project called **"IOT device for crops recommendation using AI"** which is submitted to the department of **Electronics and Computer Engineering**, IOE, Thapathali Campus, in the partial fulfilment of the requirements for the award of the Degree of Bachelor of Engineering in **Electronics, Communication and Information Engineering,** is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any university or institution for the award of any degree and we are only author of this complete work and no sources other than the listed here have been used in this work.

Aashish Pant (Class Roll No.: THA076BEI003) ____________________________

Kaustub Niraula (Class Roll No.: THA076BEI012) ____________________________

Rajendra Baskota (Class Roll No.: THA076BEI021) ____________________________

Rijan Ghimire (Class Roll No.: THA076BEI022) ____________________________

Date: March, 2023

CERTFICATE OF APPROVAL

The undersigned certify that they read and recommended to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus** a minor project work entitled **"IOT device for crops recommendation using AI"** submitted by **Aashish Pant**, **Kaustub Niraula**, **Rajendra Baskota**, and **Rijan Ghimire** in partial fulfilment for the award of Bachelor's Degree in Electronics, Communication and Information Engineering. The project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled, and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfilment of bachelor's degree of Electronics, Communication and Information Engineering.

____

Project Supervisor

Er. Saroj Shakya

Department of Electronics and Computer Engineering, Thapathali Campus

____

External Examiner

Dr. Ganesh Gautam

Department of Electronics and Computer Engineering, Pulchowk Campus

____

Project Co-ordinator

Er. Umesh Kanta Ghimire

Department of Electronics and Computer Engineering, Thapathali Campus

____

Er. Kiran Chandra Dahal

Head of the Department

Department of Electronics and Computer Engineering, Thapathali Campus

March, 2023

COPYRIGHT

ACKNOWLEDGEMENT

Foremost, we would like to extend our sincerest gratitude towards Thapathali Campus, a constituent engineering campus of Institute of Engineering, Tribhuvan University, the Department of Electronics and Computer Engineering, all the teaching /non-teaching staff members of this family for providing us with such a wonderful opportunity to enhance our skills and mind with the inclusion of this project in our curriculum.

We would also like to express our deep gratitude to our supervisor **Er. Saroj Shakya** for providing us with invaluable guidance and support.

We are also thankful to our classmates and fellow friends for their encouragement, constructive feedback and assistance which gave us even more ability to innovate and solve.

The student members of this project are listed below:

Aashish Pant (Class Roll No.: THA076BEI003)

Kaustub Niraula (Class Roll No.: THA076BEI012)

Rajendra Baskota (Class Roll No.: THA076BEI021)

Rijan Ghimire (Class Roll No.: THA076BEI022)

ABSTRACT

Agriculture is still the major industry of Nepal. According to source, it engages around 70% of the total population of the country and contributes a major portion towards the nation's GDP. Smart Farming, often heard of but never experienced, never witnessed is the way to revolutionise the farming sector. The emerging technologies such as IoT and AI can increase the quantity and quality of products while reducing the human labour required for production. The proposed project tries to increase the productivity and efficiency of growing various crops in multiple environmental conditions. It aims to construct a prototype to measure the environmental parameters such as pH, temperature, humidity and NPK and train a machine learning model. The model outputs the suitable crop for the given environment. This model is trained over a particular set of data and incorporates various seasonal plants. It aims to develop a method for farmers or hobbyist gardeners to find a better crop according to the soil that they find in their yards. The system consists of temperature, humidity and pH sensor and is connected through a web application.

Keywords: *AI, IoT, pH, NPK, ML, Smart Farming*

Table of Contents

List of figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
AP	Access Point
ARM	Advanced RISC Machine
DRL	Deep Reinforcement Learning
HTTP	Hyper Text Transfer Protocol
IC	Integrated Circuit
IDE	Integrated Development Environment
IoT	Internet of Things
JS	JavaScript
LCD	Liquid Crystal Display
MAC	Media Access Control
ML	Machine Learning
NPK	Nitrogen Phosphorus and Potassium
NTC	Negative Temperature Coefficient
pH	Potentiality of Hydrogen
SoC	System on Chip
STA	Station
SVM	Support Vector Machine
URL	Uniform Resource Locator
WSN	Wireless Sensor Network

1. INTRODUCTION

1.1 Background

Majority of the population in Nepal still rely totally on agriculture for their day-to-day operation. Agriculture has been the backbone of our country for a long time. Paddy is the most produced crop in Nepal followed by Maize, Wheat, Millet, Buckwheat and Barley. The cash crops cultivated here include oil seeds, potato, sugarcane, jute and cotton. But the method of farming in Nepal is mostly traditional which has barred farmers from obtaining their full potential. Due to cultivation of wrong plants, farmers have not been able to earn the maximum profit from their farming. Farmers in Nepal even face losses as their cultivated plants cannot thrive in their farmlands. This creates a dilemma among the farmers while selecting the crop to cultivate. They just shift towards another crop when one cannot thrive without proper research just like a hit and trial method. Although there has been some modernization with the introduction of ploughing machines, threshers and so on, these machines only help to reduce manual labour. We need systems which can automate the process of agriculture, including a system which can recommend the crops based on our soil parameters and environment.

An automated system requires implementation of Artificial Intelligence (AI) and Machine Learning (ML) algorithms combined with IoT. In the era of automation, Machine Learning algorithms are getting popular day by day. ML is being ubiquitously used in every field, like: Agriculture, Health Sectors, Computer Vision, Natural Language Processing, Military etc. AI can assist in agriculture with crop recommendation systems, disease recognition, automated irrigation and many more. In this paper we are focused on crop recommendation systems. In this paper we propose that one way to develop a crop recommendation system is to incorporate AI and IoT. With the advancement in IoT, several sensors can be incorporated in the system through which we can obtain several necessary parameters from the soil and the environment, such as: humidity, pH value, NPK value, precipitation, moisture, altitude etc. These parameters when fed into a learning model can recognize patterns in the data and can ultimately recommend the best crop based on our environment and soil parameters.

1.2 Motivation

A certain plant cannot thrive in all environments. It needs some specific parameters of soil and environment. In a similar way, based on the altitude of that place and other factors of soil and water, a certain environment is only able to support some certain groups of plants and animals. Farming without considering this fact has brought farmers at risk. Farmers aren't getting a supportive platform where they can easily figure out which crops would be most suitable for their farm. Based on the amount of precipitation and other contingent environmental factors, it can also occur that a particular crop might be suitable for this particular year and not for the other. This dynamic nature of the environment has increased tension among farmers. We are trying to solve this problem by building a system which can recommend the most suitable crop for a farmer's farm.

1.3 Problem Definition

Various factors determine if a crop can thrive in a certain farmland. Failing to consider those parameters can cause huge loss to a farmer. Traditional method of cultivation by figuring out the ripeness of the soil and being in a dilemma about figuring out the best crop isn't the optimal method. There are several values to consider for a profitable farming. The pH value which affects the availability of soil nutrients, the water holding capacity which determines the crop's ability to absorb nutrients, the altitude, moisture, and humidity are the core determining factors which determine if a crop is suitable for a particular environment. The problem is to figure out the best crop for the field given several parameters of the soil and the environment.

1.4 Objectives

1) To make a prototype of wireless sensor network of 3 nodes using pH, temperature and humidity sensor using HTTP and ESP-NOW.
2) To build an AI model using SVM and use it to recommend the best crop forthe field based on the data from sensor network.
3) To build a simple UI for farmers.

1.5 Scope and Application

1.5.1 Scope

An IoT device is assembled using sensors such as pH, temperature and humidity to measure the soil parameters. The measured parameters are communicated through the field to the database server using NodeMCU module. The database server provides easy access to the data for further analysis.

A web/mobile app is developed for farmers or agricultural technicians to view soil parameters from the field and monitor their farm from afar. Once the data is collected and stored in the database server, an ML model is used to provide crop recommendations based on the soil parameters.

The ML model can analyse the soil parameters and suggest crops that would thrive in the soil conditions. The use of an ML model can lead to increased efficiency and productivity in agriculture.

Out of scope:

- The training data is collected from open sources of internet and not by us.
- The analysis for number of users who can connect to the system at the same time is yet to be done.
- This system doesn’t include information and automation for irrigation, spraying pesticides and size of crops.
- The information about number of nodes required is not mentioned in the paper.

1.5.2 Application:

This project mainly helps the farmers to identify the correct crops to be planted during different seasons for maximum production and minimal utilisation of external resources. It avoids the need to try various crops or experimentation rather the plants can be planted without any hassle.

The system can be further developed to incorporate market values of crops according to seasons and calculate the total value of the planted produce. In this way the farmers can plant the crops which can be most beneficial financially.

This can be used to find the correct amount of nutrients needed for the plants according to the soil rather than drop bucket load of fertilisers which is not feasible economically or quantitatively. The correct amount of nutrients needed is displayed to the user and the needed nutrients can be further calculated with ease.

This model may be further developed into a smart farming solution with automated irrigation, pest control, fertilisation which will revolutionize the stone age farming techniques currently used in Nepal and many such countries to advance in their quality of living and grow the economy indirectly/directly.

1.6 Report Organization

This report comprises 8 chapters, each covering various topics related to the project. Chapter 1 introduces the smart agriculture process using IoT devices and crops predictions approaches for precision farming, project objective its applications and limitations. Chapter 2 includes the historical background and essential information on Crops prediction system and IoT in agriculture. Chapter 3 explains the hardware and software components and about the feasibility of the project. Chapter 4 describes the methodology and working principle of the project, including a block diagram of the system and the different approaches and algorithms used for predictions. Chapter 5 describes the implementation of software and hardware along with the calibration process. The result and analysis of the project are described in the chapter 6. Chapter 7 includes the future enhancements that we can implement to our project and chapter 8 contains the appendices.

2. LITERATURE REVIEW

Agriculture is one of the most important sectors in many countries, and it has seen a significant amount of research and development over the years. With the advent of the Internet of Things (IoT), the agricultural industry has seen a significant transformation in terms of the way it operates. The use of IoT in agriculture has brought a new level of efficiency, profitability, and sustainability to the field.

Traditional agriculture was limited by the farmers' lack of knowledge about certain factors that could affect crop yield, such as humidity, water levels and other climatic conditions. However, with the help of IoT, modern agriculture is now able to take these factors into account. IoT technology is used to collect data from sensors that monitor the climate, temperature, moisture, humidity, and water level of the soil. TM. Naresh and P. Munaswamy proposed a system where the information collected from the sensors is processed by an ARM-7 processor with the LPC2148 IC to determine if the soil is lacking water. If this is the case, a relay is used to turn on the motor to water the soil. This system primarily focuses on reading sensor values and predicting water necessity [1]. This approach has its limitations as it does not address the connectivity of the system, such as the number of nodes that should be present in the field or how this data is sent to the server. It also uses a threshold value for water level and soil moisture, which can be improved upon. Our proposed model takes a more advanced approach by utilizing machine learning and neural networks to recommend crops based on the parameters observed from the sensors.

V S Narayanan et al. proposed a system that presents a combination of IoT, machine learning, and mobile applications for the development of a smart irrigation system with soil and weather monitoring. The use of Random Forest Regression Algorithm for crop prediction has been shown to achieve a high accuracy rate. The system has the potential to be further enhanced in the future by implementing an NPK sensor to analyse crop nutrition, which could increase crop yield. The system uses the manual data entry in the mobile app which can be addressed by using the NodeMCU as a microcontroller or a Wi-Fi module [2].

Pradeepa Bandara, Thilini Weerasooriya et al. conducted research to implement crop recommendation system in Sri Lanka where the space for domestic lands is very less and it is absolutely essential to select the most suitable crop for plantation. They take in account environment factors such as pH, soil moisture, sunlight, temperature, and humidity. They have used an Arduino Uno board as a microcontroller and a separate ESP32 for data transmission through Wi-Fi. Two machine learning algorithms, Naive Bayes, and Support Vector Machine (SVM) were trained with the accuracy of 90%. There is also the implementation of feedback system from farmers. The feedbacks are processed using Natural Language Processing and the feedbacks are taken into account to update the model. But the model could only recommend four different crops and the paper doesn't mention about the sensor network. We have trained model with 22 different classes and are also working to maintain a Wireless Sensor Network [3].

Unlike most of the papers, the paper by Mani D et al. uses different set of parameters to predict crop yield and to recommend the most appropriate crops. They take in parameters like, district, rainfall, temperature and area of the field. The authors use region-specific attributes for 6 different crops. The basic idea is to take specific climatic as well as regional conditions to predict the optimal yield. The crop-yield prediction is done using Ridge Regression and XGB Regressor. Similarly, they have implemented XGB Regressor and LGBM classifier for crop prediction. The XGB regressor performed better than Ridge Regression for the crop yield prediction problem. Whereas the outcome of LGBM Classifier was better than that of XGB Regressor for crop prediction system [4].

Nari Kim et al in this paper compares different AI models in order to develop the best crop yield prediction model for the US. They conducted the extraction of data from satellite products and constructed a database for corn and soybean which was then fed into five major AI model for crop yield prediction. They utilized the following algorithms Support Vector Machine, Random Forest, Extremely Randomized Trees, Artificial Neural Network, Deep Neural Network for prediction. They performed an optimization process to ensure the best configurations for the layer structure, cost function, optimizer, activation function, and drop-out ratio, to improve the crop yield prediction accuracy particularly for DNN model [5].

The optimized DNN model developed in this study can also be adopted for other crops, only if the parameter optimization is conducted and more appropriate input variables should be added to a modified DNN model. We have devised a simpler approach to this problem using single SVM to predict the crops.

S. Pudumalar et al. have discussed the problem of predicting the right crop for maximum yield using the ensemble model with several learners like: Random tree, K-Nearest Neighbour and Naive Bayes. They have used several parameters of soil which play a major role in crop's ability to extract water and nutrients such as: depth, texture, pH, soil color, permeability, drainage, water holding and erosion. They have implemented Majority Voting Technique to predict the crop from the ensemble model. Several base learners are trained such that the learners are competent to each other yet being complimentary as well. During inference, each model predicts the class on its own and the class which is predicted by most of the learners is chosen. This paper presents a different approach than majority of the papers as it implements ensemble of the multiple base learners [6].

