**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

Jnana Sangama, Belagavi-560018, Karnataka, India



**A Project report on**

**“IoT-Enhanced Crop Prediction”**

Submitted in partial fulfillment of the requirements for the award of the degree in

**BACHELOR OF ENGINEERING**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

*Submitted by*

**AYAN PAL(1VE19CS025)**

**ANKIT KUMAR (1VE19CS020)**

**Under The Guidance of**

**DR. PRATHIMA V R**

**CSE-AI, HOD**

**Department of Computer Science and Engineering**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SRI VENKATESHWARA COLLEGE OF ENGINEERING**

**Vidyanagar, Bengaluru-562157**

**2022-2023**

****

**SRI VENKATESHWARA COLLEGE OF ENGINEERING**

**Vidyanagar, Bengaluru-562157**

**Department of Computer Science and Engineering & Mechanical Engineering**

**CERTIFICATE**

This is to certify that the project report entitled **“IoT-Enhanced Crop Prediction”** carried out by **AYAN PAL(1VE19CS025), ANKIT KUMAR (1VE19CS020)** bonafide of the **VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI** during the year **2022-2023**. It is certified that all changes and suggestions for the internal assessment have been included in the report that has been submitted in the department library. The Project Report has been accepted because it meets the academic standards for the Degree's Project Report. Students at **SRI VENKATESHWARA COLLEGE OF ENGINEERING** working for a Bachelor of Engineering degree in Computer Science and Engineering.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Dr. Prathima V R Dr. Nageswara Guptha Dr. Hema M S**

Professor and Head Principal ,SVCE Professor and Head

Department of CSE-AI Department of CSE

**Name of the Examiners Signature with Date**

**1.**

**2.**

****

**SRI VENKATESHWARA COLLEGE OF ENGINEERING**

**Vidyanagar, Bengaluru-562157**

**Department of Computer Science and Engineering**

**DECLARATION**

I hereby declare that the projects report entitled **“IoT-Enhanced Crop Prediction”** Project / Professional Practice report submitted in the partial fulfillment of the requirements for the award of degree of **BACHELOR OF ENGINEERING** in **COMPUTER SCIENCE AND ENGINEERING** of the **VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI** is an authentic record of my own work carried out during 2022-2023.

The report embodied in this Project report has not been submitted to any other University or Institute for the award of any Degree or Diploma.

**Place: Bengaluru**

**Date: 09-03-2023 AYAN PAL (1VE19CS025)**

**ANKIT KUMAR (1VE19CS020)**

**ACKNOWLEDGEMENT**

A long-term endeavour can only be effective with the blessings of almighty god and the counsel of well-wishers. I'd want to take this opportunity to convey my profound gratitude and heartfelt thanks to numerous key individuals whose consistent support and encouragement have enabled me to focus my efforts on the completion of this Project Report.

I consider it as our privilege to express my gratitude and respect to all those who guided me in the completion of this Project Report. I would like to express my gratitude to **Sri Venkateshwara College of Engineering (SVCE**), for their kind cooperation and encouragement which help me in completion of this Project Report.

I sincerely thanks to my guide **Dr. Prathima V R, CSE-AI, SVCE,** for her valuable guidance.

I would like to express my gratitude to **Dr. Hema M S, Professor and Head, Department of Computer Science and Engineering, SVCE,** for providing unrelenting support and constant encouragement.

I would like to express my gratitude to **Dr. Nageswara Guptha, Principal, SVCE, Bengaluru** who has always been a great source of inspiration.

I would also like to mention my special thanks to all other **Teaching and Non-Teaching** staff of the Department of Computer Science and Engineering & Mechanical Engineering, SVCE who helped for the completion of the Project Report successfully.

I also thanks to my beloved parents for their continuous support in completion of this Project Report.

**(AYAN PAL)**

**(ANKIT KUMAR )**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **PARTICULARS** | **PAGE No.** |
| **1** | **COVER PAGE** | **1** |
| **2** | **COLLEGE CIRTIFICATE** | **2** |
| **3** | **DECLARATION** | **3** |
| **4** | **ACKNOWLEDGEMENT** | **4** |
| **5** | **TABLE OF CONTENTS** | **5** |
| **6** | **LIST OF FIGURES** | **7** |
| **CHAPTER 1 -INTRODUCTION**  **1.1- RELEVANCE OF THE PROJECT**  **1.2- PROBLEM STATEMENT**  **1.3- OBJECTIVE**  **1.4- METHODOLOGY** | | **09**  **09**  **10**  **10**  **15** |
| **CHAPTER 2 – LITERATURE SURVEY**  **2.1-MACHINE LEARNING APPROACH FOR FORECASTING**  **CROP BASED ON CLIMATE PARAMETERS**  **2.2-** **PREDICTING THE IMPACTED CROP USING MACHINE**  **LEARNING ALGORITHM**  **2.3- A RESEARCH ON CROP PREDICTION USINGMACHINE**  **LEARNING ALGORITHM**  **2.4- IOT BASED SMART AGRICULTURE**  **2.5- RESEARCH ON SMART AGRICULTURAL**  **PREDICTION USING IOT** | | **18**  **18**  **19**  **21**  **22**  **24** |
| **CHAPTER 3 – SYSTEM REQUIREMENTS SPECIFICATION**  **3.1- FUNCTIONAL REQUIREMENTS**  **3.2-NON-FUNCTIONAL REQUIREMENTS**  **3.3-HARDWARE REQUIREMENTS**  **3.4-SOFTWARE REQUIREMENTS** | | **26**  **27**  **29**  **29**  **38** |
| **CHAPTER 4 – SYSTEM ANALYSIS AND DESIGN**  **4.1- SYSTEM ARCHITECTURE**  **4.2- FLOWCHART**  **4.3- CIRCUIT DIAGRAM** | | **41**  **42**  **43**  **44** |
| **CHAPTER 6 – RESULTS AND DISCUSSION** | | **54** |
| **CHAPTER 8 – CONCLUSION AND FUTURE SCOPE** | | **57** |
| **CHAPTER 9 – REFERENCES** | | **58** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIG.No.** | **PARTICULARS** | **PAGE No.** |
| **1.4.1** | Types of Machine Learning | 12 |
| **1.4.2** | Supervised Learning | 12 |
| **1.4.3** | Unsupervised Learning | 13 |
| **1.4.4** | Reinforcement Learning | 13 |
| **1.4.5** | Steps in ML | 14 |
| **1.4.6** | Dataset | 15 |
| **2.4** | Smart Agriculture | 23 |
| **3.3.1** | ESP 32 | 30 |
| **3.3.2** | ESP Architecture | 30 |
| **3.3.3** | Soil Moisture Sensor | 33 |
| **3.3.4** | Agriculture Land | 33 |
| **3.3.5** | LCD Screen | 34 |
| **3.3.6** | DHT11 | 35 |
| **3.3.7** | Ultrasonic sensors | 36 |
| **3.3.8** | Waterproof digital temperature thermal probe sensor | 37 |
| **3.4.1** | Python | 38 |
| **3.4.2** | Arduino blink ide | 39 |
| **3.4.3** | Toolbar | 40 |
| **3.4.4** | Arduino Port Number | 41 |
| **4.1** | System Functionalities | 42 |
| **4.2** | Flowchart | 43 |
| **4.3** | Circuit diagram | 44 |
| **4.3.1** | Circuit Implementation | 44 |
| **5.1** | Tkinter application | 54 |
| **5.2** | Test case 1 | 55 |
| **5.3** | Test case 2 | 55 |
| **5.4** | Test case 3 | 56 |

**CHAPTER 1**

**INTRODUCTION**

India ranks second in the world in terms of agricultural output. In 2009, farming and related industries employed over 50% of all workers and contributed 16.6% of GDP. The financial contribution of agriculture to India's GDP is constantly declining. A range of factors influence plant crop production, including climatic, geographic, biological, political, and economic issues. It might be difficult for farmers to raise numerous crops, especially if they are unaware of market values. According to Wikipedia, the farmer suicide rate in India during 2004 and 2005 was between 1.4 and 1.8 per 100,000 persons. Suicides among farmers surged from 5650 in 2014 to nearly 8000 in 2015.

Using technology to improve agricultural awareness has become increasingly important in recent years. Food insecurity is caused by seasonal climate variations that impair essential resources such as soil, water, and air. In a situation where crop yield rates are consistently falling short of meeting demand, a smart system that can solve the issue of diminishing crop output is necessary. To address this issue, we propose a system that would allow farmers to select crops based on economic and environmental concerns in order to optimise production, so assisting them in meeting the nation's expanding demand for food supply. To ensure that farmers receive the most possible yield from their crops, the system will offer crop yield and crop selection depending on weather conditions that are optimal for the crop. The system analyses variables like rainfall, temperature, area (measured in hectares), season, etc. to predict crop yield. Additionally, the system aids in advising whether a specific time is the best one to apply fertilisers.

Crop type prediction under certain climate conditions is a key agricultural issue. Every farmer seeks to predict production and assess whether it will meet their objectives. Previously, yield prediction estimates were reliant on a farmer's prior knowledge of a certain crop. Weather, pests, and harvest operation planning all have a significant impact on agricultural yield. Accurate knowledge regarding the history of crop type is critical for making agricultural risk management decisions.

* 1. **RELEVANCE OF THE PROJECT**

India has recently experienced economic and societal instability as a result of an increase in small- and marginal-farmer suicide rates. Our goal is to provide these farmers and communities with the assistance and resources they require to address these concerns. They frequently find it difficult to believe in and acquire access to educational programmes and training that teach them how to better their financial situation and crop production. Because of the complexity of the issues involved, as well as the inherent reluctance to trust assistance from outside the community, any service or product intended to help must be thoroughly produced and assessed to assure positive outcomes and effective adoption.

There is currently no software that advises crops based on a range of criteria such as soil type and meteorological conditions such as temperature and rainfall. The modern systems are likewise hardware-based, which adds the cost and complexity of maintenance. Based on weather circumstances, the proposed system provides an application that reliably predicts the farmer's most profitable crop. The user's location is determined via GPS, and they are then questioned about the soil type in the area. The soil and meteorological databases are utilised to determine crop production in each site based on the user's location.

The outcome of system processing is sent to the user's application after it has been completed. Past crop performance is also used in order to generate credible crop yield results. The most productive crop is advised based on yield after taking into account a number of different scenarios and other filters based on user requirements. Our goal was to foresee a beneficial outcome using data so that farmers and aid workers could make well-informed planning decisions. Although there are numerous approaches to assist our target audience in living better lives, it was our responsibility to accomplish so.

**1.2 PROBLEM STATEMENT**

The goal of the problem statement is to forecast agricultural productivity by using machine learning techniques. The project's goal is to help users choose the best crop to grow in order to optimise yield and, as a result, profit.

The proposed system tries to improve on the faults of present systems and make predictions by studying structured data.The technique we advocate is to increase the variety of crops that can be cultivated throughout the season and to create a system that considers the factors that have the greatest impact on how well a crop develops.

**1.3 OBJECTIVES**

* The goal of this research is to forecast the types of crops that will grow under given climatic conditions and, as a result, recommend appropriate crops for that location. The following actions are involved.
* Gather information on temperature, rainfall, moisture, water level, crop yields, and soil types. Arrange the datasets that have been combined. Data cleaning improves data quality and, as a result, overall productivity by eliminating erroneous, incomplete, and irrelevant data.
* Use exploratory data analysis to examine the entire dataset and discover the key features. It is used to identify trends, identify outliers, and provide graphical representations of various features.
* To assess the crop's health in terms of location, time, and potential consequences.
* To create and develop a crop forecast technique that assesses soil quality in real time.

**1.4 METHODOLOGY**

The system estimates the crop using machine learning, and Python is utilised as the programming language because it is well-known for use in machine learning research. Machine learning uses historical data and expertise to learn from past experiences and produce a trained model. The model then predicts the outcome. A better dataset collection will increase the classifier's accuracy. Machine learning approaches such as regression and classification have been shown to outperform other statistical models.

Crop production is entirely dependent on geographic factors such as soil chemistry, rainfall, terrain, soil type, temperature, and so on. These elements have a substantial impact on crop yield. These factors have a significant impact on crop yield growth. Furthermore, market conditions influence which crop(s) should be planted for the best return on investment. To forecast the yield, we must analyse each aspect separately.We developed a system in the agriculture industry that uses machine learning techniques to estimate crop production based on characteristics such as rainfall, temperature, acreage, season, and so on.

**1.4.1 MACHINE LEARNING**

Machine learning is without a doubt one of the most significant and effective technologies in use today. Machine learning is a data-to-knowledge conversion tool. a half-century ago Data has become increasingly abundant in recent years. Because this massive amount of data is meaningless, we evaluate it to uncover its underlying patterns. Machine learning algorithms reveal crucial underlying patterns in complex data that we might otherwise miss.

One can use hidden patterns and topic expertise to make difficult decisions and anticipate the future. The process through which computers learn the rules controlling a phenomenon is known as machine learning. They must put several laws to the test and observe how well they work.

**1.4.2 BASIC TERMINOLOGY**

* Dataset: A dataset is a collection of data examples that provide characteristics necessary for resolving the issue.
* Features: Vital facts that advance our comprehension of a topic. To aid in learning, these are given into a machine learning system.
* Model: A machine learning algorithm's internal representation of a phenomenon that it has learned. It discovers this from the training data that is presented to it. The product of training an algorithm is the model. A decision tree algorithm, for instance, might be trained to create a decision tree model.

**1.4.3 TYPES OF MACHINE LEARNING**

Machine Learning can be supervised, unsupervised, semisupervised, or reinforcement learning. Each type of Machine Learning has a unique method, but they all adhere to the same fundamental technique and theory.

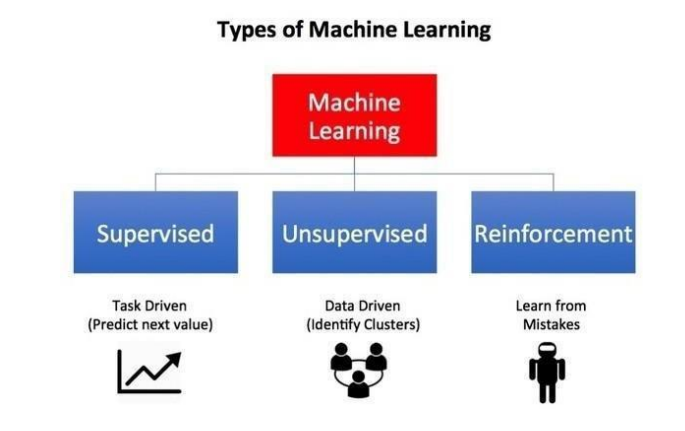
****

Figure 1.4.1 Types of Machine Learning

**Supervised Learning:** The most commonly used machine learning paradigm is supervised learning. We can provide data in the form of example-label pairs to learning algorithms so that they can predict the label for each example and receive feedback on whether or not they did so correctly. The algorithm will eventually develop a rough approximation of the precise nature of the relationship between instances and labels.

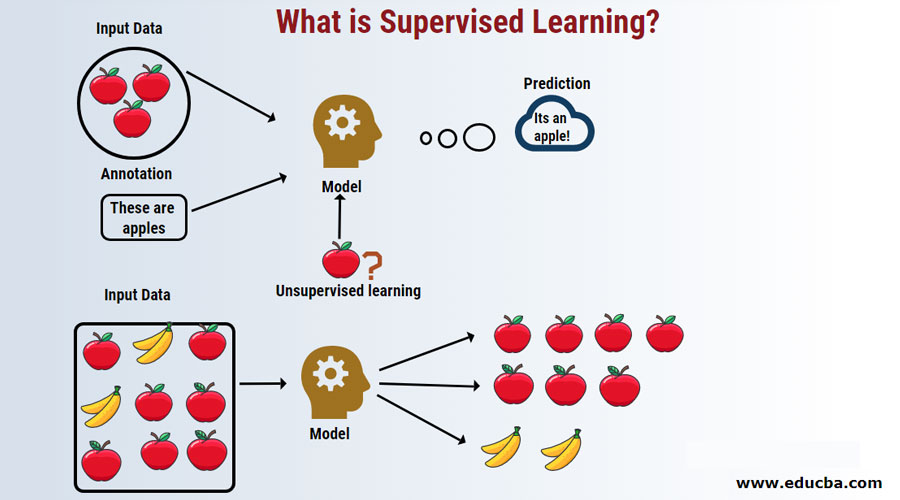


Figure 1.4.2 Supervised Learning

**Unsupervised Learning:** The polar opposite of supervised learning is unsupervised learning. There are no labels on it. Instead, a large amount of data would be provided, together with the tools required for the algorithm to interpret the data's properties. Then, by learning to group, cluster, and organise the data, it will be able to figure out how to organise the data so that a human can enter and interpret the newly organised data. Unsupervised learning is also known as data-driven learning because it is based on data and its features. The data and how it is organised determine the consequences of an unsupervised learning activity.

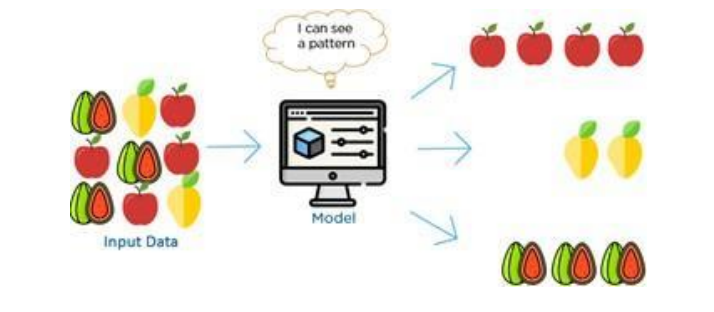
****

Figure 1.4.3 Unsupervised Learning

**Reinforcement Learning:** When compared to reinforcement learning, there are some distinctions between supervised and unsupervised learning. Behaviour is the foundation of reinforcement learning. It is influenced by the realms of psychology and neuroscience. We need an agent, an environment, a mechanism to connect the environment and the agent via a feedback loop, and all of this for each reinforcement learning assignment. To connect the agent to the environment, we give it with a set of environmental-impacting actions. To keep the agent connected to the world, we require it to receive two signals on a continuous basis: an updated state and a reward (our reinforcement signal for behaviour).