Another important aspect of agriculture is selecting the appropriate crops based on the soil type to develop a profitable agriculture system. Farmers often face a dilemma while selecting the crops and end up choosing the wrong one, affecting crop production. To address this issue, M. Bouni, B. Hssina, K. Douzi, and S. Douzi proposed a system which uses various learners such as Random Forest, K-Nearest Neighbour, Naive Bayes, and Deep Reinforcement Learning (DRL) to train the model and ultimately recommend a crop based on parameters such as temperature, soil, moisture, rainfall, and humidity [7]. The study found that Neural Network-based Deep Reinforcement Learning and Random Forest gave the best performance. However, the authors did not incorporate the communication of data between different channels, such as the node configuration of the sensors or the medium through which they communicate with each other. It also does not define how the data is collected for testing. Our model, on the other hand, provides a more end-to-end solution for crop recommendation, ranging from data acquisition to the final predicted output.

M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem conducted a study focusing on survey of the role of IoT in agriculture for the implementation of Smart Farming [8]. The paper describes a system that includes databases, servers, gateways for communication, a user interface/IoT-based application for disease recognition and a fire alarm. The system uses multiple sensors to measure real-time data, such as pressure, gas, temperature, and moisture, which is then processed by a microprocessor. The data collected by the system is fed through an image/video processor, making the system more reliable and efficient. The proposed system can be used in various domains, including precision farming, livestock management, and greenhouse monitoring. The paper also addresses the hardware and networking challenges that can occur while implementing these concepts.

In our model, we aim to connect the nodes wirelessly, and if the farm does not have access to the internet, a long-range communication module can be deployed to provide internet access. the idea of WSN can also be implemented in such a situation deploying many sensing units in the field and collecting the information through a central node [9]. Wireless communication is the most suitable for the areas where this prototype is to be used. The hardware challenges include the exposure of the sensors to the hard environment 24/7.

3. REQUIREMENT ANALYSIS

3.1 Hardware Requirements

3.1.1 NodeMCU esp8266

We chose NodeMCU for our project over other microcontrollers like Arduino because of its built-in Wi-Fi capabilities and its economic cost. Also, NodeMCU is smaller and compact than Arduino making it more suitable for IoT projects. It also includes power management system to reduce overall power consumption of the device. It includes firmware that works on Espressif systems ESP8266 Wi-Fi SoC and hardware based on the ESP-12 module. NodeMCU can function both as a data sender (publisher) and a data receiver (subscriber) [10]. The 32-bit LX106 RISC microprocessor present in its board can run at clock frequencies between 80 and 160 MHz and operate with real-time operating systems. The NodeMCU device has 128 KB of RAM and 4 MB of Flash memory for storing data and programs [11]. It supports UART, SPI, and I2C interface.

NodeMCU can be powered using a Micro USB jack and VIN pin (External Supply).

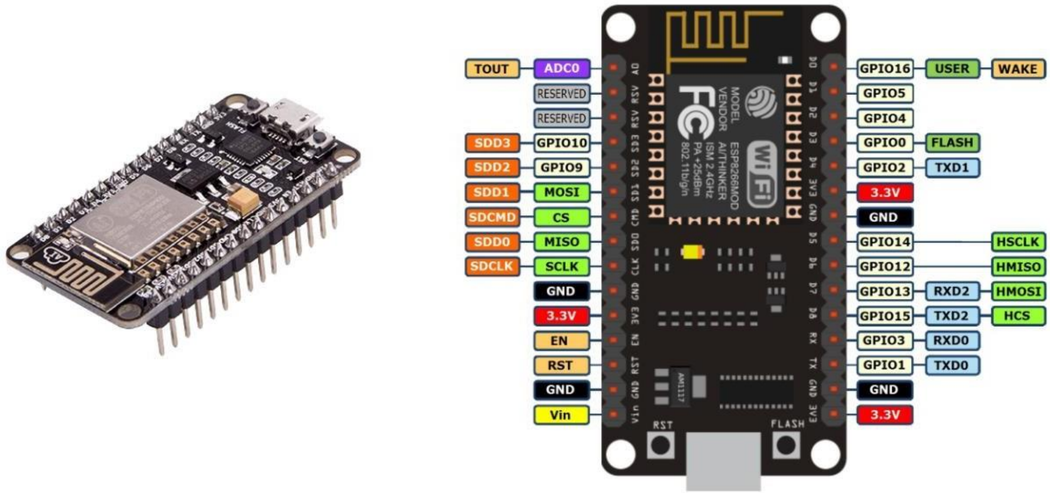


Figure 3-1: ESP8266 NodeMCU

3.1.2 DHT11 Sensor

DHT11 Temperature and Humidity Sensor features a temperature and humidity sensor complex with a calibrated digital signal output. This sensor includes a capacitive sensing element to measure the humidity and a thermistor with Negative Temperature Coefficient (NTC) to measure the temperature. It is able to accurately measure temperature between 0°C and 50°C and humidity level between 20% and 90% RH. It connects to a high-performance 8-bit microcontroller and has a sampling rate of 1Hz i.e., takes a reading once per second. The main purpose of the sensor is to measure the temperature and humidity of the surrounding air. [12].

This sensor consists of a capacitive humidity sensing element to measure humidity and a thermistor to measure the temperature of the surrounding air. The humidity sensing capacitor consists of two electrodes between which a moisture holding substrate is kept as dielectric and the capacitance changes when there is a change in humidity levels as the properties of the dielectric also change. Also, to measure the temperature a negative temperature coefficient thermistor is kept whose resistance decreases when the temperature increases. These changes are then measured by an IC and processed, computed into digital values that can be taken as output from the DATA pin of this sensor. The sensor works in this manner with ±2°C accuracy for temperature and ±5% accuracy for humidity.

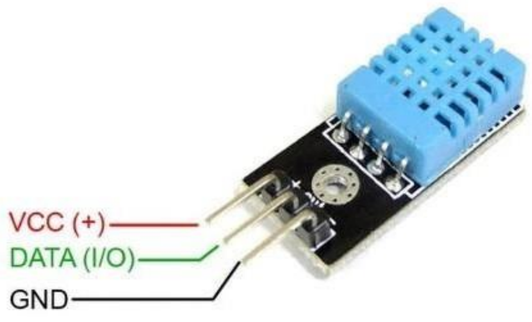


Figure 3-2: DHT11 Senor Module

3.1.3 pH sensor

The pH sensor consists of a pH module and a probe which is used to measure the pH of solutions. It measures the alkalinity and acidity of the solutions and can operate when a voltage of 5V is provided to the sensor. It measures the pH of a solution by comparing it with the reference electrode present in its glass tubing and an output analog value is given by between 0 – 1023. The calibration of the sensor is required for the proper performance, and it has to be done regularly in a certain interval.

The basic working principle of this sensor is potentiometry as such it measures the electric potential of the solutions and outputs some voltage. The sensor contains two electrodes consisting of a reference electrode of known electric potential and another sensor electrode which is to be dipped in the unknown solution. The reference electrode is generally neutral (pH=7). We know that acids have high concentration of H^+ions due to which they can conduct electric current. When the sensor electrode is inserted into a solution, there is exchange H^+ ions. This way the circuit is completed, and a certain potential is developed across it. The electric potential is then calculated by the difference between the reference electrode and the sensor electrode. This potential is then received as output from the pin of the pH module.

Figure 3-3: pH sensor with Module

3.1.4 NPK sensor

NPK sensor measures the level of Nitrogen(N), Phosphorus(P) and Potassium(K) content of the soil. These components are extremely important to determine the fertility of the soil. The NPK sensor consists of high-quality probe and can be kept inside soil for a very long time. Its probes are rust resistance, electrolytic resistance as well as salt and alkali corrosion resistance. We require a Modbus module (RS485) to read data from the sensor as it contains Modbus communication port and cannot be directly read through a microprocessor. The sensor operates on 9-24V and has the resolution up to 1mg/kg. Its measuring range is 0-1999mg/kg and works on different baud rate like 2400, 4800 and 9600.

There are various types of Soil NPK sensors. One of them uses electromagnetic induction to measure the electrical conductivity of the soil. This conductivity is directly related to the levels of N, P, and K in the soil. The sensor emits a low-frequency electromagnetic field that induces an electric current in the soil. The current is then measured and analysed to determine the levels of N, P, and K in the soil.

Another type of soil NPK sensor uses optical method to measure the levels of N, P, and K in the soil. This sensor emits light of different wavelengths onto the soil and measures the reflectance or absorption of the light by the soil. Different wavelengths of light are absorbed or reflected differently by the soil depending on the levels of N, P, and K present in the soil. The sensor then analyses this data to determine the nutrient levels.

Figure 3-4: Soil NPK sensor

3.2 Software Requirements:

3.2.1 Scikit-learn:

We used Scikit-learn to train some of our machine learning models as it is an efficient library to build reliable machine learning models. It provides a range of algorithms, such as linear regression, decision trees, random forests, support vector machines. Also, it offers various tools for model selection, pre-processing and evaluation. We used Scikit-learn to train our SVM model. Training with Scikit-learn also provided us with the advantages of easy deployment.

3.2.2 Jupyter notebook:

Jupyter Notebook provides a platform for interactive computing and data analysis. Using Jupyter Notebook allowed us to create documents that contain live code, equations, visualizations, and narrative text. We used jupyter notebook because of the interactive platform it provides for visualization and evaluation of code cells which is difficult with other code editors. It also enabled collaboration between the team members as multiple people can edit the notebook simultaneously.

3.2.3 Fast API

We used FastAPI to create API endpoints as it allowed us to create easy-to-use APIs with the features of automatic data validation and serialization. It enables users to APIs quickly and efficiently because of its simple and intuitive syntax. Also, the key advantage of using FastAPI is that it provides automatic documentation of our custom-made APIs. It supports asynchronous programming, making it a good choice for building highly scalable APIs. We used API endpoints from FastAPI to perform both HTTP request and response from NodeMCU as well as fronted React server.

3.2.4 ReactJS

We used ReactJS to take advantages of the concept of virtual DOM. Using virtual DOM improves the performance of the apps as it only reloads the updated components of the page rather than reloading the whole page. Everything in React is designed as a component which makes it easier to handle the codebase. The components are reusable, making it easier to develop and maintain large-scale applications.

We used ReactJS to create our frontend server where users can connect and see the recommendations for their fields.

3.2.5 MySQL

We used SQL to create a database to store the information about the node and the person associated with the node. Since there are number of nodes in a wireless sensor network, it is important to keep track of incoming sensor data from field with the owner associated with it. MySQL is an open-source Relational Database Management System (RDBMS) and is a popular choice for web-based applications due to its ease of use, reliability, and performance.

4. SYSTEM ARCHITECTURE AND METHODOLOGY

4.1 System Block Diagram

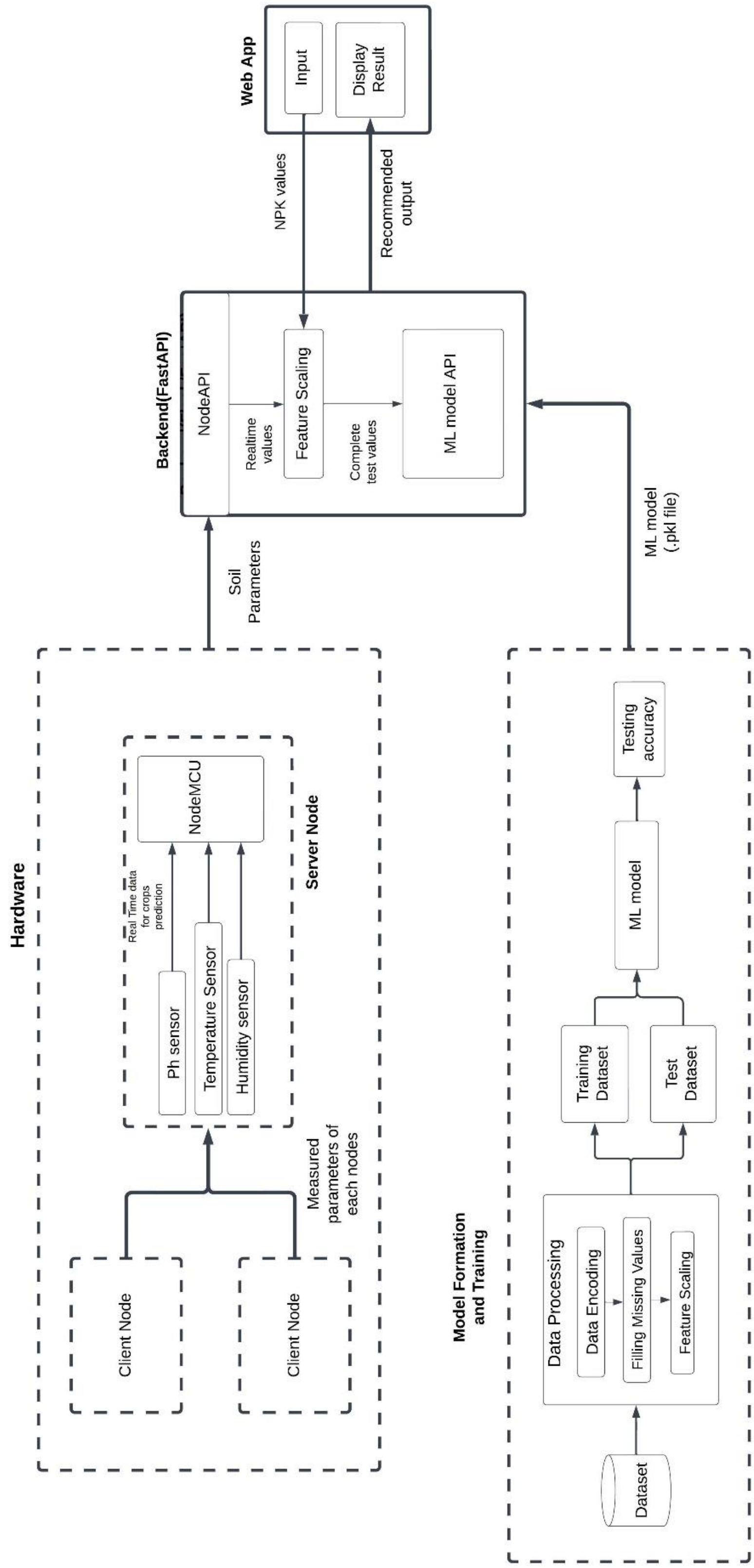


Figure 4-1: System Block Diagram

4.1.1 Hardware

The hardware part of this project consists of a NodeMCU microcontroller connected with several sensors to measure the data according to the soil conditions and send these data into the backend server. Multiple NodeMCU are designed to communicate with each other using Wireless Sensor Network (WSN) technique according to the client-server architecture.