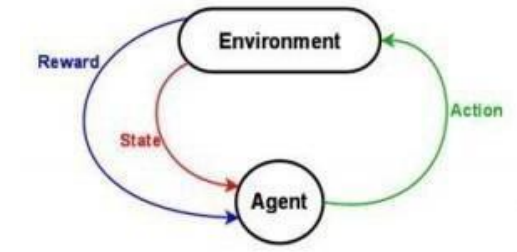
****

Figure 1.4.4 Reinforcement Learning

**1.4.4 BASIC PROCESS**

i. Data Collection: Collect the data from which the algorithm will learn.

ii. Data Preparation: Format and design the data into the best possible format, extracting significant features and reducing dimensionality.

iii.Training: This is the stage where the Machine Learning algorithm really learns by displaying it the data that has been collected and prepared. It is also known as the fitting stage.

iv.Evaluation: Examine the model to see how well it works.

v. Tuning: To improve the model's performance, fine-tune it.

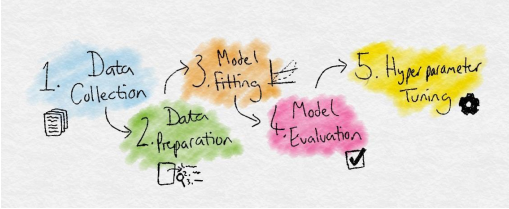
****

Fig 1.4.5 Steps in ML

**1.4.5 DATASET**

Machine learning is strongly reliant on data. It is the component that best facilitates algorithm training. It creates experience by utilising historical data and information. When the dataset is collected more precisely, the accuracy will improve.

The first step is to collect information. The investigation necessitates the use of two datasets. One simulates the yield prediction method, while the other forecasts the weather.To put it another way, the average temperature and rainfall. These two variables are expected to be used as inputs for crop yield forecasting. The meteorological data and crop yield information in datasets are sourced from the websites and https://www.kaggle.com/srinivas1/agricuture-crops-production-in-india, respectively.Given that these are the most important variables influencing crops, the yield prediction module dataset should include the following columns: State, District, Crop, Season, Average Temperature, Average Rainfall, Soil Type, Area, and Production. The class variable, also known as the dependent variable, is "Production." There are eight independent variables and one dependent variable. The datasets were combined to achieve this. When the two datasets were joined, the common attribute was location. Because of their high rates of farmer suicide, Maharashtra and Karnataka are the only two states we are considering.

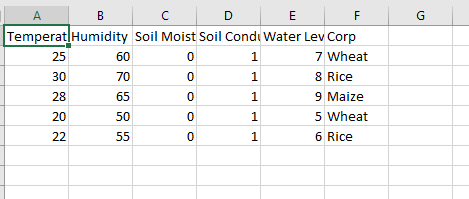


Fig 1.4.6 Dataset

**1.4.6 ALGORITHM USED**

There is a wide range of machine learning algorithms accessible. The most common divisions are classification, regression, grouping, and association. While clustering and association are unsupervised learning techniques, classification and regression are not.

**DecisionTreeClassifier:** To implement decision trees, we need to collect the data in a proper format to build a decision tree. We will be using the iris dataset from the sklearn datasets databases, which is relatively straightforward and demonstrates how to construct a decision tree classifier.

### The Benefits of Scikit-Decision The goal variable in Learn's Tree Classifier can be either numerical or categorised. For the sake of clarity, we shall keep the flowers' categorical nature in the iris dataset.

### Importing the Dataset:

import pandas as pd

import numpy as np

from sklearn.datasets import load\_iris

data = load\_iris()

#convert to a dataframe

df = pd.DataFrame(data.data, columns = data.feature\_names)

#create the species column

df['Species'] = data.target

#replace this with the actual names

target = np.unique(data.target)

target\_names = np.unique(data.target\_names)

targets = dict(zip(target, target\_names))

df['Species'] = df['Species'].replace(targets)

The following step will be used to extract our testing and training datasets. The goal is to guarantee that the model is not trained on all of the given data, enabling us to observe how it performs on data that hasn't been seen before. If we use all of the data as training data, we risk overfitting the model, meaning it will perform poorly on unknown data.

**CHAPTER 2**

**LITERATURE REVIEW**

A literature survey is a systematic and thorough examination of all sorts of published literature as well as additional sources, such as dissertations, to discover as many articles as possible that are relevant to a specific topic. Prediction of agricultural products is critical in agriculture. It aids in improving net produce, improved planning, and profit maximisation. We reviewed a few research papers connected to our project issue in order to attain better results.

**2.1 MACHINE LEARNING APPROACH FOR FORECASTING CROP TYPE BASED ON CLIMATIC PARAMETERS**

**Authors:** S.Veenadhari, Dr. Bharat Misra & Dr. CD Singh

**Publication:** International Conference on Computer Communication and Informatics (ICCCI - 2014), Jan, 2014

The main goal of this research was to develop a website that will allow users to investigate the impact of meteorological conditions on crop productivity in specific Madhya Pradesh districts. The districts were chosen based on the amount of land cultivated with that particular crop. Based on this criterion, the first five districts with the highest selected crop area were picked. The crops were chosen based on the most common crops in the district chosen for the study. Wheat, maize, paddy, and soybeans were chosen as crops. The yields of these crops were totaled over a 20-year period using secondary data. Rainfall, maximum and minimum temperatures, possible evapotranspiration, cloud cover, and sunlight hours are all likely climatic elements for the corresponding years. Secondary sources were used to obtain the frequency of wet days. The methods used for analysis included considering numbers above the threshold as one child and the remainder as second child. It also handles missing attribute values. In pseudo code, the general algorithm for building decision trees is:

• For each attribute a : find out about the normalized information gain from splitting on a

• Let a\_best be the attribute value with the highest normalized information gain.

• Create a decision node that splits on a\_best .

• Recurse on the sublists obtained by splitting on a\_best, and add those nodes as children

of node.

In this relevance analysis method, they estimated the information resource gain for each attribute characterising the samples in S. The most discriminating attribute in the supplied collection was determined to have the most information gain. They calculated the information acquired for each attribute to generate a ranking of the attributes. This score can be used for relevance analysis when deciding which traits to employ in idea description. On the system utilised to construct a web-based application. The database is SQL Server 2008.

Forecasts were accurate in 18 of the 20 years of data and incorrect in two, placing the prediction accuracy of the built model at 90% in the instance of soybean in the Dewas district. The forecast accuracy of the generated model ranged from 76 to 90% for the selected crops and areas. Based on these observations, the overall prediction accuracy of the created model is 82.00%. This study focuses on relevance method analysis to ensure trustworthy prediction. This is performed by computing and comparing the information gain of each attribute. It does, however, preclude the investigation of other supervised machine learning techniques methods such as linear regression and random forest.

**2.2** **PREDICTING THE IMPACTED CROP USING MACHINE LEARNING ALGORITHM**

**Author:** P.Priya, U.Muthaiah & M.Balamurugan

**Publication:** International Journal of Engineering Sciences & Research Technology (IJESRT), April, 2018

In this research Machine learning approaches are integrated with python programming. Python is the industry standard for statistics, data analysis, and machine learning. Because it is a programming language rather than a statistical package, you can create your own unique objects, methods, and packages. It is free and platform agnostic, thus it can be used with any operating system. R scripts thoroughly define the stages of our study and make it simple to reproduce and/or update analysis, allowing us to quickly test out new ideas and/or fix difficulties. All of the research's datasets came from the Indian government's publicly available sources. This data was gathered from 1997 to 2013 for various rice crop seasons such as Kharif and Rabi. Only a small number of critical parameters with the greatest impact on agricultural productivity were chosen from the large initial dataset for the current study. The following parameters are included in the dataset: rainfall, season, temperature, and crop productivity. In addition, this study analyses two machine learning algorithms: decision trees and Random Forest.

* Decision Tree: Decision tree classifiers use a greedy approach, an attribute chosen in the first stage cannot be used again, but it may be used in subsequent rounds to enhance classification. Furthermore, it overfits the training data, which might result in poor outcomes for unseen data. To overcome this limitation, ensemble modelling is used. Ensemble models combine the results of multiple models. The output of an ensemble model often outperforms the output of any single individual model.
* Random Forest: To forecast the outcome, this ensemble classifier employs many decision tree models. Each tree is trained using a different subset of the training data. A forest is a collection of trees, and because the trees are trained on randomly selected subsets, random forests are collections of trees. This can be used to tackle classification and regression problems. The class is determined by the total number of votes cast across all trees, and the average of the results is used for regression. The procedure they used in this paper is as follows.
* Divide the loaded data sets into two groups, such as training data and test data, with a 67%/33% split ratio.
* Before summing the data sets, compute the Mean and Standard Deviation for the needed tuples. Compare the summarised data list to the original data sets to compute the likelihood.
* The highest probability produced is used for prediction based on the results. By comparing the resultant class value to the test data set, the accuracy can be estimated. The precision can range between 0% and 100%.

**2.3 A SURVEY ON CROP PREDICTION USING MACHINE LEARNING ALGORITHM**

Authors: Sriram Rakshith.K, Dr. Deepak.G, Rajesh M, Sudharshan K S, Vasanth S & Harish Kumar

Publications: International Journal for Research in Applied Science & Engineering Technology

Machine Learning-Based Crop Yield Prediction (IJRASET Volume 7, Issue IV, April 2019) This study focuses on the tactics and measures utilised to improve farming in order to make the agricultural industry more trustworthy and simple for farmers. Among these strategies and procedures are instilling technical knowledge and developments, as well as applying machine learning to anticipate the suitable crop by sensing parameters such as soil, weather, and market trends. The pH of the soil, the nitrogen-phosphate-potassium concentration, the temperature, precipitation, and humidity are all considered. They consider data mining approaches such as artificial neural networks and fuzzy information networks.

1. Artificial Neural Network: Neural networks are made up of input, output, and hidden layers, with neurons providing input to the ANN, which is then executed by specified units in the hidden levels and utilised by the output layers to produce output. As the amount of data increases, neural networks become more precise. ID3 and other optimisation algorithms are used in artificial intelligence and machine learning for tomato crop detection. To develop a tomato crop-specific expert system. Machine learning techniques and methods are used in maize production. Along with rice genotypes that are adaptive and well-suited to drought conditions, maize is another prevalent crop and a major source of cereals that must be adopted.

2. Fuzzy Information Network: Neural networks are used to forecast and analyse crop yields. The neuro-fuzzy inference system uses the repository organ, ground biomass, and soil moisture content as inputs. Another difficulty is that yield forecasting is falling behind schedule when using remote sensing. The algorithm is designed in a such way that it ignores the first year and only uses the data after that. The deviation is calculated by comparing the yield to the one that was left out. Because the impacts of climate change is thought to play a big role in the sector of agriculture, the study that was done to integrate the element of incomplete information used a web-based decision support system with an agricultural fuzzy logic advancement.

The detailed information about soil rainfall arrangement will be taken into consideration to bring about accomplished results.

3. Data Mining Techniques: Data mining is used to examine acquired data in order to provide insights for decision making.We can lessen the risk connected with agriculture by forecasting the information or results obtained through data mining tools.

**2.4 IOT BASED SMART AGRICULTURE**

Authors: Nikesh Gondchawar1 , Prof. Dr. R. S. Kawitkar2

Publications: International Journal of Advance Research in Computer and Communication Engineering Vol. 5, Issue 6, June 2019

The contemporary scenario of declining water tables, shrinking rivers and storage tanks, and unpredictable weather highlights the critical necessity for proper water management. To deal with this, temperature and moisture sensors are strategically placed to monitor crops.

[1] A microcontroller-based gateway can be designed with an algorithm that uses threshold values for soil moisture and temperature to control water flow. Photovoltaic panels may power the system, and it may also include a duplex communication link based on a cellular Internet interface that allows data inspection and irrigation scheduling to be adjusted via a web page.

[2] The advancement of technology in Wireless Sensor Networks has enabled them to be used in the monitoring and control of greenhouse parameters in precision agriculture.

[3] After doing research in this subject, scientists discovered that agricultural output is constantly declining. However, agricultural technology plays a vital role in both raising output and minimising the demand for more work. Some research activities are intended to assist farmers by giving them with access to agricultural productivity-boosting technology. Y. Kim created a remote sensing and control irrigation system based on a distributed wireless sensor network with the goal of enhancing productivity while using as little water as feasible. The system included information on variable rate irrigation design and instrumentation, wireless sensor networks, and real-time field sensing and control utilising the necessary software. Five in-field sensor stations were used to build the overall system. These sensor stations collect data and communicate it via GPS to the base station, where irrigation is regulated using the system's database. The method allows for precision watering as well as a potential low-cost wireless solution.

[4] In studies employing wireless sensor networks, researchers evaluated soil-related factors such as temperature and humidity. Subsurface sensors were added to communicate with relay nodes through an efficient protocol with a very low duty cycle, extending the system's longevity. A microprocessor, sensors, and a universal asynchronous receiver transmitter (UART) interface were used to build the system, and data transfer was accomplished through hourly sampling, buffering, transmission, and status message checking. The system's cost and the sensor's placement beneath the earth, which weakens radio frequency (RF) transmissions, were drawbacks.

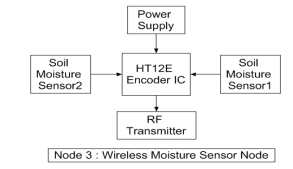


Figure 2.4 Smart Agriculture

The Raspberry Pi has successfully interfaced with all three Nodes' sensors and microcontrollers, allowing wireless communication between them. According to all observations and practical tests, the project provides a comprehensive solution to field activities, irrigation challenges, and storage issues through the employment of remote-controlled robots, smart irrigation systems, and smart warehouse management systems, in that order. The use of such a device in the field can undoubtedly help to increase crop productivity and total production.

**2.5 RESEARCH ON SMART AGRICULTURAL PREDICTION USING IOT**

Authors: Shweta A M, Dr V. Nagaveni

Publications: Journal of Computer Programming and Multimedia Volume 4 Issue 1,2021

Temperature, humidity, and light are the most critical elements for the quality and productivity of plant development. Continuous monitoring of various environmental variables provides significant information to the grower, allowing him or her to better understand how each element impacts growth and how to maximise crop productivity –

[1]. In particular during the winter in northern nations, the ideal greenhouse micro climate adjustment might help us increase production and produce impressive energy savings

[2]. WSNs outperform wired networks in terms of precision, fault tolerance, flexibility, cost, autonomy, and robustness. They are made up of hundreds of nodes that can sense, act, and communicate. Furthermore, the emergence of IoT and M2M communications positions it to play a significant enabling role in a variety of industries, including the military, environment, health, home, and others.

[3]. IoT is a broad term that refers to a variety of technologies that allow devices to communicate with one another, either with or without human intervention. Furthermore, the suggested setup can be combined with various internet and messaging services (e.g., Web, WAP, SMS) to provide farmers with communication. Wireless sensor networks (WSNs) are one of the most important technologies of the twenty-first century, and they are ideal for dispersed data collection and monitoring in harsh environments such as greenhouses. The Internet of Things (IoT) is another of the most important technologies of the twenty-first century, with hundreds of applications in civil, health, military, and agricultural fields. Multiple measurement stations are required to track down the local climate parameters in various sections of the greenhouse in order to ensure that the automation system for large-scale greenhouses runs effectively. The expense, susceptibility, and complexity of transporting the measurement equipment would be increased by cabling. A WSN prototype is used in this publication to measure greenhouse temperature, light, pressure, and humidity. Measurement data has been provided via the Internet of Things. Farmers can operate their greenhouses using this method from their internet-connected smartphones or computers.

There are numerous examples of flexible IoT application studies in the literature. While considering the system, an additional information management sub-system is established. The information acquired is provided in a format that agricultural research facilities can use. Liu Dan et al. present the design and implementation of an agriculture greenhouse environment monitoring system based on ZigBee connectivity, using an esp32 chip as the primary component, in this study. Furthermore, esp acts as the foundation for the wireless sensor and control nodes that control the environment data. At the front end, this system encompasses data processing, data transmission, and data receipt.The temperature sensor on the terminal node processes the ambient temperature in real time and delivers the results to the intermediate node over a wireless ZigBee network. The intermediate node combines all of the data before sending it to the PC through a serial connection. Data can be seen and analysed by staff members at the same time, and data storage on a PC is also possible. By managing the functioning of fans and other temperature-controlling apparatus, real-time data enables for autonomous temperature management in the greenhouse.

**CHAPTER 3**

**SYSTEM REQUIREMENT SPECIFICATIONS**

A software requirements specification (SRS) is the detailed description of a future software system. It provides a set of use cases that demonstrate the user interactions that the product must support, as well as the functional and nonfunctional requirements that must be met. It is critical to list the requirements and describe how to meet them in an SRS. The team's capacity to comprehend how they would approach the project allows them to save time. This allows the team to identify potential hazards and constraints early on.

An SRS can also be seen of as a full sketch of a software systems that needs to be constructed, with both functional and non-functional requirements included. The use cases for the user's interactions with the software system can be included. The software requirement specification document includes all important project requirements. We must understand the software system well in order to create it. To do this, continuous client connection is essential to collect all requirements.

A good SRS specify how the software system will interact with all internal pieces, hardware, other programmes, and human users in a variety of real-world scenarios. It is critical that testers understand every detail described in this document in order to avoid errors in test cases and expected outcomes.

**Qualities of SRS**

• Correct

• Unambiguous

• Complete

• Consistent

• Ranked for importance and/or stability

**3.1 FUNCTIONAL REQUIREMENTS**

A functional requirement describes the service that the software must provide. It gives data about software or software systems. A function is simply the inputs, behaviours, and outputs of a software system. A computation, data manipulation, business process, user interaction, or any other specific attribute can be used to determine a system's likely function. A functional requirement in software engineering and systems engineering can range from a high-level abstract description of the sender's desire to complicated mathematical functional requirement specifications.

With the use of functional software requirements, you may capture the system's anticipated behaviour.Benefits of functional requirements:

* Assists you in determining whether the application offers all of the functionalities listed in the functional requirements for that application.
* The functionality of a system or one of its subsystems can be defined with the use of a functional requirement document.
* Functional requirements and requirement analysis both help to find the requirements that are still missing. They assist in precisely defining the desired system behaviour and functionality.
* The least expensive fixes are for errors discovered during the functional requirement gathering stage.
* Support user objectives, jobs, or actions

**3.1.1 Basic Requirements**

**1.Data Collection:** This project's dataset is made up of information acquired from credible sources and combined to provide the required data collection. Crop and meteorological data sources for our datasets can be obtained at https://www.kaggle.com/srinivas1/agricuture-crops-production-in-india. The names of the crops, production, area, average temperature, average precipitation (mm), season, year, and the names of the states and districts are all included. The dependent variable, also known as the class variable, is "Production." There is one dependent variable and eight independent variables.