Client Node: Each client node consists of pH, humidity and temperature sensors. All of them take their individual readings from the soil. The measured parameters of each soil of each client node are sent to a server node.

Server Node: The server node too contains all the sensors as in a client node. The server node has one additional duty i.e., to communicate with the backend server. The server node is responsible to send all the data collected from its clients as well as its own sensor readings to a backend server through HTTP post request.

4.1.2 Backend

FastAPI is used in the backend to fetch the data from sensor end points and to send the output to the frontend. It fetches data from server NodeMCU and NPK values from the frontend.

Feature Scaling: This block represents the feature scaling part where the collected data are scaled according to the mean and standard deviation values obtained from the training set. The detailed implementation can be found in 5.4.3.

ML model API: This block represents the pickle file of a trained Machine Learning Model and the APIs required for inferencing/prediction. The scaled data is fed into the ML model utilizing its API for prediction. This outputs the most suitable crop. The output is then sent to frontend through FastAPI.

4.1.3 Model Formation and Training

At first, the dataset was extracted from Kaggle, and data was not collected by us. Then the dataset was split into train and test set in the ratio of 80:20. Then various data pre-processing steps were carried out such as data encoding, filling missing values and feature scaling (explained in 5.1). The training set was used to train 3 ML models as described below and the accuracy comparison and their analysis were carried out to pick the best algorithm. The best ML model with its parameters was saved into a pickle file and loaded into the backend server to process the incoming data.

4.1.4 Web App

Input: Users are presented with a simple UI where they are requested to enter their phone number and the NPK values from their field. The NPK values should be inserted based on different tests such as chemical means, laboratory tests etc. This method was used as the NPK sensor could not be utilized. The data collected here is sent into the backend server where all the computation happens. As multiple number of nodes can belong to a single person, an NPK input form for multiple nodes is designed which can be sent to the backend server in a single click.

Output: As the backend generates the recommendation for the provided soil parameters, the recommendation is received by the frontend. Then the output (recommendation) is displayed here in an easy-to-understand format.

4.2 Algorithms Used

4.2.1 Artificial Neural Network

It is a machine learning model which is inspired by the structure and functioning of a human brain. The neural network consists of an input layer, number of hidden layers and an output layer. All of these layers consist of number of neurons where each neurons performs complex mathematical calculations. Neurons in one layer are connected to all the neurons in pervious layer and the next layer. We have implemented the same feed forward network in our project. The calculations in neurons are done with the help of weights and biases associated with them.

Each node has its own input, from which it receives communications from other nodes and/or from the environment and its own output, from which it communicates with other nodes or with the environment. Finally, each node as a function f through which it transforms its own global input into output (Fig. 4.2.1) [14].

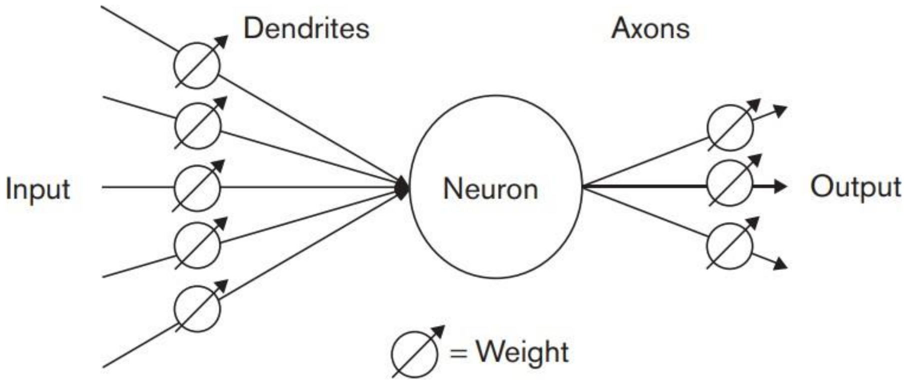


Figure 4-2: Representation of single neuron

Forward propagation:

In this method, the neural network carries the computation in the forward direction. The computation happens at input neurons first which are passed onto the next layer which used the output of first layers for its computation. In this way a linear/non-linear function is learned by the neural network going through many layers.

Each of these connections has a so-called weight, which modifies the input or output value. The value of these connection weights is determined during the training process. This functionality is the basis for the ANN's learning capability. Therefore, it is important to understand that there are no classification rules written into the algorithm. The network just learns to understand and classify input patterns from examples [14].

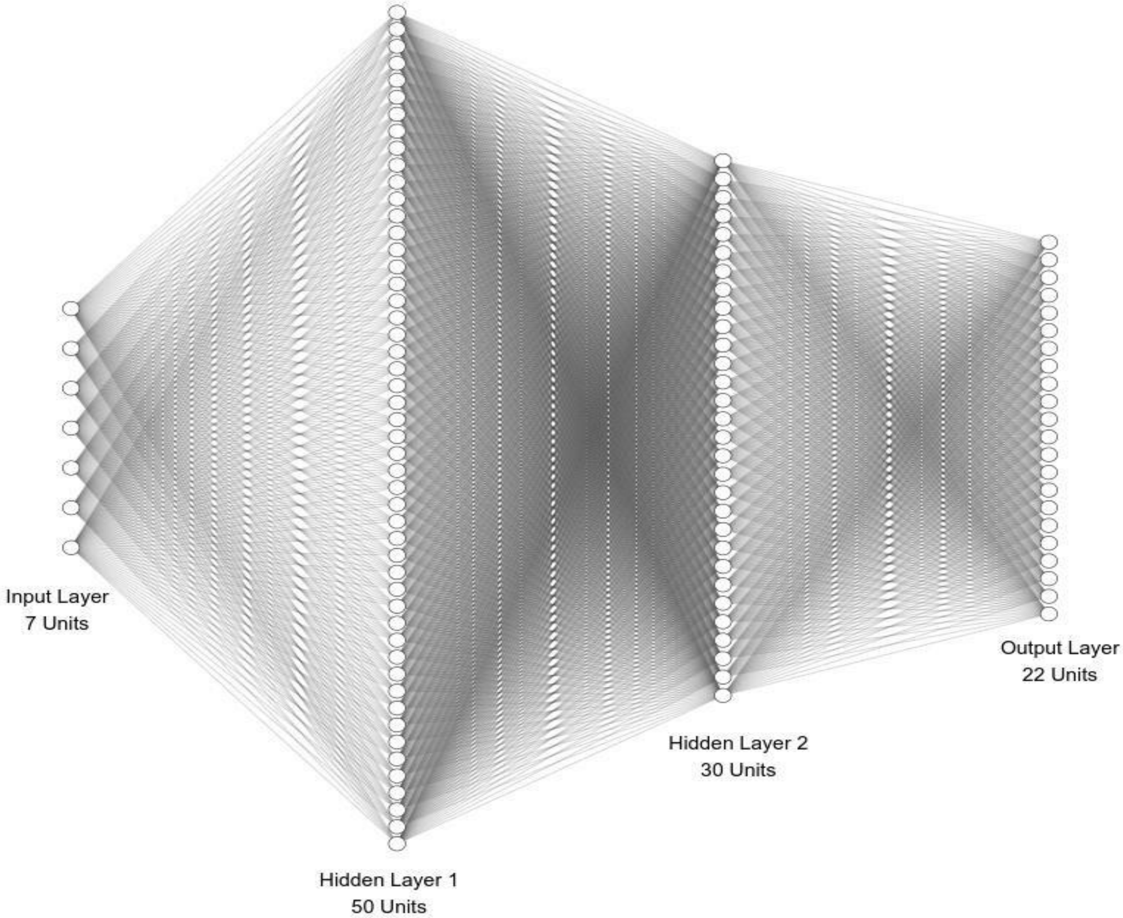


Figure 4-3: Artificial Neural Network

The neural network uses various activation functions to determine the overall output of the neuron and weather it will pass the signal to other neurons in the network.

If an activation function is not used in a neural network, then the output signal would simply be a simple linear function which is just a polynomial of degree one. [15]

Backward propagation:

In the backpropagation step, the error between the predicted outputs and actual outputs is calculated, and the gradient of the error with respect to the weights and biases is computed using the chain rule. The weights and biases are then updated for the best output with minimal error using gradient descent optimization or any other optimizers.

Gradient descent is an optimization technique which takes repeated steps towards the minima to minimize the cost function. Gradient is the partial derivative of a function along one axis. This algorithm takes the path with the steepest slope.

The learning rate η determines the size of the steps we take to reach a (local) minimum. In other words, we follow the direction of the slope of the surface created by the objective function downhill until we reach a valley [16].

The computations performed by the neurons at each layer are as follows:

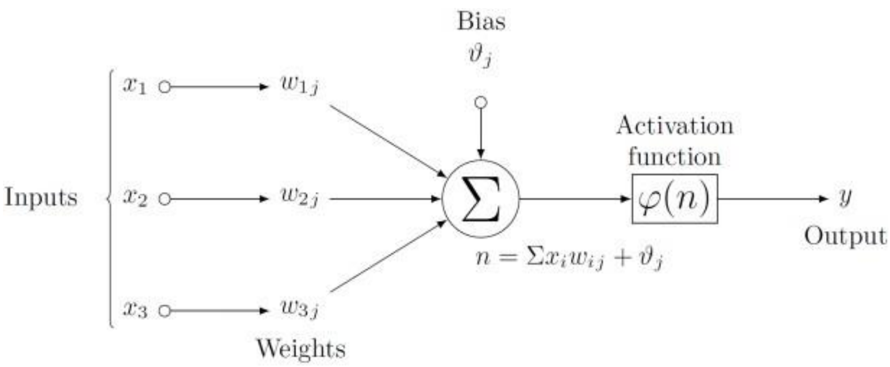


Figure 4-4: Calculation of Neural Network

At each neuron the calculation (X * W + B) is carried out and then passed onto an activation function to determine the output of a single neuron. Here X is the activation of the previous node and W and B are the weights and biases associated with the neurons.

4.2.2 Logistic Regression

Logistic Regression is a statistical method for analysing a dataset in which there are one or more independent variables that determine an outcome. It is used for binary classification problems where the dependent variable can only have two possible values (e.g., 0 or 1, yes or no, true or false). The logistic regression model uses a logistic function to model a probability estimate of the dependent variable given the independent variables. The output of the logistic function is transformed into a binary outcome using a threshold, typically set at 0.5. The threshold for classification is not always set at 0.5 but it can be set to any value depending on the specific problem and the trade-off between false positives and false negatives. The coefficients of the independent variables are estimated using maximum likelihood estimation. Logistic regression models can be regularized to prevent overfitting, i.e., the tendency of the model to fit the training data too closely and perform poorly on new data and also to prevent the coefficients from becoming too large leading to unstable models. This is done by adding a penalty term to the cost function used to determine the coefficients, which discourages the model from fitting the training data too closely.

In logistic regression, we fit a S shape curve which predicts two values 0 or 1. this is given by the Sigmoid function. This function is used to map any value within the range of 0 or 1. there are different types of logistic regression such as binomial or multinomial. We used this algorithm to predict the 22 different classes of crops. For this we utilized one vs all approach. In one vs all approach, we decompose a multinomial classification problem into many binary problems which makes it easier to handle. We decomposed the problem of classification of 22 classes into 21 binary classifiers which separates a single class. Once all the classifiers are trained, a prediction is performed for a new data by passing it through all the classifiers and the choosing the class with the highest prediction.

4.2.3 Support Vector Machine (SVM)

Support Vector Machine is a supervised machine learning algorithm that is commonly used for classification. It finds the optimal separating hyperplane between two classes. The hyperplane is responsible for maximizing the margin between the closest points between two classes. The data points closest to the hyperplane are known as support vectors. The choice of the hyperplane is based on the concept of maximum margin classifier, where the goal is to find a hyperplane that has the largest margin, or distance, between the two classes. SVM is particularly useful when the data is not linearly separable, as it can find non-linear decision boundaries by transforming the data into a higher dimensional space where the data becomes linearly separable.

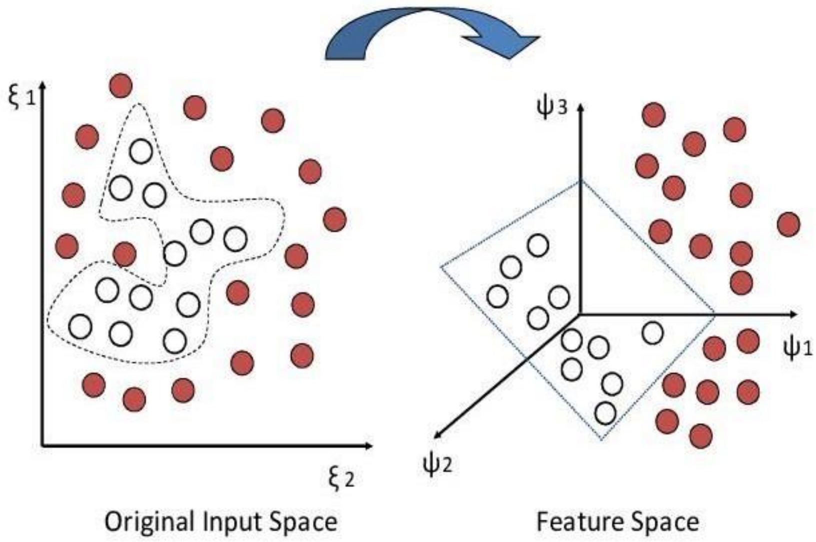


Figure 4-5: Support Vector Machine

The input data is transformed into higher dimensional space (feature space) using mathematical functions knows as kernel tricks or kernel functions. The choice of the kernel function depends on the structure of the data and the type of problem being solved. The purpose of the transformation is to create a linear boundary between the classes.