**2.Data Preprocessing:** Preprocessing is the process of converting raw data into a machine learning-friendly format. A data scientist can use an applied machine learning model to obtain more accurate results by using structured and clean data. As part of the technique, data must be cleansed, cleaned up, and sampled. In this case, data pre-processing focuses on detecting attributes with null or inaccurate values, as well as links between other attributes. Data Another advantage of pre-processing is that it allows you to learn how each parameter impacts the target parameter. Our datasets were preprocessed using the EDA method. Depending on the relevance of the field, all incorrect and null values were handled by either deleting the record or setting the default value for that specific attribute.

**3.Data Splitting:** When using a dataset for machine learning, the training and test sets should be separated into two groups. With an 80% split ratio, or 100 records, we divided the dataset in two. The training set contained 80 recordings, while the final set contained 20 records.

**4.Model Training:** A data scientist can begin creating a model after preprocessing the acquired data and separating it into train and test sets. This procedure involves "feeding" the algorithm with training data. An algorithm will process the data, generating a model capable of locating a target value (attribute) in new data and providing the response required for a predictive analysis. The goal of model training is to create a model. We used the random forest method to create our model. After training, the model can forecast the yield based on the other dataset properties.

**5.Model evaluation and testing:** This step's purpose is to create the simplest basic model capable of quickly and exactly creating a target value. A data scientist can accomplish this by fine-tuning models. Model parameters are tuned in this way to get the best performance out of an algorithm.

**3.2 NON-FUNCTIONAL REQUIREMENTS**

Non-Functional Requirement (NFR) describes the quality of a software system. They evaluate the software system based on non-functional criteria such as responsiveness, usability, security, portability, and other critical success factors. Non-functional requirements may prevent systems from meeting user needs. You can impose restrictions or limitations on the system design across numerous agile backlogs by using nonfunctional requirements. A website with over 10,000 concurrent users, for example, should load in 3 seconds. Rather of identifying specific actions, they define standards that can be used to assess how well a system is doing. They may be linked to emergent system characteristics such as dependability, response speed, and store occupancy. External factors such as user needs, financial constraints, organisational policies, the need for interoperability with other software and hardware systems, and others, as well as internal factors such as product requirements, organisational requirements, user requirements, basic operational requirements, and so on, can all result in non-functional requirements.

**Benefits of Non Functional Requirements:**

* The nonfunctional requirements ensure the software system follows legal and compliance rules.
* They ensure the reliability, availability, and performance of the software system.
* They ensure good user experience and ease of operating the software.
* They help in formulating security policy of the software system.

**3.3 HARDWARE REQUIREMENTS**

The hardware requirements specify the physical computer resources required for a system to function properly. Because they may form the basis of a contract for the system's implementation, the hardware specifications should be a thorough and consistent description of the entire system.

The following is a list of the hardware requirements:

**3.3.2 Soil Moisture Sensor:**

The Soil Moisture Sensor is used to determine the moisture content of the soil. It determines the quantity of moisture or water content in the soil. Coefficients are used by the soil moisture sensor to make computations. It calculates the amount of water in the soil. It detects the amount of water in the soil and receives and transmits analogue signals for digital display.

It sends signals to Arduino for extra processing and display of soil condition information, data, or values.

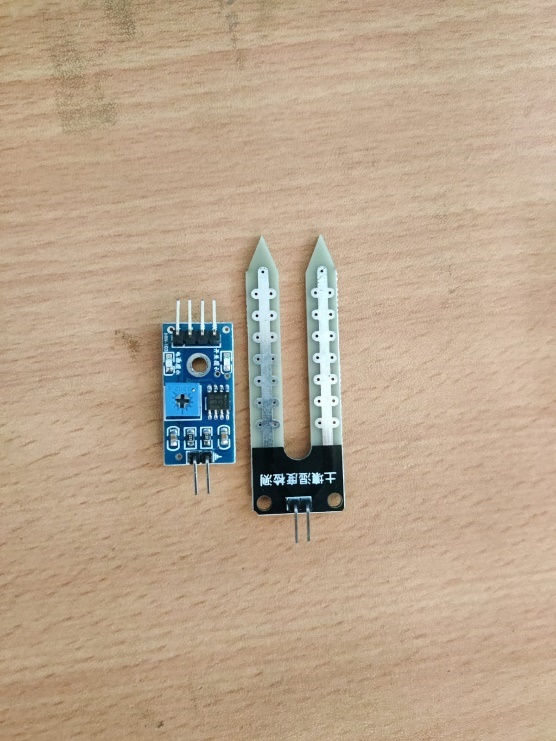
 

Figure 3.3.3 Soil Moisture Sensor Figure 3.3.4 Agriculture Land

**3.3.3 LCD Data Display Interface:**

This LCD device is most commonly used in Arduino, however it may be used with any 3.3V controller. It is a low-cost monochrome LCD module with 84 × 48 pixels. It can be used to display both visuals and text. The PCD8544 driver is used to power this display.

This device's pin arrangement is similar to that of a 16x2 LCD module, except that instead of 8 data pins, there is one serial data in (Din) pin and one clock pin.

The list of the pins and their description are listed below.

RST: Active low pin, therefore 0V Resets the LCD , Before transmitting anything to the device CE: Cheap Enable is used to enable it.

DC: Data/Command is used to choose between the Rata Register and the Command Register.

DIN: Data In is used to transfer serial data to the display. It could be either Data or Command.

CLK: The clock is used to sync the display with the controller.

VCC: The pin 5V or 3.3V is used to power the device.

BL: This pin is used to power the display's backlight.

GND: This is used to connect the gadget to the ground.



Figure 3.3.5 LCD Screen

Things we need

* Resistors
  1. 1k Ohms x 5 Nos.
  2. 330 Ohms / Potentiometer 1k
* Jumper Wires
* Bread Board

Connection Diagram:

pin 7 - Clock (CLK)

pin 6 - Data In (DIN)

pin 5 - Data/Command select (D/C)

pin 4 - Chip Enable/select (CE/CS)

pin 3 - Reset (RST)

**3.3.4 DHT11**

This project is about the interfacing of a DHT11 module with ESP 32. The data for temperature and humidity are displayed on the Serial Terminal.

Components Required:

1. ESP 32
2. DHT11 Module
3. Data Cable
4. Jumper wires

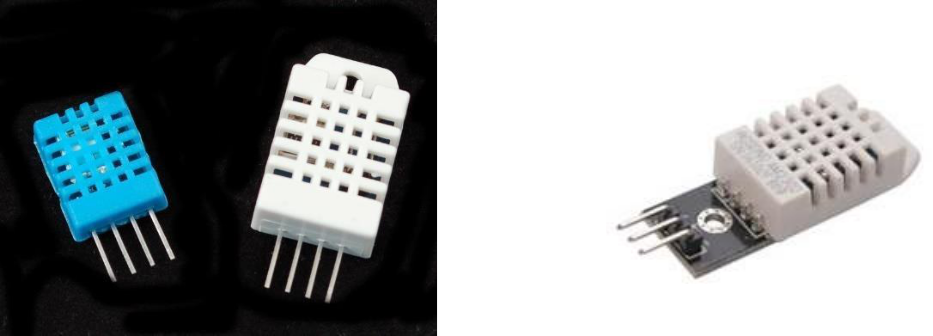
DHT11 sensors are simple and slow, but they are ideal for enthusiasts who want to do rudimentary data logging. DHT sensors are composed of two components: a capacitive humidity sensor and a thermistor. There's also a simple chip inside that converts analogue to digital and outputs a digital signal with the temperature and humidity. Any microcontroller can easily read the digital signal.****

Figure 3.3.6 DHT11

There are two versions of the DHT sensor, they look a bit similar and have the same pinout, but have different characteristics. Here are the specs:

DHT11:

* Ultra-low-cost
* Power and I/O from 3 to 5 volts
* Maximum current consumption during conversion (when requesting data) is 2.5mA.
* Accurate humidity readings of 20-80% are possible with this device.
* Accurate temperature measurements from 0 to 50°C
* A sampling rate of no more than 1 Hz (once per second)
* Body dimensions: 15.5mm x 12mm x 5.5mm
* 4 pins separated by 0.1"

Pinout: The module is powered by VCC. You may connect it directly to the Arduino's 5V pin. The temperature and humidity measurements are transmitted digitally via the data pin. GND is the Ground Pin and must be linked to the Arduino's GND pin.

**3.3.5 ULTRASONIC SENSORS**

An ultrasonic sensor is a piece of equipment that uses ultrasonic sound waves to detect the distance to a target item and converts the sound that is reflected back into an electrical signal. Ultrasonic waves move faster than audible sound, which is the type of sound that people can hear. The transmitter, which makes the sound using piezoelectric crystals, and the receiver, which detects the sound after it has travelled to and from the target, are the two primary components of an ultrasonic sensor.



Figure 3.3.7 Ultrasonic sensors

Ultrasonic sensors are frequently used in conjunction with proximity sensors. They are used in automobile technology for self-parking and anti-collision protection. Ultrasonic sensors are utilised in a variety of applications, including industrial and robot obstacle detection systems. Ultrasonic sensors are less prone than infrared (IR) sensors to be interfered with by smoke, gas, and other airborne particles in proximity sensing applications (but the physical components are still affected by factors such as heat). Ultrasonic sensors are also employed as level sensors in closed containers (such as vats in the chemical industry) to detect, monitor, and control liquid levels. Most crucially, ultrasound technology has allowed doctors to capture photos of interior organs, detect cancer, and safeguard the safety of unborn babies.

**3.3.6 WATERPROOF DIGITAL TEMPERATURE THERMAL PROBE SENSOR:**

This is a 1 metre long, pre-wired, waterproof digital temperature sensor probe based on the DS18B20 sensor. It comes in handy when you need to measure anything far away or in damp conditions. Because they are digital, there is no signal degradation even over long distances.

These 1-wire digital temperature sensors' integrated digital-to-analog converter may provide up to 12 bits of precision and is reasonably accurate (0.5°C across much of the range). Each one has a unique 64-bit ID that is burned in at the factory to distinguish it from the others, and they all operate great with any microcontroller through a single digital pin. You can even connect several to the same pin. 3.0-5.0V systems are appropriate.

****

Figure 3.3.8 Waterproof digital temperature thermal probe sensor

**3.4 SOFTWARE REQUIREMENT**

The software requirements are a description of the target system's features and functionalities. Users' expectations of the software product are communicated through requirements. From the client's perspective, the requirements can be evident or buried, known or unknown, expected or unexpected.

1. **Python:** It is a high-level, object-oriented programming language with integrated dynamic semantics that is mostly used for developing websites and mobile applications. It is highly appealing in the field of rapid application development since it provides options for dynamic binding and typing. Python has a distinct syntax that prioritises readability and is hence straightforward, making it easy to learn. Python code is significantly easier to read and translate for developers than code written in other languages. Teams can collaborate without significant language and experience barriers, lowering the cost of programme development and maintenance. Python additionally supports modules and packages, allowing for programme modularization and code reuse across projects.

****

Figure 3.4.1 Python

1. **Tkinter:** Tkinter is Python's default graphical user interface library. The combination of Python and Tkinter allows for the rapid development of GUI applications. Tkinter provides an excellent object-oriented interface for the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

• Add the Tkinter module.

• Make the primary window of the GUI application.

• Include one or more of the widgets listed above in the GUI application.

• Enter the main event loop to respond to each user-triggered event.

1. **Arduino IDE:** The Arduino IDE is open-source software that is used to create and upload code to Arduino boards. The IDE programme is suitable with a variety of operating systems, including Windows, Mac OS X, and Linux. C and C++ programming languages are supported. This sentence refers to the Integrated Development Environment.

Sketching is a term used to describe writing a programme or piece of code in the Arduino IDE. We must connect the Genuino and Arduino board to the IDE in order to upload the sketch prepared in the Arduino IDE software. The sketch is saved with the ".ino" file extension.

The Arduino IDE will appear as:

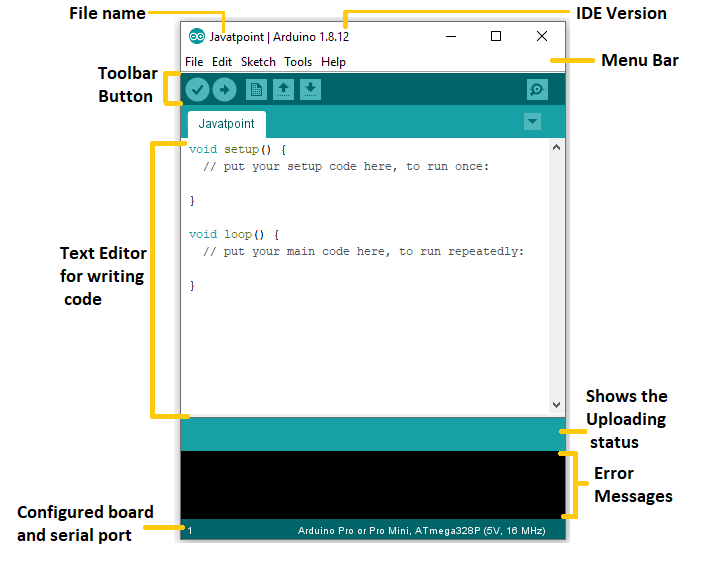


Figure 3.4.2 Arduino blink ide

**Toolbar Button:** The icons displayed on the toolbar are **New, Open, Save, Upload,** and **Verify**.

It is shown below:

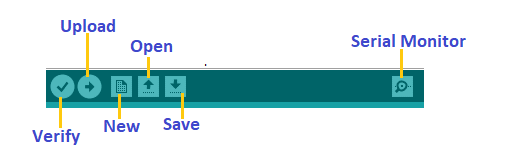


Figure 3.4.3 Toolbar

### Upload

The Upload button compiles and runs our on-screen code. It then transfers the code to the linked board. Before we upload the schematic, we must ensure that the correct board and ports are chosen.

A USB connection is also required to connect the board and the PC. Once you've completed all of the preceding steps, click the Upload button in the toolbar.

The most recent Arduino boards can be reset automatically before starting Upload. In the case of older boards, we must push the Reset button. When the uploading is completed successfully, the Tx and Rx LEDs will begin to blink.

### If the upload fails, the notice will be displayed in the error window.

### We don't need any further hardware to upload our programme with the Arduino Bootloader. A bootloader is a tiny programme that is loaded into the microcontroller on the board. On PIN 13, the LED will blink.

### Open

### The Open button is used to open a previously saved file. The currently open window will be opened with the selected file.

### Save

### The save button saves the current sketch or code.

### New

### It starts a new window or creates a new drawing.

### Verify

### The Verify button is used to inspect the drawing or written code for compilation errors.Serial Monitor

The serial monitor button is present on the right corner of the toolbar. It opens the serial monitor.

It is shown below:

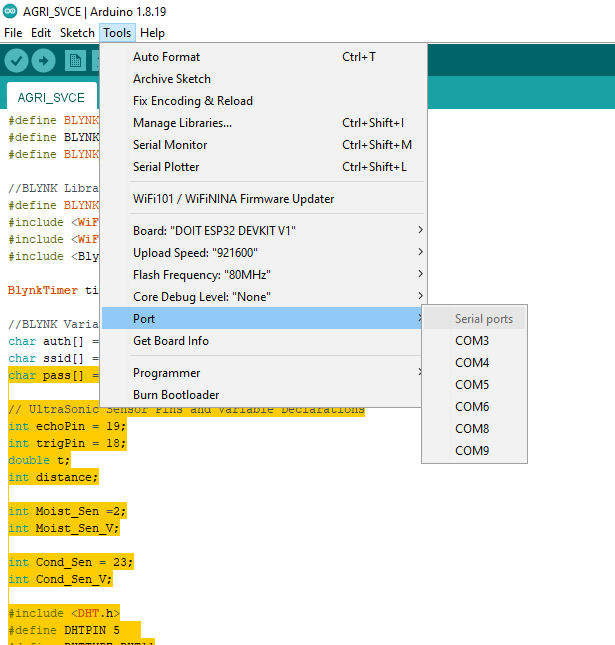
****

Figure 3.4.4 Arduino Port Number

**CHAPTER 4**

**SYSTEM DESIGN AND ANALYSIS**

**4.1 SYSTEM ARCHITECTURE**

The systematic process of constructing a system includes procedures such as planning, analysis, design, deployment, and maintenance. System analysis is the process of collecting and analysing data, identifying problems, and disassembling a system into its constituent elements. System analysis is performed to analyse a system or its components in order to determine its objectives. It is a problem-solving strategy that improves the system and ensures that all of its pieces perform efficiently to accomplish their intended functions. The analysis determines the best course of action for the system.