The widely used kernel functions are:

Polynomial kernel:

The polynomial kernel is defined as:

$$K(x, y) = (x * y + c)^d \tag{4-1}$$

where x and y are input vectors, c is a constant and d is the degree of the polynomial. This type of kernel is useful when the data is not linearly separable in the original feature space. By increasing the degree of the polynomial, more complex non-linear boundaries can be modelled. It transforms the input into dimension d and tries to find the decision boundary in that dimension.

If a decision boundary is not found in that dimension, then the dimension space is increased.

Radial basis kernel:

$$K(x, y) = e^{-\gamma(x-y)^2} \tag{4-2}$$

The equation for the RBF kernel is defined as follows:

Here, x and y are two input data points, ||x - y|| is the distance between the two data points, and γ is a parameter known as the bandwidth or spread. The bandwidth parameter controls the shape of the radial basis function and its value must be chosen carefully to avoid overfitting or underfitting the data. The gamma determines the influence of the individual data points on the decision boundary. Assigning large value to gamma allows fewer data points to influence on the decision boundary thereby causing overfitting.

4.3 Activation Functions Used

We trained our Neural Network using sigmoid as well as ReLU activation functions for hidden layers. The output layer uses SoftMax activation. The algorithm for logistic regression uses sigmoid as well. The activation functions used are described below:

4.3.1 Sigmoid activation

The sigmoid function introduces non-linearity by mapping any input value to an output between 0 and 1. It is defined as:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4-3}$$

Where x is the input to activation function.

The output of the sigmoid function approaches 1 as the input value increases and approaches 0 as the input value decreases. This allows for binary classification as a threshold value can be used to classify between two classes.

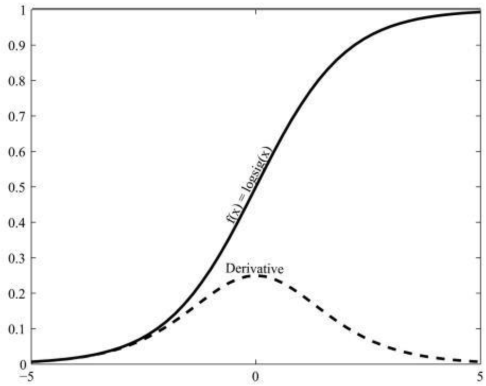


Figure 4-6: Sigmoid Activation Graph

4.3.2 ReLU activation

We have also tested our Neural Network using another commonly used activation function called Rectified Linear Unit (ReLU). It is computationally efficient than sigmoid activation and has been shown to work well for many deep learning tasks. It solves the vanishing gradient problem of sigmoid function. It is defined as:

$$f(x) = \max(0, x) \tag{4-4}$$

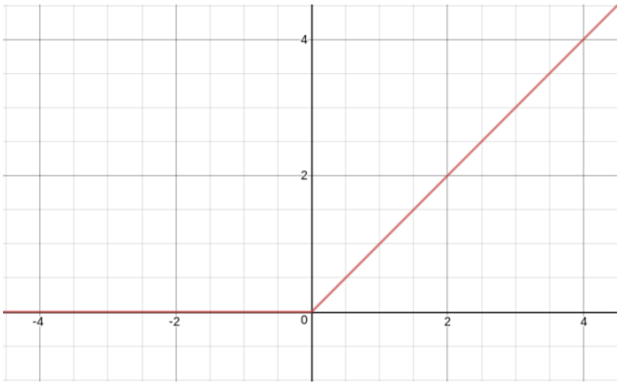


Figure 4-7: ReLu Activation Graph

4.3.3 SoftMax activation

We require SoftMax activation in the output layer of the neural network since we have categorical data. There are 22 different categories/labels in our dataset representing different crops. The SoftMax function generally maps a vector of real numbers to a probability distribution over the classes. It is defined as:

$$f(x) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{xj}} \tag{4-5}$$

Where x is the input vector and x_i is the i^{th} element in the input vector. The summation goes over all the elements of the input vector. The SoftMax function transforms the input values into a set of values that sum up to 1, representing the predicted probability of each class. This allows the network to make predictions based on the most likely class.

4.4 Accuracy

Since our dataset is equally balanced in all categories, we have used accuracy as the evaluation metric. Accuracy is a simple metric which is easy to understand and infer. Accuracy gives the fraction of correct predictions over the total predictions made. It is defined as:

$$Accuracy = \frac{No.\,of\ correct\ predictions}{Total\,predictions} \tag{4-6}$$

4.5 Wireless Sensor Network (WSN)

A wireless sensor network (WSN) is a type of network that consists of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions. The data gathered by the sensors are transmitted wirelessly to a central location for processing. Each device, or node, in the network is equipped with one or more sensors that collect various physical parameters like, temperature, light, pressure, humidity etc. The nodes communicate with each other and with the central device known as the sink or a base station. The base station is connected through the internet to transmit the data. [9]

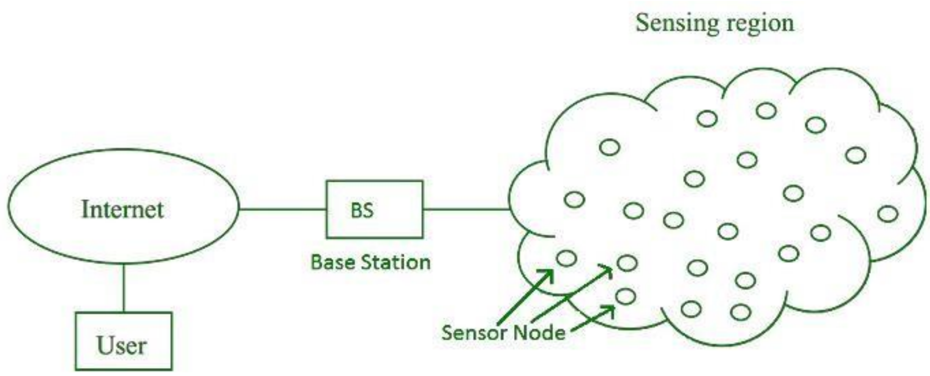


Figure 4-8: An Overview of Wireless Sensor Network [17]

Each sensor comprises of:

Sensing Unit: We use various sensors to monitor number of environmental parameters like temperature, humidity and pH.

Communication Unit: Data transmitted form one sensor node to another and finally to a base station in multi-hop communication. We transmit data from each node using ESP8266 Wi-Fi module of NodeMCU.

Processing Unit: Each sensor node also contains a processing unit for processing of data which is generally of low capacity.

Storage Unit: Each sensor unit comprises of a small storage unit to store some data.

Analog to Digital Converter: It converts the analog reading of various sensors to digital form which are then transmitted.

Power: Each node requires power to function its electrical and electronic components. The NodeMCU runs on 3.3V DC. The input can be provided between 5V to 10V which is regularized using LDO stabilizer in NodeMCU.

4.5.1 Client-Server Architecture

We have implemented client-server architecture for communication between the nodes. The server node is configured with both AP (Access Point) and STA (Station Mode) whereas the client node is configured with only STA mode. When initializing the server in STA-AP mode, it creates its own Wi-Fi network which can be accessed by other nodes and also connects to another network using STA mode to share information to other sources at the same time. While in STA mode the node connects to a certain Wi-Fi SSID specified in it. The server node uses AP mode to fetch soil parameters from all of the nodes connected to it (the client nodes). At the same time the server node is also responsible to send all of the data collected from client nodes to the database server connected over the internet. The master node also sends POST request to the backend server thereby transmitting data from field to the backend where the data is fed into the model and recommendation is generated.

4.5.2 ESP Web server

A web server is a program that listens to incoming requests and responds to those actions accordingly. The ESP8266 module facilitates the use of web server in it to create web pages and communicate data. The web server should listen for incoming HTTP requests and respond correctly. The request can be made to any endpoint in the server and specific data can be retrieved or posted into the server. The HTTP requests can be made from any browser and the server can be accessed.

GET and POST request can be made from the client to the server. A GET request is made from the client to retrieve data from the server. When a URL is entered from a client a GET request is made into the server and server responds with specified data. Whereas a POST request is made from the client in order to push data into the server. For example, when we click submit button in a form a post request is sent into the server with the form inputs.

4.5.3 ESP-NOW

ESP-NOW is a connectionless Wi-Fi communication protocol which was developed by Expressif that enables multiple devices to communicate and share data with each other. it can carry up to 250 bytes of data. the devices communicate with each other by encapsulating the application data in a vendor-specific action frame which is transmitted from one Wi-Fi device to another. The frame contains the MAC address of the device to which the data is being sent to and also the MAC address of the device sending the data. The default ESP-NOW bit rate is 1 Mbps. The pairing between devices is needed before the communication takes place. After pairing the data can be sent and no handshake is required. After the establishment of a connection, if one of the devices loses power or resets it will again connect to its peer to continue the communication.

This protocol has certain limitations such as, when the Wi-Fi is initialized in station mode 10 encrypted peers can be added or when the Wi-Fi is initialized in Access point (soft AP) mode or access point station (soft AP + station) mode at most 6 peers can be connected. This protocol supports one way and two-way communication.

4.6 Flowcharts

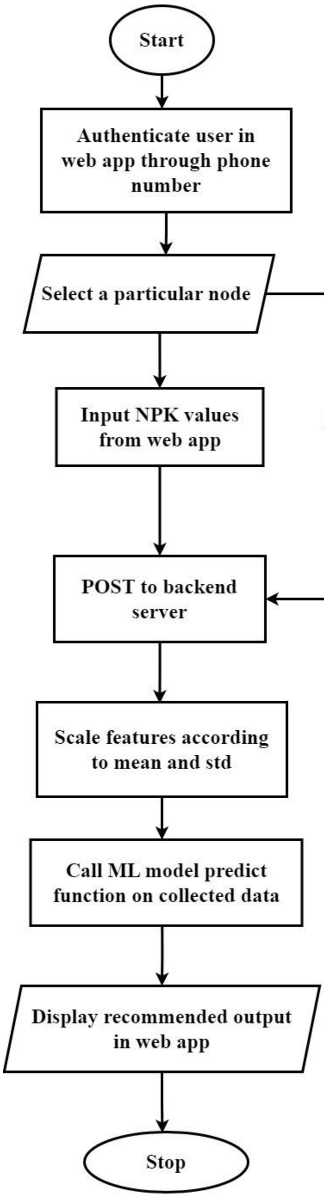


Figure 4-9: Flowchart of proposed system

Firstly, the user is logged into their account using personal phone number in the web app. The nodes connected in their field can be seen and NPK inputs for all the nodes. At the same time, the server node reads data from all the client nodes and the data from the web app, server node is sent into the backend server. The received data is then scaled according to the available mean and standard deviation values for each parameter. This scaled data is fed into the ML model and recommended output is generated. The recommended output is fetched in the web app and displayed appropriately.

5. IMPLEMENTATION DETAILS

We made an IOT device combining various soil sensors such as pH, temperature and humidity in a microcontroller (NodeMCU ESP8266). The pH sensor determines the soil's pH level, which can impact the growth and health of plants.

Likewise, the temperature measures the environment temperature which influences different biological processes such as seed germination, root growth and nutrient uptake. These sensors measure different parameters of the soil which can influence the growth of a plant. The DHT11 sensor is used to measure the humidity and temperature of the atmosphere it sends the calibrated data in binary format to NodeMCU for processing using single wire communication protocol.

This collected information is then sent to a database server through HTTP requests. FastAPI endpoints are implemented to transfer data from NodeMCU to the backend server and vice versa. The data is then stored in a database, pre-processed (feature scaling) and finally fed into the trained machine learning model.

5.1 Pin configuration

The hardware components of the system have been soldered into a matrix board for proper presentation and to reduce the hassle of rearranging all the wires. All the nodes are placed in separate matrix boards for proper identification and easy access to each component. Components such as pH sensor and DHT11 module have also been soldered together.

The DHT11 module has 3 pins which have been connected such that the Vcc pin is connected to 3.3V supply of the nodeMCU module, GND pin is to the G pin of nodeMCU and the DATA pin of the module is connected to pin D1 in nodemcu represented by GPIO5.

The pH sensor also has 3 pins +5V, GND and output pin. As the pH sensor requires +5V output, a voltage regulator 7805 IC has been connected to a 9V battery to provide an output of 5V to the pH sensor. The GND has been connected to the GND of nodeMCU and the output pin to the A0 pin as the output received from the sensor is an analog output. Furthermore, the nodeMCU has also been powered by the 5V output of the regulator.

5.2 Hardware calibration

5.2.1 Calibration of DHT11

The calibration of the DHT sensor was done by comparing the sensor value with the value from the internet. During our calibration and testing, we found that the readings from the sensor were almost close to the internet value (true value). The offset of maximum ±1°C was obtained for temperature value. Similarly, the offset of about ±5% was obtained for humidity. Since, the sensor was providing us with almost correct values most of the time, we didn’t feel the necessity to use any offset value for the DHT sensor. The temperature and humidity readings of the sensor and the corresponding values from the internet at various instant of time is shown in a table below.

Table 5-1 Calibration of DHT11 Sensor

Time	**Sensor Reading**		**Reading from Internet**	
	Temperature (^{0}C)	**Humidity (in %)**	**Temperature (^{0}C)**	**Humidity (in %)**
7:00	13	75	12	80
10:00	22	53	19	58
14:00	22	40	25	36
15:00	23	35	23	45
20:00	18	60	17	64

5.2.2 Calibration of pH sensor

The pH sensor was calibrated using buffer solutions of pH 4, 7, and 10. The following steps were taken during the calibration process:

Step 1: The pH sensor was placed in the pH 7 buffer solution (distilled water), and was allowed to stabilize for 10 minutes. Ten consecutive analog readings were recorded.

Step 2: The ten values recorded for each buffer solution were sorted in ascending order. The six values in the middle were selected, and their average was calculated. The values that were too low or too high were eliminated to eliminate any random values that could interfere with the actual results. The same process was repeated for the pH 4 and pH 10 buffer solutions.

Step 3: The 10 consecutive average analog values were recorded for each pH levels and tabulated as below with their timestamp.

Table 5-2 Analog Reading of pH senor for calibration

Analog value for pH 4		**Analog Value for pH 7**		**Analog Value for pH 10**	
Timestamp	**Analog Reading**	**Timestamp**	**Analog Reading**	**Timestamp**	**Analog Reading**
49:00.0	538	02:31.8	727	30:00.0	921
49:01.0	542	02:33.1	728	30:01.0	921
49:02.0	540	02:34.0	729	30:02.0	922
49:03.0	537	02:35.7	729	30:04.0	922
49:04.0	541	02:37.0	730	30:05.0	922
49:05.0	542	02:38.3	729	30:06.4	923
49:06.0	540	02:39.6	730	30:07.7	923
49:07.0	539	02:40.9	730	30:09.0	924
49:08.0	538	02:42.2	730	30:10.3	924
49:09.0	537	02:43.5	730	30:11.6	924
49:10.0	540	02:44.8	729	30:12.9	924
49:11.0	541	02:46.1	729	30:14.2	924
49:12.0	542	02:47.4	728	30:15.5	924
49:13.0	544	02:48.7	727	30:16.8	924
49:14.0	536	02:50.0	727	30:18.0	924
49:15.0	540	02:51.3	726	30:19.0	924
49:16.0	541	02:52.6	727	30:21.0	925
49:17.0	539	02:53.9	726	30:22.0	926
49:18.0	538	02:55.2	726	30:23.4	927
49:19.0	542	02:56.5	727	30:24.7	926
Average Value	**539.6**	**Average Value**	**728.2**	**Average Value**	**923.7**

The average analog reading value of the pH solution was plotted in a graph with respective pH level. The slope and y-intercept of the graph was noted which is used for the calibration of the pH sensor.

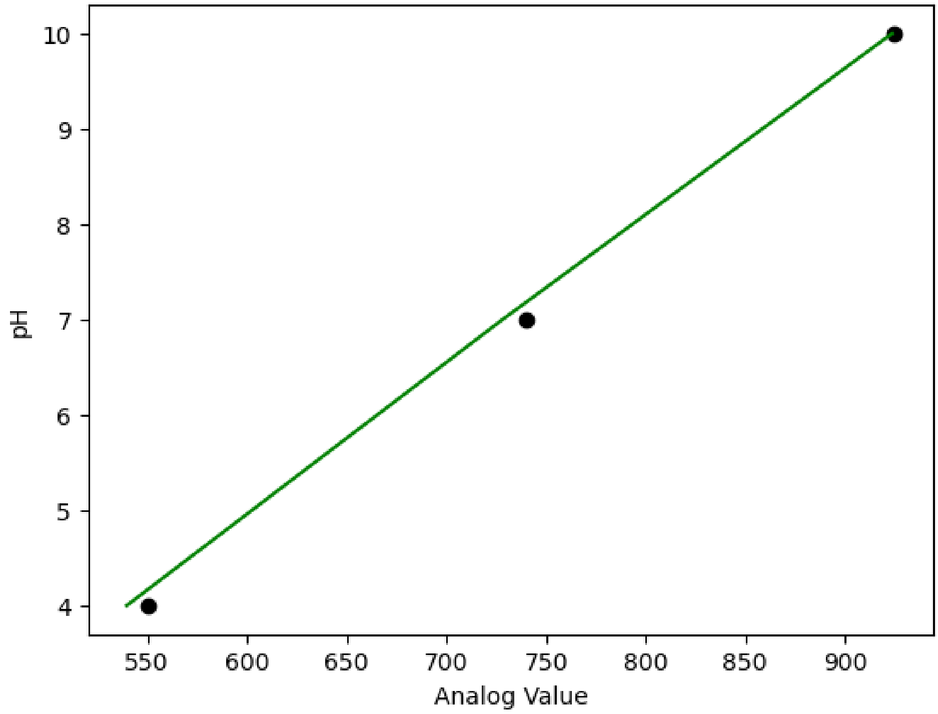


Figure 5-1 Graph of analog reading vs pH of buffer solution

The values obtained from the above tables were plotted and a line was obtained for the best fit. The slope of the line was 65.11 and the y-intercept of the line was 272.52.

5.3 Data Collection

Every machine learning algorithm requires data to produce the output. The dataset collected and used to train the model should be of high reliability because machine learning models become stronger or robust when the dataset used to train those models is legit. We collected our dataset from an online community of data scientists and machine learning practitioners, Kaggle. Kaggle is a subsidiary of Google LLC which holds thousands of datasets. We used the crop recommendation dataset from Kaggle which consists of 2200 datapoints divided into 22 categories/labels. Each datapoint consists of 7 features which are: N, P, K, pH, humidity, temperature, and rainfall. We didn’t use rainfall and used only remaining 6 parameters to train out model.

5.4 Dataset Exploration

The dataset we used for training is evenly balanced in all categories. Each label consists of 100 data points.

Table 5-3: Crops with their available datapoints

Label	Data Points
rice	100
maize	100
jute	100
cotton	100
coconut	100
papaya	100
orange	100
apple	100
muskmelon	100
watermelon	100
grapes	100
mango	100
banana	100
pomegranate	100
lentil	100
blackgram	100
mungbean	100
mothbeans	100
pigeonpeas	100
kidneypeas	100
chickpea	100
coffee	100

Here figure 5-1 shows a small example of how the dataset looks like.

	N	P	K	temperature	humidity	ph	label
1584	28	130	196	22.134506	94.676957	6.062356	15
571	2	56	23	26.653330	59.790234	7.550091	5
1319	103	17	51	25.111892	80.026213	6.209888	13
79	81	41	38	22.678461	83.728744	7.524080	0
1266	40	121	199	26.181597	81.038863	6.315586	12

Figure 5-2: Sample of Dataset

The label is stored as 'y' to be predicted using the values available I.e., 'X' consisting of 6 features (N, P, K, Temperature, humidity and pH).

We also plotted correlation heatmap to visualize correlation among the features. From figure 5.2 we can infer that no two features in our dataset are highly correlated. We can see that the highest correlation is between the features, P and K with the correlation coefficient of 0.74. But with our analysis and model training we concluded to not drop any of the features.

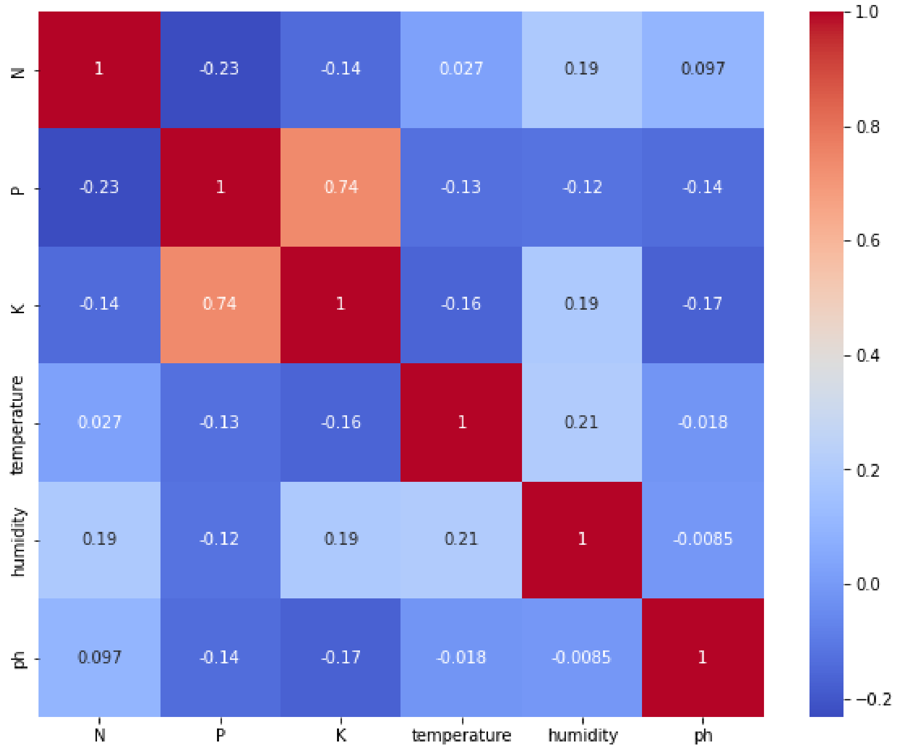


Figure 5-3: Correlation heatmap of the features

5.5 Data Pre-processing

5.5.1 Data Encoding

The label column in the dataset is encoded using One hot encoding. It is the method by which categorical variables are converted into numeric form that can be understood by the ML model. It involves converting each unique category value in a categorical variable into a new binary column, where each row has a 1 in the column corresponding to the category it belongs to and 0s in all other columns. This gives us a sparse matrix where most entries are 0 and only a single entry for a row is 1.

For a single example we have 22 classes so a matrix Y [1, 22] is constructed and value 1 is placed at the column of the label e.g., for categorical value represented by 2 the element at location Y [1,3] = 1.

	0	1	2	3	4	5	6	7	8	9	...	12	13	14	15	16	17	18	19	20	21
apple	0	0	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	0	0	0
banana	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
blackgram	0	0	0	0	0	0	0	1	0	0	...	0	0	0	0	0	0	0	0	0	0
chickpea	0	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
coconut	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	0	0	0
coffee	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
cotton	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	0	0
grapes	0	0	0	0	0	0	0	0	0	0	...	1	0	0	0	0	0	0	0	0	0
jute	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	1	0
kidneybeans	0	0	0	1	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
lentil	0	0	0	0	0	0	0	0	1	0	...	0	0	0	0	0	0	0	0	0	0
maize	0	1	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
mango	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
mothbeans	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
mungbean	0	0	0	0	0	0	1	0	0	0	...	0	0	0	0	0	0	0	0	0	0
muskmelon	0	0	0	0	0	0	0	0	0	0	...	0	0	1	0	0	0	0	0	0	0
orange	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	1	0	0	0	0	0
papaya	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	1	0	0	0	0
pigeonpeas	0	0	0	0	1	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
pomegranate	0	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	0	0	0	0	0
rice	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
watermelon	0	0	0	0	0	0	0	0	0	0	...	0	1	0	0	0	0	0	0	0	0

Figure 5-4: One-Hot Encoding

5.5.2 Filling Missing values

It refers to the process of filling missing or null values in the dataset to make it suitable for the ML model. This is an important step in the data pre-processing phase, as missing values can cause problems with model accuracy and performance.

There are various imputation techniques to fill out the missing values in a dataset. One of the methods with high reliability is the Multiple Imputation by Chained Equations (MICE) algorithm. It generates multiple imputed datasets and combines the result to get more accurate estimate of the missing values. It is an iterative process which works by iteratively imputing missing values based on available values in the dataset. The available values are fed to learning models like, linear regression, logistic regression to predict the missing values. As there aren't any missing values in our dataset, this step was not required in the training process.

5.5.3 Feature Scaling

It refers to the technique which is used to normalize the values in a dataset. This step is very important as neural network is sensitive to the scale of the input. If the features are on a different scale, some features with high value may dominate the distance calculation and prevent algorithm from generalizing properly. Feature scaling solves this problem by bringing all the features to a similar range.

The method we used for feature scaling is Standardization. This method scales the values of a feature so that they have a mean of 0 and a standard deviation of 1.

$$z_{ij} = \frac{(x_{ij} - \mu)}{\sigma} \tag{5-1}$$

Here, μ is the mean and σ denotes the standard deviation of a particular feature.

5.6 Training

The model training was carried out using different algorithms like: Support Vector Machine, Logistic Regression and Neural Network. The dataset collection and data pre-processing jobs were carried out. Then dataset was separated into train and test set not overlapping with each other. The ratio of train to test is 80:20 and the distribution of the test set is maintained. The model is iterated and refined using the train set and real-world accuracy simulation is done using the test set.

Moreover, the test set is used for hyperparameter tuning. It is the process of choosing correct values of variables such that the accuracy increases. We did tuning of hyperparameter like: learning rate, activation function, number of epochs and types of kernels in SVM.

The neural network consists of two hidden layers of 50 units and 30 units with 7 input units and 22 output units. It was trained for 3000 epochs with learning rate of 0.5. No data augmentation steps were carried out during training. Similarly, the SVM was trained for 50 epochs with RBF kernel. The regularization parameter was set to 1.