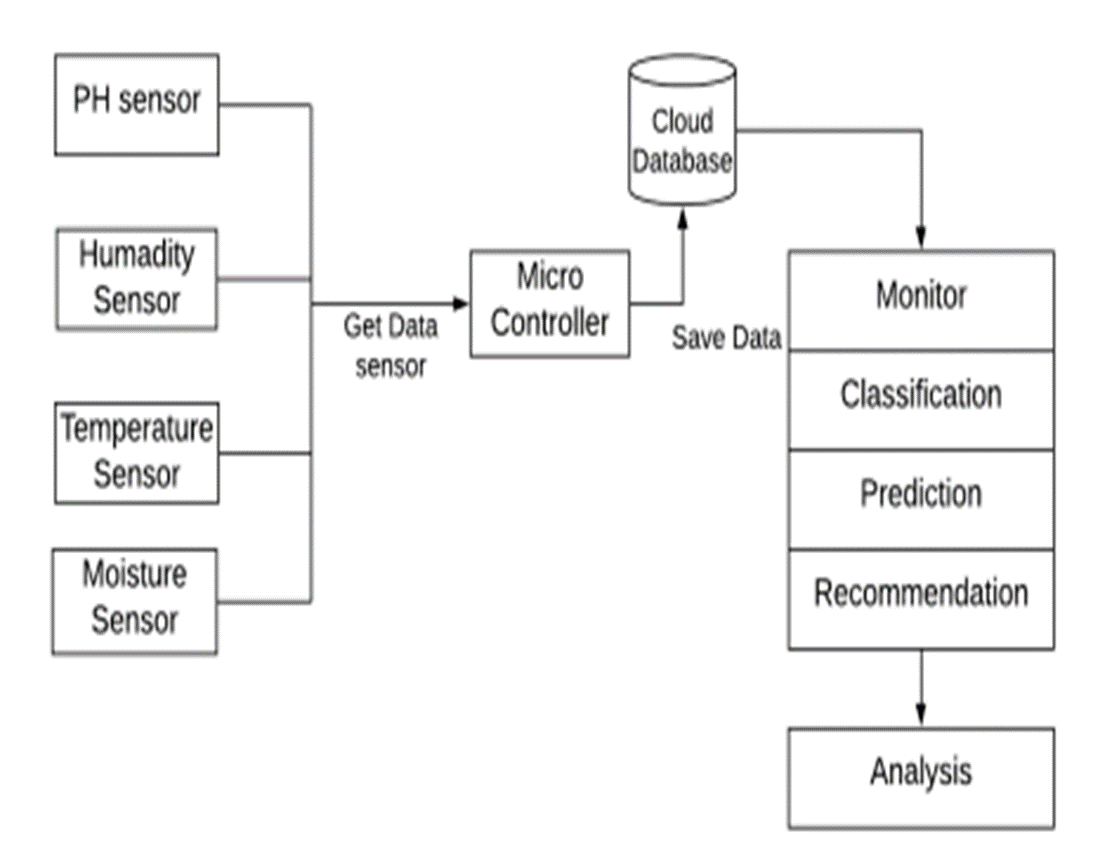


Figure 4.1 System Functionalities

**4.2 FLOWCHART**

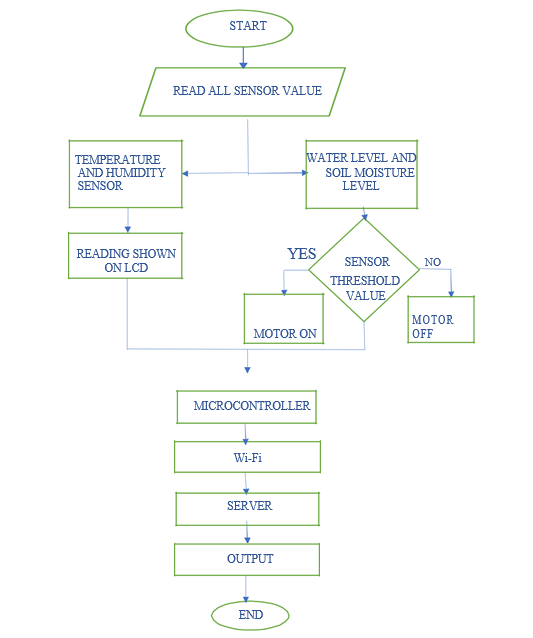
****

Figure 4.2 Flowchart

**4.3 CIRCUIT DIAGRAM**

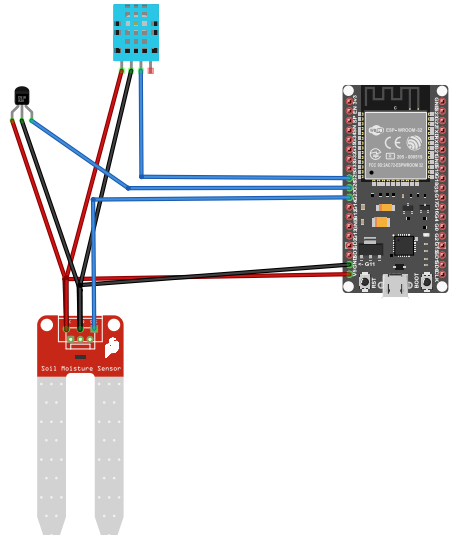
****

Figure 4.3 Circuit diagram

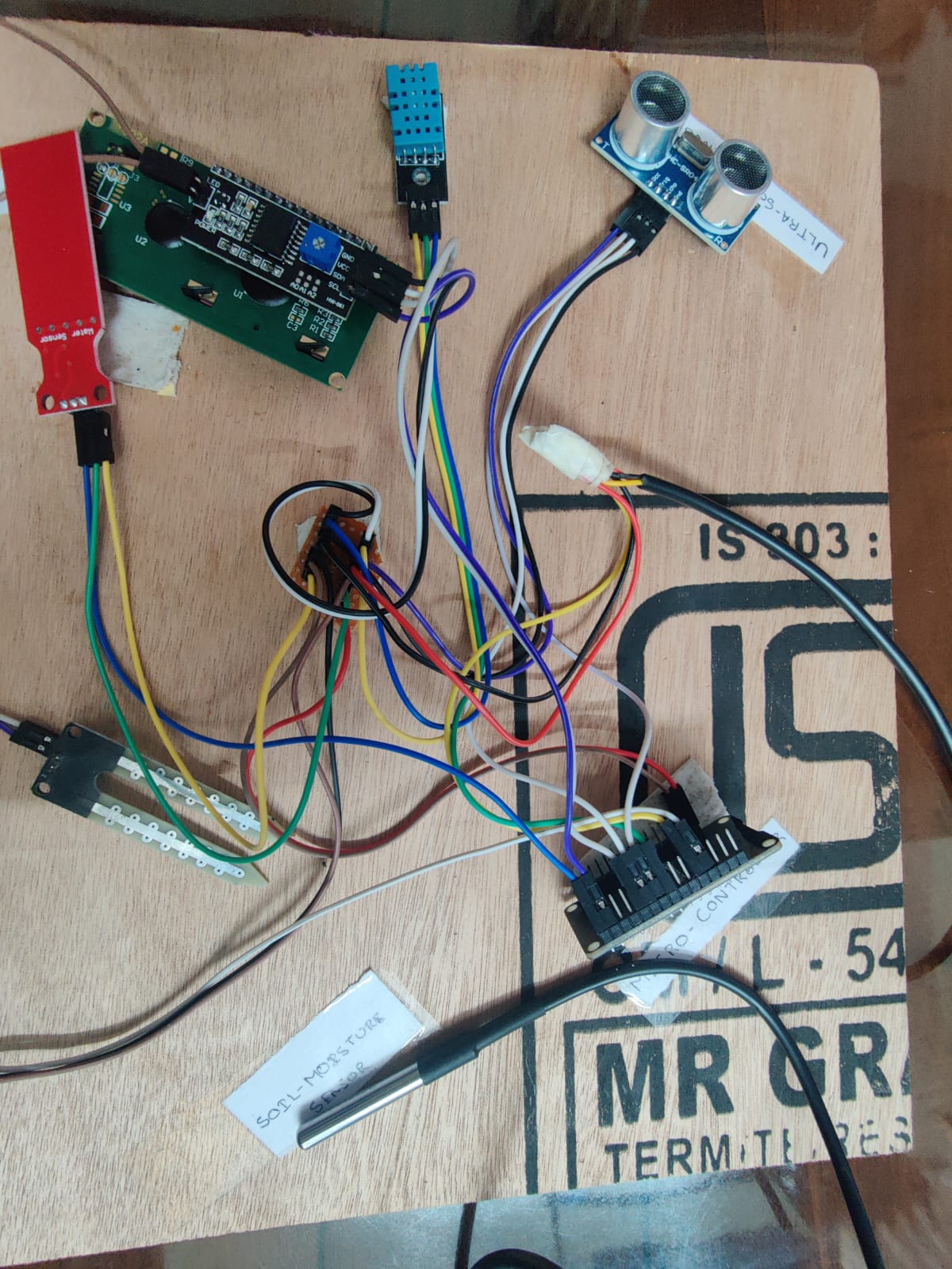
****

Figure 4.3.1 Circuit Implementation

**4.4 ARDUINO CODE:**

#define BLYNK\_TEMPLATE\_ID "TMPLCzlw2xQl"

#define BLYNK\_TEMPLATE\_NAME "MEDECATION"

#define BLYNK\_AUTH\_TOKEN "ai0Tq3u5tBMjGTGhExT7vTyClWHq9x6y"

//BLYNK Library Initalization.