5.7 Inferencing

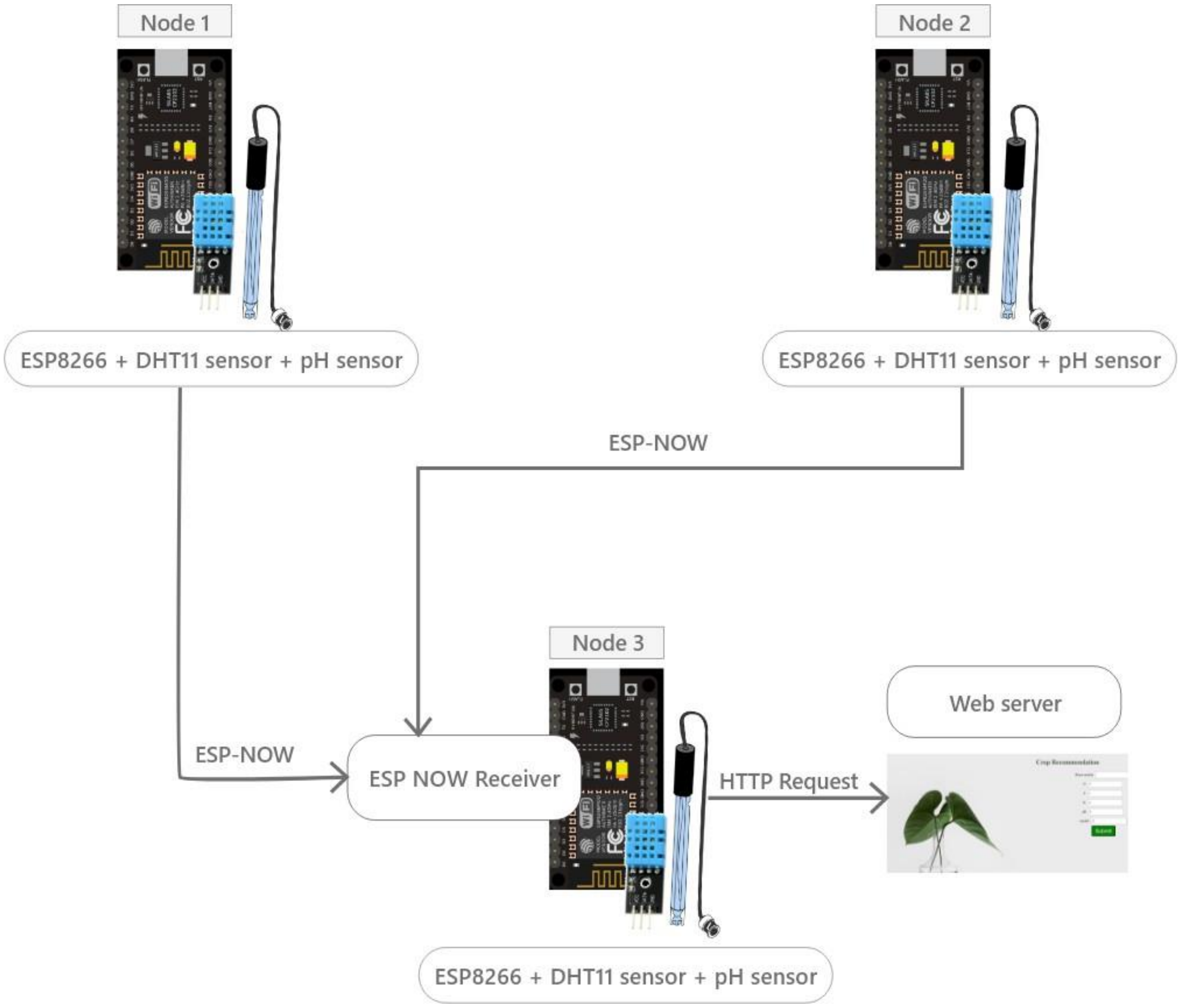


Figure: Client-Server Architecture

During real time testing, the data from fields are collected and are then fed to our trained machine learning model to generate the recommendation. Number of nodes are deployed in the field, and we have implemented client-server architecture to communicate between the nodes.

Here, each node consists of pH, humidity and temperature sensors and is controlled through NodeMCU. The sensor reading is being taken by every node. A server node consists of multiple client nodes. Each client sends its sensor readings to its own server node. The client sends its node ID, temperature, humidity and pH readings to the server. This communication between client and server is maintained by taking the advantage of ESP-NOW which uses the MAC address of the other node to create the communication. This way the server node gets sensor readings from all of its client nodes. Now, it is the job of the server node to send the sensor readings to the backend server. The server node serializes all the collected data from its clients into a JSON object and then sends the data to the backend server. It does this through a HTTP post request.

The backend server contains a pickle file of our trained Machine Learning model and also contains a function to predict. The prediction function takes all the soil parameters like, N, P, K, temperature, humidity, and pH and then generates the recommendation based on those values. The backend server also contains a SQL database which maps the user's phone number with his/her nodes with the help of node ID obtained from each node. This is important as a user is required to enter his/her phone number as a frontend client to generate his/her crop recommendation.

We also have a frontend server of ReactJS. Users upon their desire to see the crop recommendation for their field can connect to our frontend server from a web browser. The users are then asked to enter their phone number. The phone number is then sent to the backend server through HTTP post request. The users can then provide the N, P and K values of their field. (N, P and K values are taken from the frontend for now due to unavailability of the NPK sensor). Upon submitting those values to the backend, the backend server then fetches other parameters like, temperature, humidity, and pH from the node from the field. Then the crop recommendation is done using all those parameters as mentioned above. Finally, the backend sends the recommended crop as a response to the frontend.

6. RESULTS AND ANALYSIS

6.1 ML model Analysis

Three ML models were trained for this purpose which are logistic regression, neural network and SVM. The results from these models are listed below:

6.1.1 Logistic regression:

Logistic regression was implemented for 22 classes by using one vs all method utilizing L2 regularization but only 81% accuracy was obtained, so it was discarded at an early stage.

6.1.2 Neural Network:

Neural network was implemented from scratch by the use of python libraries such as NumPy. The codes for forward propagation, backward propagation, accuracy calculation was written all by us. Gradient descent was used as optimization method. Then hyperparameter tuning was performed.

Table 6-1: Neural Network Accuracy Comparision

Layers	**Train accuracy ‘Relu’**	**Test accuracy ‘Relu’**	**Train accuracy ‘sigmoid’**	**Test accuracy ‘sigmoid’**
2	0.926	0.925	0.862	0.856
3	0.936	0.928	0.216	0.213
4	0.219	0.232	0.045	0.045
Epoch = 3000				

The above table shows the accuracy of different layered neural networks using ReLu and sigmoid as activation functions in all other layers and SoftMax in the last layer. This data is collected at 3000 epochs as in this range the accuracy doesn't change by much.

Sigmoid performs well in 1 hidden, 1 output layer architecture but increasing the hidden layers drops the accuracy drastically.

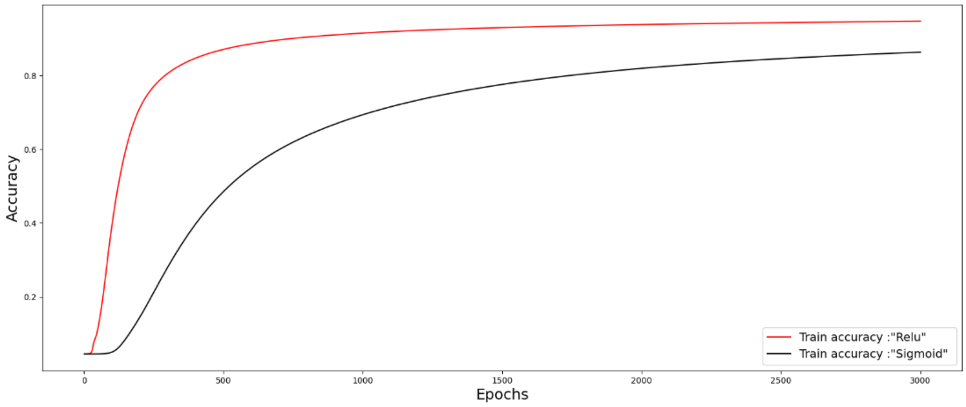


Figure 6-1: Accuracy of 2-layer Neural Network

From figure 6.2, it can be seen that in 2 layers using ReLu activation function performs much better which can also be verified from the above table. The sigmoid activation doesn't perform as well as ReLu from the beginning and cannot catch up as the epoch progresses forward.

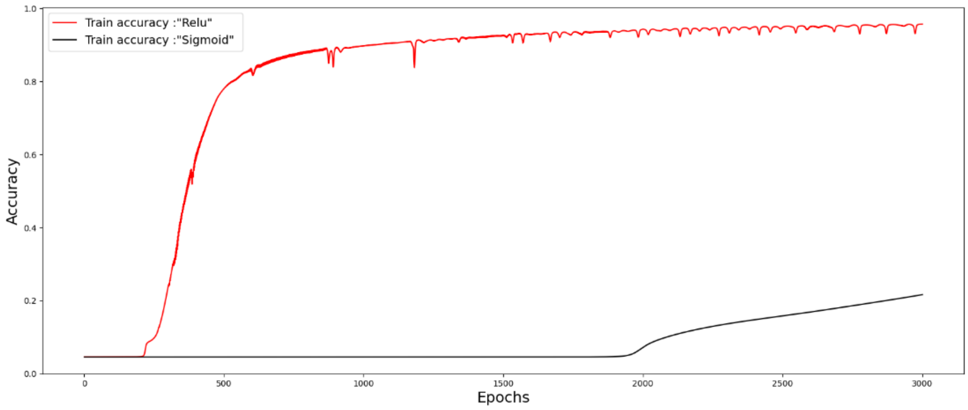


Figure 6-2: Accuracy of 3-layer Neural Network

it gave the best accuracy at 3000 epochs. It can also be seen that the sigmoid activation function performs poorly with 1 output and 2 hidden layers of 40 and 30 neurons.

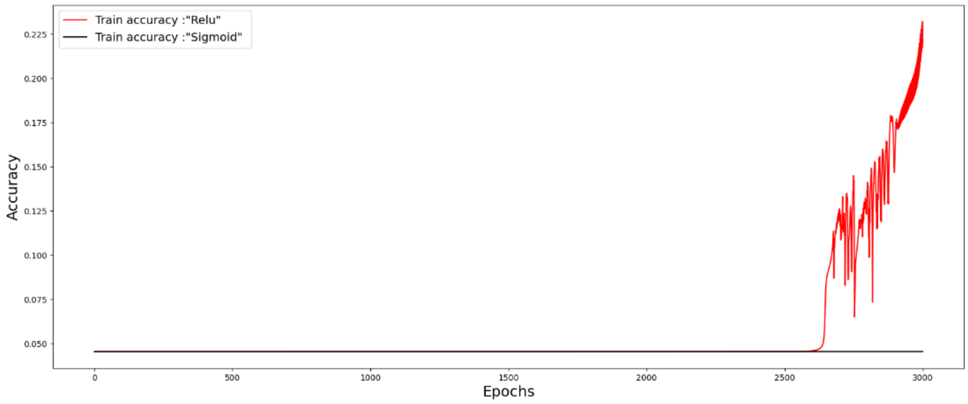


Figure 6-3: Accuracy of 4-layer Neural Network

The above figure shows the accuracy comparison in 4-layer neural network. It shows that using ReLu activation accuracy starts to increase at around 1900 epochs and using sigmoid activation the accuracy doesn't increase even till 3000 epochs.

6.1.3 SVM:

An SVM model was trained for 50 epochs. The train and test accuracies were noted for each epoch. The test accuracy started to increase as the training progressed. The highest test accuracy was obtained at 40th epoch then the accuracy started to decrease gradually. The highest train and test accuracies were obtained as 95.17% and 94.09% respectively. Different kernels were tested in this approach and the kernel which gave the best accuracy was selected I.e., RBF kernel.

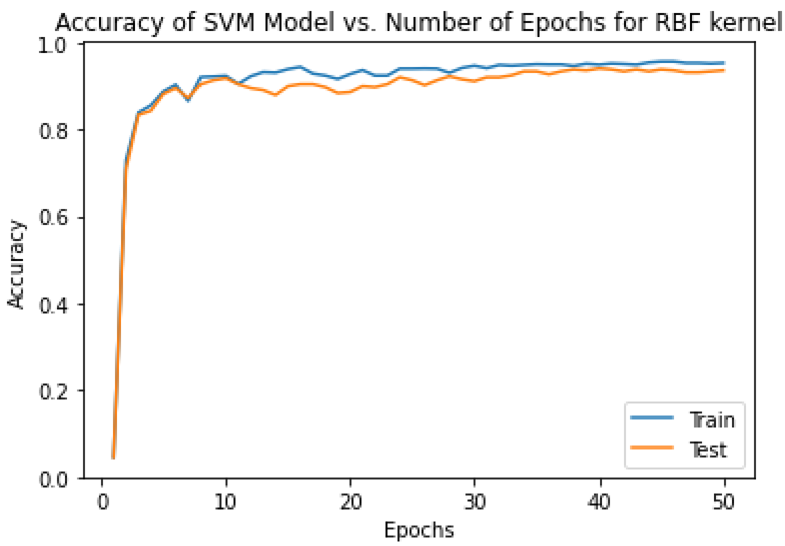


Figure 6-4: SVM model RBF kernel Accuracy vs epochs graph

Using the RBF kernel, train and test accuracy were seen to be similar from the beginning and training accuracy reached saturation very early at around 10 epochs.

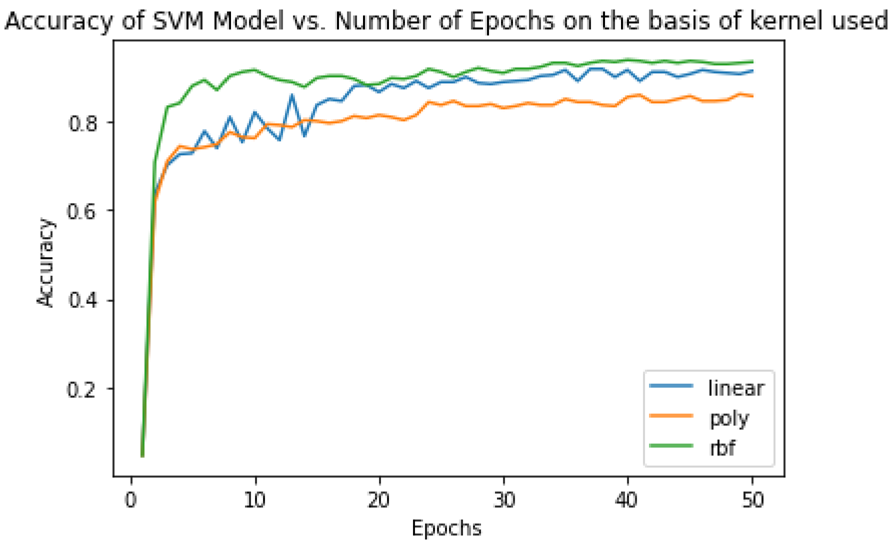


Figure 6-5: Accuracy vs Epochs graphs for different kernels of SVM

The above figure 6-6 shows the accuracy of various kernel RBF, polynomial and linear at different epochs. It is seen that linear and RBF kernel perform similar, but accuracy of linear kernel fluctuates whereas RBF gives a steady output. The polynomial kernel has poor performance as compared to the other two kernel.

Table 6-2: ML model accuracy comparision

Models	**Train accuracy**	**Test accuracy**
Logistic regression	0.826	0.8145
3-layer Neural network	0.936	0.9282828
SVM	0.951	0.940

The above table shows the overall performance statistics of different models trained using the same dataset. SVM with RBF kernel gives the best accuracy. Thus, SVM was selected as ML model.

6.2 Hardware Analysis:

NodeMCU is a low-cost, open-source development board that can be used to collect data from sensors for NPK, temperature, humidity, and soil pH. The data collected from these sensors can be used as input for the crop prediction system. However, the NPK sensor we have was giving fluctuating and inaccurate value also it is difficult to manage another NPK sensor in Nepali market. Due to this reason the NPK value will be obtained through a chemical process. The collected data must be transmitted to the prediction system, which can be done wirelessly using the inbuilt ESP8266 Wi-Fi module.

The DHT11 temperature and humidity sensor is a crucial component in a crop prediction as it measures the environmental conditions that affect crop growth. It is calibrated in the laboratory that is extremely accurate (±5 %) on humidity calibration. The calibration coefficients are stored as programs in the OTP memory, which are used by the sensor's internal signal detecting process. It can further be calibrated by comparing it with the reference value measured using standard devices. The DHT11 sensor uses a single-wire communication protocol, specifically the Single-Wire Digital Interface (One-Wire Protocol). The NodeMCU sends a start signal to the DHT11, and the DHT11 responds with the humidity and temperature readings, encoded as a sequence of bits.

The soil pH sensor contains the probe electrode that can measure the Soil pH value from 3 to 9 with high accuracy up ±0.3PH. The sensor has an IP68 protective case & is sealed with High-density epoxy resin which can prevent moisture from entering the body interior part. The sensor may require 5-10 minutes for stabilization after poweringon. The Soil Ph Sensor has 4 pins as it needs to be connected to RS485 or MAX485 Module. The communication protocol that it uses is Modbus. The device works as a client. The NodeMCU sends the inquiry frame, and the sensor sends the response frameform which we can calculate the pH value.

6.3 Networking Analysis

The network between the client and server was tried out with the help of ESP-NOW protocol and ESP8266 Web Server. As the architecture of the network is many-to-one various difficulties were faced during the testing period and the better option was then selected to move forward. The detailed description of the findings is given below.

6.3.1 ESP8266WebServer

A web server was created on port 80 in server node and client nodes sent the request to '/feed' endpoint in the server. The data from various sensors was being read by the client node and at the end being sent to the server node by POST request. The communication took place successfully in case of one server one client node, but when more client nodes were added to the network the data could not be received in the server node at the same instance. The data received was inconsistent and the POST request by client nodes could not be captured by the server. At an instance only one request from the client could be processed by the server due to this the data was inconsistent. Due to this reason, this approach was discarded, and a new approach was initiated.

6.3.2 ESP-NOW

ESP-NOW protocol was implemented for the communication between server and multiple client nodes. The client initialized ESP-NOW role as a SLAVE and the server initiates ESP-NOW role as COMBO to receive and transmit data to-and-fro. The client nodes stored the MAC address of server node and established the peer-to-peer connection on startup. The data from client nodes is sent to the MAC address stored in their memory then the server node calls a call-back function upon receiving data from client nodes and stores that data into a new STRUCT variable. This variable is then posted into hosted backend server.

6.4 System analysis

The goal of the website is to help farmers make informed decisions about which crops to cultivate based on parameters such as soil type, temperature, NPK, rainfall, and pH value of the soil. The UI of the crop recommendation website is designed with usability in mind. It is easy for users to navigate and understand, with clear labels and intuitive controls for entering soil parameters. The user will be provided a page to login into their account with the help of their phone numbers. As one user can have multiple nodes, the phone number of the user acts as a unique identification. Then after logging in, a page is provided which consists of multiple form inputs to enter NPK values for various nodes belonging to the user. The user can select only the nodes for which they want the prediction for while all the other nodes will remain disabled for input if they are not selected by the user. This approach had to be taken as the output obtained from the NPK sensor was not reliable and had been giving fluctuating values.

Once the user has entered the required parameters, the website will provide a clear and concise recommendation for which crop to cultivate for the various nodes that are selected. The recommendation is based on a multi-label classification model that considers the composition of soil and its characteristics. By keeping these factors in mind, we can help ensure that our crop recommendation website is effective and useful for farmers looking to make informed decisions about crop cultivation.

7. Future Enhancements

Connection of non-Wi-Fi zone (Farm) to Wi-Fi zone (House): Analysis and designing of the client-server architecture for large fields to include LoRa communication between the field device (where internet is not available) and the control station.

Smart farm care: We can include number of other functionalities like, water level detection and smart irrigation with help of moisture sensor. Also, we can develop a system to identify crop diseases. Similarly monitoring several soil parameters like N, P, K we can determine the amount of fertilizers required for the proper growth of the plants.

Intrusion detection: We can deploy camera to develop an alert system upon intrusion of animals. We can train CNN model to identify the intruder.

Feedback system: We can extend this project further by developing a feedback system and use a Machine Learning model to take actions according to the feedback. Users can provide their feedback about our model's prediction. We can adjust our model parameters based on the feedbacks provided.

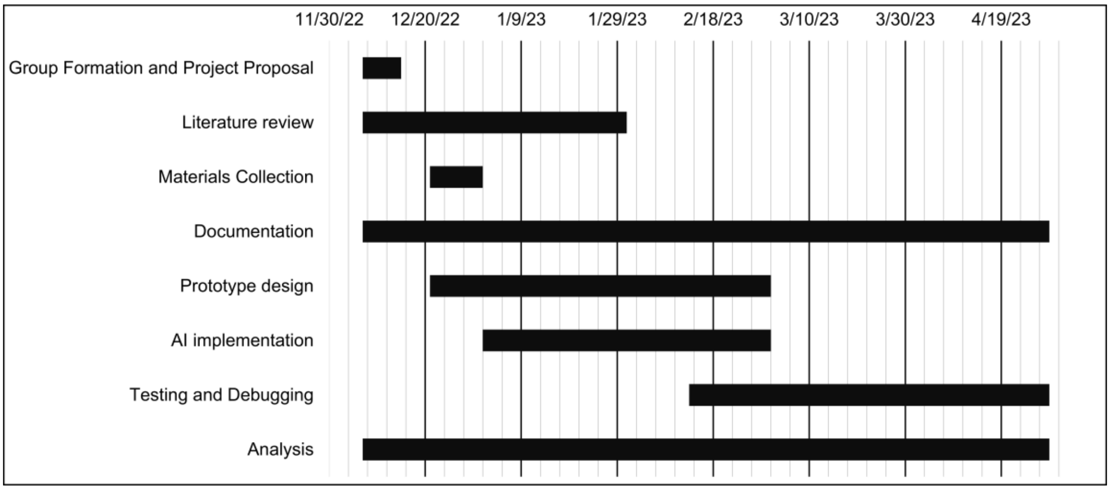
8. Conclusion

We aimed to develop an IoT device for crop recommendation, with the goal of improving agricultural practices and sustainability. Using various sensors to measure temperature, humidity, pH of soil and different machine learning algorithms, we successfully developed a prototype device that can recommend crops based on the NPK, pH value of soil, temperature, and humidity of environment. Our experiments and evaluations show that the device can accurately predict the best crops to plant in a given area and has the potential to increase crop yield and quality while minimizing resource consumption. There are limitations to our project, such as less reliable dataset, more accurate sensors to measure soil parameters. We believe that our device represents a promising step towards more efficient and sustainable farming practices. Future work can focus on expanding the device's capabilities and integrating it with other agricultural technologies to create a more holistic and effective farming ecosystem.

9. APPENDICES

Appendix A: PROJECT SCHEDULE

Table 9-1: Project Timeline



Appendix B: PROJECT BUDGET

Table 9-2: Project Budget

S.No.	Components	Quantity	Unit Cost	Price (Rs.)
1	NodeMCU	3	750	2250
2	pH Sensor Module	1	2500	2500
3	pH Sensor Probe	1	2000	2000
4	DHT11 sensor module	3	280	840
		Total		**7590**

Appendix C: Hardware Block Diagram

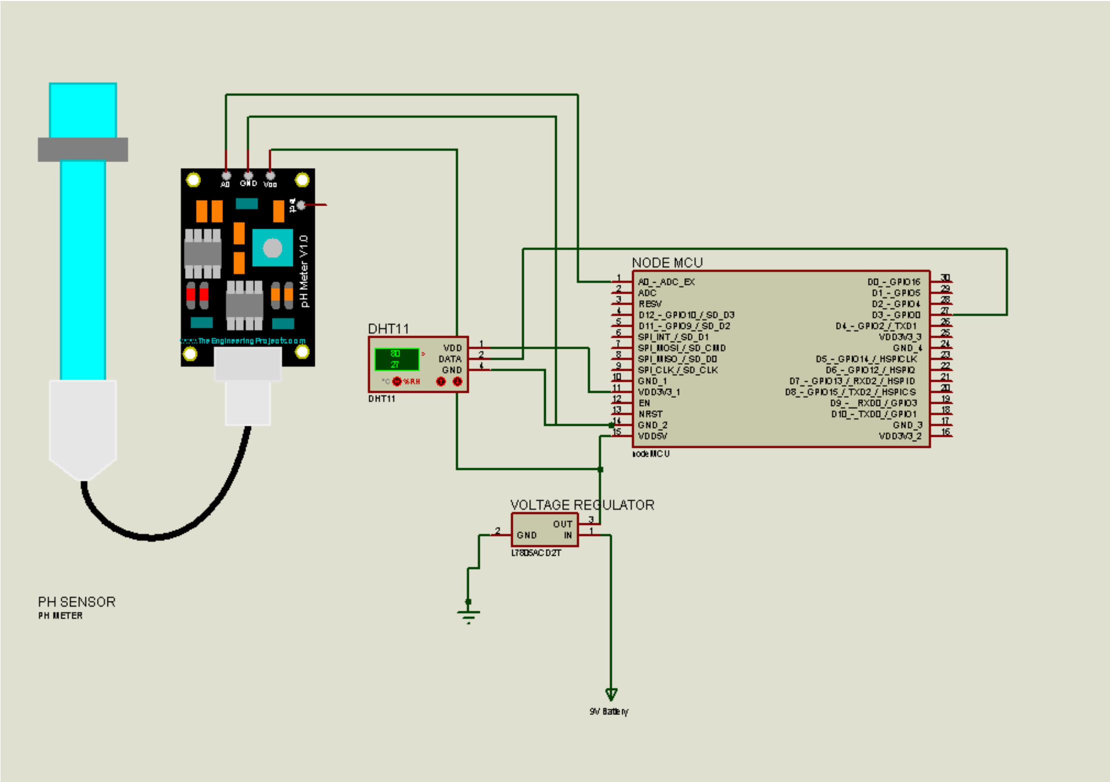


Figure 9-1: Hardware Block Diagram

Appendix D: Data Normalization Code

```
In [4]: X_train,X_test,y_train,y_test = train_test_split(X,Y,stratify=Y,random_state=1,test_size=0.2)

        mean = np.mean(X_train,axis=0,keepdims = True)
        std = np.std(X,axis = 0, keepdims=True)

        X_train = (X_train-mean)/std
        X_test = (X_test-mean)/std

        output_classes = 22
```

Figure 9-2: Python code for data normalization

Appendix E: Logistic Regression Code

```
for iter in range(10000):
    start = time.time()
    Z = np.dot(X_train,W)+B
    Y_pred = sig(Z)
    loss = Y_pred - Y1
    dW = (1/r)*(np.dot(X_train.T,loss)+lamb*W)
    dB = (1/r)*np.sum(loss)
    W = W - alpha * dW
    B = B - alpha * dB
    end = time.time()
```

Figure 9-3: Python code for Logistic Regression

Appendix F: Neural Network Code

```
In [16]: import time
         start = time.time()
         classification1 = NN(3,[7,50,30,output_classes])
         classification1.fit(X_train = X_train.T,Y_train = temp,X_test = X_test.T,
                             Y_test = temp_test,eph=1000,alp=0.9,activation = 'relu')
         classification1.plot_acc(region='test')
         classification1.plot_acc(region='train')
         end = time.time()
         print(end-start)

         train cost = -0.03010398808163289, test cost = -0.03131311585956855,
         train_acc = [0.97912567], test_acc = [0.97688653]
```

Figure 9-4: Python code for Neural Network

Appendix G: SVM Code

```
num_epochs = 100
train_accuracies = []
test_accuracies = []

clf = SVC(max_iter=num_epochs)
clf.fit(X_train, y_train)

# Predicting the labels for the training and testing sets
y_train_pred = clf.predict(X_train)
y_test_pred = clf.predict(X_test)

# Calculating the accuracy for the training and testing sets
train_accuracy = accuracy_score(y_train, y_train_pred)
test_accuracy = accuracy_score(y_test, y_test_pred)

train_accuracies.append(train_accuracy)
test_accuracies.append(test_accuracy)
```

Figure 8-5: Python code for SVM

Appendix H: Plagiarism Report

IoT Device for Crop Recommendation using AI

ORIGINALITY REPORT

17%

SIMILARITY INDEX

PRIMARY SOURCES

1	www.researchgate.net Internet	372 words — 3%
2	www.coursehero.com Internet	196 words — 1%
3	www.mdpi.com Internet	96 words — 1%
4	how2electronics.com Internet	72 words — 1%
5	docplayer.net Internet	67 words — < 1%
6	ijsrst.com Internet	67 words — < 1%
7	docs.espressif.com Internet	52 words — < 1%
8	dora.dmu.ac.uk Internet	51 words — < 1%
9	openaccess.uoc.edu Internet	51 words — < 1%
10	mgnirsa.ac.in Internet	

50 words — < 1%

11	www.geeksforgeeks.org Internet	46 words — < 1%
12	V. Jyothi, D. NagaJyothi, M. Srilatha. "E-Campus using Super Sensor Nodes", 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 2021 Crossref	39 words — < 1%
13	mafiadoc.com Internet	36 words — < 1%
14	www.politesi.polimi.it Internet	36 words — < 1%
15	repository.sustech.edu Internet	33 words — < 1%
16	www.ijeast.com Internet	31 words — < 1%
17	Showkat Ahmad Bhat, Imtiyaz Hussain, Nen-Fu Huang. "Soil suitability classification for crop selection in precision agriculture using GBRT-based hybrid DNN surrogate models", Ecological Informatics, 2023 Crossref	30 words — < 1%
18	www.science.gov Internet	27 words — < 1%
19	István Matijevics. "Chapter 29 Wireless Sensors Networks – Theory and Practice", Springer Science and Business Media LLC, 2009 Crossref	26 words — < 1%