#define BLYNK\_PRINT Serial

#include <WiFi.h>

#include <WiFiClient.h>

#include <BlynkSimpleEsp32.h>

BlynkTimer timer;

//BLYNK Variables

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "Redmi";

char pass[] = "Helloworld";

// UltraSonic Sensor Pins and Variable Declarations

int echoPin = 19;

int trigPin = 18;

double t;

int distance;

int Moist\_Sen =2;

int Moist\_Sen\_V;

int Cond\_Sen = 23;

int Cond\_Sen\_V;

#include <DHT.h>

#define DHTPIN 5

#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

float temp;

float humid;

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27,16,2);//SCL 22 //SDA 21

#include <OneWire.h>

#include <DallasTemperature.h>

#define ONE\_WIRE\_BUS 4

OneWire oneWire(ONE\_WIRE\_BUS);

DallasTemperature sensors(&oneWire);

void setup()

{

Serial.begin(9600);

dht.begin();

sensors.begin();

Blynk.begin(auth, ssid, pass);

pinMode(trigPin,OUTPUT);

pinMode(echoPin,INPUT);

pinMode(Cond\_Sen,INPUT);

pinMode(Moist\_Sen,INPUT);

timer.setInterval(2000,Sensor\_loop);

lcd.init(); // Initialize the LCD

lcd.backlight(); // Turn on the backlight

lcd.clear();

}

void loop ()

{

Blynk.run();

timer.run();

}

void Sensor\_loop()

{

//Getting the Ultrasonic Sensor Value.

digitalWrite(trigPin,LOW);

delayMicroseconds(2);

digitalWrite(trigPin,HIGH);

delayMicroseconds(10);

digitalWrite(trigPin,LOW);

t=pulseIn(echoPin,HIGH);

distance=(t\*0.034/2);

Moist\_Sen\_V = digitalRead(Moist\_Sen);

Cond\_Sen\_V = digitalRead(Cond\_Sen);

temp = dht.readTemperature();

humid = dht.readHumidity();

lcd.clear();

Blynk.virtualWrite(V0,"Water level: "+String(distance));

Serial.println("WATER LEVEL : "+String(distance));

lcd.setCursor(0,0);

lcd.print("WL:"+String(distance));

Blynk.virtualWrite(V1,"Soil Moisture: "+String(Moist\_Sen\_V));

Serial.println("Soil Moisture: "+String(Moist\_Sen\_V));

Blynk.virtualWrite(V2,"Soil Cond: "+String(Cond\_Sen\_V));

Serial.println("Soil Cond: "+String(Cond\_Sen\_V));

Blynk.virtualWrite(V3,"Temperature: "+String(temp));

Serial.println("Temperature: "+String(temp));

lcd.setCursor(8,0);

lcd.print("T:"+String(temp));

Blynk.virtualWrite(V4,"Humidity: "+String(humid));

Serial.println("Humidity: "+String(humid));

lcd.setCursor(0,1);

lcd.print("H:"+String(humid));

sensors.requestTemperatures();

float temperatureC = sensors.getTempCByIndex(0);

Serial.print("Temperature: ");

Serial.print(temperatureC);

Serial.println("C");

Blynk.virtualWrite(V5,"Water Temperature: "+String(temperatureC));

Serial.println("Water Temperature: "+String(temperatureC));

lcd.setCursor(8,1);

lcd.print("WT:"+String(temperatureC));

delay(3000);

}

**4.5 PYTHON CODE:**

from tkinter import \*

import pandas as pd

from sklearn.tree import DecisionTreeClassifier

# load the dataset

crop\_data = pd.read\_csv("crop\_data.csv")

# separate the features and target variable

X = crop\_data.iloc[:, :-1] # features

y = crop\_data.iloc[:, -1] # target variable

# create a decision tree classifier model

dt\_model = DecisionTreeClassifier(random\_state=42)

# train the model on the data

dt\_model.fit(X, y)

# create a Tkinter window

root = Tk()

root.title("Crop Prediction")

# create labels and entry fields for each input feature

temp\_label = Label(root, text="Temperature:")

temp\_label.grid(row=0, column=0)

temp\_entry = Entry(root)

temp\_entry.grid(row=0, column=1)

humid\_label = Label(root, text="Humidity:")

humid\_label.grid(row=1, column=0)

humid\_entry = Entry(root)

humid\_entry.grid(row=1, column=1)

soil\_moist\_label = Label(root, text="Soil Moisture:")

soil\_moist\_label.grid(row=2, column=0)

soil\_moist\_entry = Entry(root)

soil\_moist\_entry.grid(row=2, column=1)

soil\_cond\_label = Label(root, text="Soil Conductivity:")

soil\_cond\_label.grid(row=3, column=0)

soil\_cond\_entry = Entry(root)

soil\_cond\_entry.grid(row=3, column=1)

water\_level\_label = Label(root, text="Water Level:")

water\_level\_label.grid(row=4, column=0)

water\_level\_entry = Entry(root)

water\_level\_entry.grid(row=4, column=1)

# create a function to predict the crop

def predict\_crop():

# get input values from the user

temp = float(temp\_entry.get())

humid = float(humid\_entry.get())

soil\_moist = float(soil\_moist\_entry.get())

soil\_cond = float(soil\_cond\_entry.get())

water\_level = float(water\_level\_entry.get())

# make a prediction with the model

new\_data = pd.DataFrame({'Temperature': [temp], 'Humidity': [humid], 'Soil Moisture': [soil\_moist], 'Soil Conductivity': [soil\_cond], 'Water Level': [water\_level]})

prediction = dt\_model.predict(new\_data)

# display the predicted crop

result\_label.configure(text="Predicted crop: " + prediction[0])

# create a button to predict the crop

predict\_button = Button(root, text="Predict Crop", command=predict\_crop)

predict\_button.grid(row=5, column=1)

# create a label to display the predicted crop

result\_label = Label(root)

result\_label.grid(row=6, column=1)

# run the Tkinter event loop

root.mainloop()

**CHAPTER 5**

**RESULT AND DISCUSSION**

The Tkinter application is the first screen that the user sees in the final implementation of the application. As shown in the Figure, the user can put the values collected from the sensors and predict the suitable crop to determine the suitable food supply in the area.

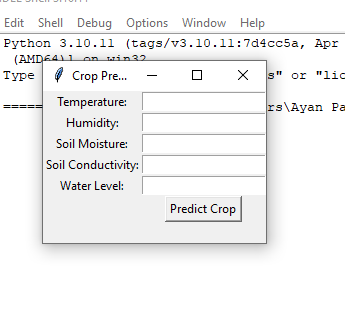
****

Figure 5.1 Tkinter application

The system provides three main functionalities:

Provide information about the crop that is appropriate for the location given the existing environmental conditions and soil quality.

Prepare a set of threshold values for each crop that are suitable environmental conditions for the crop to grow on the specific land.

Compare the following details of existing prepared information with real-time environment information to determine which crop can be grown in the soil.

**TEST CASES**

**Test1-**

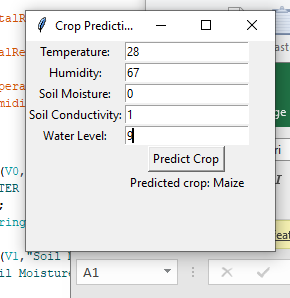
****

Figure 5.2 Test case 1

**Resulted crop under the following climate-Maize**

**Test2-**

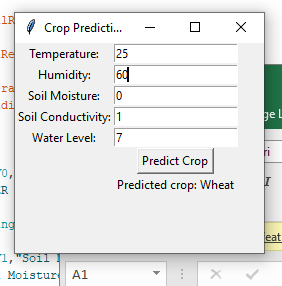
****

Figure 5.3 Test case 2

**Resulted crop under the following climate-Wheat**

**Test3-**

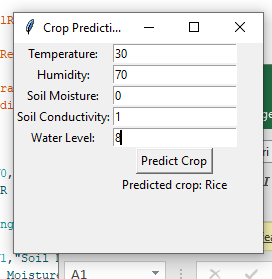


Figure 5.4 Test case 3

**Resulted crop under the following climate-Rice**

**CHAPTER 6**

**CONCLUSION AND FUTURE SCOPE**

This strategy is suggested to address the rising number of farmer suicides while also assisting farmers in their financial development. The Crop Recommender system assists farmers in making crop selection decisions as well as yield projections for specific crops. Furthermore, it educates the user on where to plant the police and which crop is best for the environment. Relevant datasets were gathered, analysed, and trained using machine learning algorithms.IoT devices will monitor real-time climate impacts and forecast crop yields in the future.

The system tracks the user's location and uses that information to retrieve data from the backend. As a result, the user is only required to provide a few parameters, such as the region and soil type.

The next work will be focused on supplying the sequence of crops to be produced based on soil and weather conditions, as well as updating the datasets on a regular basis to give reliable predictions. The Future Work envisions a fully automated system that will accomplish the same thing. Another feature that we are attempting to achieve is the provision of the appropriate fertilisers for the given crop and area. This will require research on fertilisers and their interactions with soil and climate. We also hope to anticipate crisis situations, such as the recent increase in crop prices.

**CHAPTER 7**

**REFRENCES**

1. Tragos, E. Z., Angelakis, V., Fragkiadakis, A., Gundlegard, D., Nechifor, C. S., Oikonomou, G & Gavras, A. (2014, March). Enabling reliable and secure IoT-based smart city applications. In 2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS) (pp. 111-116). IEEE.
2. Shah, J., & Mishra, B. (2016, January). IoT enabled environmental monitoring system for smart cities. In 2016 International Conference on Internet of Things and Applications (IOTA) (pp. 383-388). IEEE.
3. Pasha, S. (2016). ThingSpeak based sensing and monitoring system for IoT with Matlab Analysis. International Journal of New Technology and Research, 2(6).
4. Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012, December). Future internet: the internet of things architecture, possible applications and key challenges. In 2012 10th international conference on frontiers of information technology (pp. 257-260). IEEE.
5. https://components101.com/development-boards/nodemcu-esp8266-pinout-features- and-datasheet
6. Kumar, N. S., Vuayalakshmi, B., Prarthana, R. J., & Shankar, A. (2016, November). IOT based smart garbage alert system using Arduino UNO. In 2016 IEEE Region 10 Conference (TENCON) (pp. 1028-1034). IEEE.
7. Kumar, S., & Jasuja, A. (2017, May). Air quality monitoring system based on IoT using Raspberry Pi. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 1341-1346). IEEE.