20 www.jetir.org
Internet
22 words — < 1%

21 www1.vnua.edu.vn
Internet
22 words — < 1%

22 Hiroya Ikeda, Hiroki Yamane, Yuta Takishita, Mutsumi Kimura, Yasuhiko Nakashima. "Influence of characteristic variation of oxide semiconductor and comparison of the activation function in neuromorphic hardware", Nonlinear Theory and Its Applications, IEICE, 2020
Crossref
21 words — < 1%

23 Wen-Tsai Sung, Ihzany Vilia Devi, Sung-Jung Hsiao. "Smart Lamp Using Google Firebase as Realtime Database", Intelligent Automation & Soft Computing, 2022
Crossref
21 words — < 1%

24 ujcontent.uj.ac.za
Internet
21 words — < 1%

25 www.analyticsvidhya.com
Internet
21 words — < 1%

26 Mohammad Shahin, F. Frank Chen, Ali Hosseinzadeh, Neda Zand. "Using Machine Learning and Deep Learning Algorithms for Downtime Minimization in Manufacturing Systems: An Early Failure Detection Diagnostic Service", Research Square Platform LLC, 2023
Crossref Posted Content
20 words — < 1%

27 elibrary.tucl.edu.np
Internet
20 words — < 1%

28	P. Kanaga Priya, N. Yuvaraj. "An IoT Based Gradient Descent Approach for Precision Crop Suggestion using MLP", Journal of Physics: Conference Series, 2019 Crossref	19 words — < 1%
29	personalpages.manchester.ac.uk Internet	19 words — < 1%
30	repository.ju.edu.et Internet	19 words — < 1%
31	uu.diva-portal.org Internet	18 words — < 1%
32	Soumya Roy, Yuvika Vatsa, Moumita Sahoo, Somak Karan. "chapter 13 Machine Learning-Based Algorithms Towards Crop Recommendation Systems", IGI Global, 2023 Crossref	17 words — < 1%
33	Yan Long, Alexander Curtiss, Sara Rampazzi, Josiah Hester, Kevin Fu. "VeriMask", Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2021 Crossref	17 words — < 1%
34	kth.diva-portal.org Internet	17 words — < 1%
35	www.codingdict.com Internet	17 words — < 1%
36	dyuthi.cusat.ac.in Internet	16 words — < 1%
37	ore.exeter.ac.uk	

Internet

16 words — < 1%

38 Mohamed Bouni, Badr Hssina, Khadija Douzi, Samira Douzi. "Towards an Efficient Recommender Systems in Smart Agriculture: A deep reinforcement learning approach", Procedia Computer Science, 2022
Crossref

15 words — < 1%

39 researchonline.federation.edu.au
Internet

15 words — < 1%

40 www.publishoa.com
Internet

15 words — < 1%

41 nanopdf.com
Internet

14 words — < 1%

42 repository.untag-sby.ac.id
Internet

14 words — < 1%

43 Abhishek Silwal, Abhishek Ghimire, Kapalik Khanal, Umesh Kanta Ghimire. "A Model of Cube-Sat", Journal of Innovations in Engineering Education, 2020
Crossref

13 words — < 1%

44 Rohini Jadhav, Pawan Bhaladhare. "Farmer's Assistant in Agricultural Sector by using Machine Learning and Deep Learning", 2022 OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON), 2023
Crossref

13 words — < 1%

45 flipkarma.com
Internet

13 words — < 1%

46	link.springer.com Internet	13 words — < 1%
47	www.udemy.com Internet	13 words — < 1%
48	M.R. Nithya, P. Lakshmi, J. Roshmi, R. Sabana, R.Udhaya Swetha. "Machine Learning and IoT based Seed Suggestion: To Increase Agriculture Harvesting and Development", 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 2023 Crossref	12 words — < 1%
49	Zhang, Zhilu. "Reliable Deep Learning with Application to Digital Histopathology Image Analysis", Cornell University, 2022 ProQuest	12 words — < 1%
50	www.diva-portal.se Internet	12 words — < 1%
51	"Software Engineering Perspectives in Intelligent Systems", Springer Science and Business Media LLC, 2020 Crossref	11 words — < 1%
52	arxiv.org Internet	11 words — < 1%
53	digital.library.unt.edu Internet	11 words — < 1%
54	downloads.hindawi.com Internet	11 words — < 1%
55	dspace.bracu.ac.bd Internet	11 words — < 1%

56	dspace.daffodilvarsity.edu.bd:8080 Internet	11 words — < 1%
57	dspace.univ-adrar.edu.dz Internet	11 words — < 1%
58	ijeecs.iaescore.com Internet	11 words — < 1%
59	ijsrset.com Internet	11 words — < 1%
60	sersc.org Internet	11 words — < 1%
61	Oliveira, Tiago Emanuel da Silva. "Small Antennas For 5G and IoT", Instituto Politecnico de Leiria (Portugal), 2023 ProQuest	10 words — < 1%
62	Syeda Iqra Hassan, Muhammad Mansoor Alam, Usman Illahi, Mohammed A. Al Ghamdi, Sultan H. Almotiri, Mazliham Mohd Su'ud. "A Systematic Review on Monitoring and Advanced Control Strategies in Smart Agriculture", IEEE Access, 2021 Crossref	10 words — < 1%
63	iotbasedcontrollinglightusingnodemcu.blogspot.com Internet	10 words — < 1%
64	www.ijnrd.org Internet	10 words — < 1%
65	Bhuvaneswari Swaminathan, Saravanan Palani, Ketan Kotecha, Vinay Kumar, Subramaniyaswamy	9 words — < 1%

V. "IoT Driven Artificial Intelligence Technique for Fertilizer Recommendation Model", IEEE Consumer Electronics Magazine, 2022
Crossref

66 G Sai Pravallika, L. Kundana, K. Sri Thanvi, G. Sirisha, Ch. Rupa. "Proficient Smart Soil based IoT System for Crop Prediction", 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020 — 9 words — < 1%
Crossref

67 Jon Ander Gómez, Juan Arévalo, Roberto Paredes, Jordi Nin. "End-to-end neural network architecture for fraud scoring in card payments", Pattern Recognition Letters, 2018 — 9 words — < 1%
Crossref

68 apessay.elementfx.com — 9 words — < 1%
Internet

69 dspace5.zcu.cz — 9 words — < 1%
Internet

70 help.highbond.com — 9 words — < 1%
Internet

71 nozdr.ru — 9 words — < 1%
Internet

72 uir.unisa.ac.za — 9 words — < 1%
Internet

73 www.slideshare.net — 9 words — < 1%
Internet

74 "Proceedings of Third International Conference on Intelligent Computing, Information and Control Systems", Springer Science and Business Media LLC, 2022 — 8 words — < 1%
Crossref

75 "Proceedings of the 2nd International Conference on Green Energy, Environment and Sustainable Development (GEESD2021)", IOS Press, 2021 — 8 words — < 1%
Crossref

76 B Shabari Shedthi, Vidyasagar Shetty, Anusha, Rakshitha R Shetty, Anisha Shetty, B.A Divyashree Alva. "Crop and Nutrient Recommendation System using Machine Learning for Precision Agriculture", 2022 International Conference on Artificial Intelligence and Data Engineering (AIDE), 2022 — 8 words — < 1%
Crossref

77 Chun-Yi Liu, Cheng-Long Chuang, Chia-Pang Chen, Wan-Yi Chang, Jyh-Cheng Shieh, Cheng-Han Lin, Chwan-Lu Tseng, Joe-Air Jiang. "Development of an embedded system-based gateway for environmental monitoring using wireless sensor network technology", 2011 Fifth International Conference on Sensing Technology, 2011 — 8 words — < 1%
Crossref

78 Harsh Kakadiya, Janavi Popat, Neeraj Kumar Singh, Lalit Tak, Mahshooq Abdul Majeed, Soumya Mudgal, Vasundhara Mahajan. "Chapter 27 Analysis and Prevention of Denial of Service Attacks in Smart Grid Using IoT", Springer Science and Business Media LLC, 2022 — 8 words — < 1%
Crossref

79 Madhumathi R, Arumuganathan T, Shruthi R. "Soil NPK and Moisture analysis using Wireless Sensor Networks", 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020 — 8 words — < 1%

Crossref

80	Sulove Bhattarai, Sudip Bhujel, Santosh Adhikari, Shanta Maharjan. "Design and Implementation of a Portable ECG Device", Journal of Innovations in Engineering Education, 2020 Crossref	8 words — < 1%
81	docs.rs-online.com Internet	8 words — < 1%
82	ebin.pub Internet	8 words — < 1%
83	erepo.uef.fi Internet	8 words — < 1%
84	ijetsr.com Internet	8 words — < 1%
85	jusst.org Internet	8 words — < 1%
86	pt.scribd.com Internet	8 words — < 1%
87	pure.coventry.ac.uk Internet	8 words — < 1%
88	spectrum.library.concordia.ca Internet	8 words — < 1%
89	uis.brage.unit.no Internet	8 words — < 1%
90	Arseniy Vasilenko, Svetlana Studennikova, Albina Agibalova. "Integration of a non-contact	7 words — < 1%

temperature sensor in the terminal using remote monitoring of readings", E3S Web of Conferences, 2021
Crossref

91 Dudi Darmawan, Doan Perdana, Abrar Ismardi, Indra Wahyudhin Fathona. "Investigating the electrical properties of soil as an indicator of the content of the NPK element in the soil", Measurement and Control, 2022
Crossref
6 words — < 1%

92 Nari Kim, Kyung-Ja Ha, No-Wook Park, Jaeil Cho, Sungwook Hong, Yang-Won Lee. "A Comparison Between Major Artificial Intelligence Models for Crop Yield Prediction: Case Study of the Midwestern United States, 2006–2015", ISPRS International Journal of Geo-Information, 2019
Crossref
6 words — < 1%

93 R. Silva. "Accessing remote special files in a distributed computing environment", [1993] Proceedings The 2nd International Symposium on High Performance Distributed Computing, 1993
Crossref
6 words — < 1%

EXCLUDE QUOTES	ON	EXCLUDE SOURCES	OFF
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	OFF

REFERENCES

[1] M. Naresh and P. Munaswamy, "Smart Agriculture System using IoT Technology," 2019. [Online]. Available: www.ijrte.org

[2] V. Sathya Narayanan *et al.*, "Soil and Weather Monitoring System with Crop Prediction for Farmers Using Iot and Machine Learning," 2021.

[3] T. Harinditha Ruchirawya *et al.*, "Crop Recommendation System," 2020. [Online]. Available: https://www.researchgate.net/publication/346627389

[4] D. Mani and R. Edinburgh, "Crop-Yield Prediction And Crop Recommendation System." [Online]. Available: https://ssrn.com/abstract=4111856

[5] N. Kim, K. J. Ha, N. W. Park, J. Cho, S. Hong, and Y. W. Lee, "A comparison between major artificial intelligence models for crop yield prediction: Case study of the midwestern United States, 2006–2015," *ISPRS International Journal of Geo-Information*, vol. 8, no. 5. MDPI AG, May 21, 2019. doi: 10.3390/ijgi8050240.

[6] S. Pudumalar, E. Ramanujam, R. H. Rajashree, C. Kavya, T. Kiruthika, and J. Nisha, "Crop Recommendation System for Precision Agriculture," *IEEE Eighth International Conference on Advanced Computing (ICoAC)*, 2016.

[7] M. Bouni, B. Hssina, K. Douzi, and S. Douzi, "Towards an Efficient Recommender Systems in Smart Agriculture: A deep reinforcement learning approach," *Procedia Comput Sci*, vol. 203, pp. 825–830, 2022, doi: 10.1016/j.procs.2022.07.124.

[8] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, "A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming," *IEEE Access*, vol. 7. Institute of Electrical and Electronics Engineers Inc., pp. 156237–156271, 2019. doi: 10.1109/ACCESS.2019.2949703.

[9] M. A. Matin and M. M. Islam, "Overview of Wireless Sensor Network," in *Wireless Sensor Networks - Technology and Protocols*, InTech, 2012. doi: 10.5772/49376.

[10] S. S. Prayogo, Y. Mukhlis, and B. K. Yakti, "The Use and Performance of MQTT and CoAP as Internet of Things Application Protocol using NodeMCU ESP8266," *Proceedings of 2019 4th International Conference on Informatics and Computing, ICIC 2019*, Oct. 2019, doi: 10.1109/ICIC47613.2019.8985850.

[11] Espressif Systems, "ESP8266EX Datasheet," 2015. [Online]. Available: http://bbs.espressif.com/

[12] Aosong Guangzhou Electronics Co. Ltd., *Temperature and Humidity Module DHT11 Product Manual*. Accessed: Feb. 01, 2023. [Online]. Available: https://components101.com/sites/default/files/component_datasheet/DHT11-Temperature-Sensor.pdf

[13] “Crop Recommendation Dataset | Kaggle.” https://www.kaggle.com/datasets/atharvaingle/crop-recommendation-dataset (accessed Feb. 01, 2023).

[14] E. Grossi and M. Buscema, “Introduction to artificial neural networks,” *European Journal of Gastroenterology and Hepatology*, vol. 19, no. 12. pp. 1046–1054, Dec. 2007. doi: 10.1097/MEG.0b013e3282f198a0.

[15] S. Sharma, S. Sharma, and A. Athaiya, “ACTIVATION FUNCTIONS IN NEURAL NETWORKS,” *International Journal of Engineering Applied Sciences and Technology*, vol. 4, pp. 310–316, 2020, Accessed: Feb. 01, 2023. [Online]. Available: http://www.ijeast.com

[16] S. Ruder, “An overview of gradient descent optimization algorithms,” Sep. 2016, [Online]. Available: http://arxiv.org/abs/1609.04747

[17] “Wireless Sensor Network (WSN) - GeeksforGeeks.” https://www.geeksforgeeks.org/wireless-sensor-network-wsn/ (accessed Feb. 01, 2023).