

AN EFFICIENT AND INTELLIGENT DECISION MAKING FOR SUSTAINABLE SOIL SOLUTION

**Vishal Prakash Birari^{*1}, Gaurav Ambarsing Pawar^{*2}, Dnyaneshwar Komal Solanke^{*3},
Payal Jivan Pawar^{*4}**

^{*1,2,3,4}Dept. Of CSE, MIT polytechnic Dhanore, Yeola, India.

DOI : <https://www.doi.org/10.56726/IRJMETS49125>

ABSTRACT

Fertilizer use is typically under the limited control of farmers. For the farmers to achieve higher yields and reduce fertilizer loss, competent guidance is required for the best use of these fertilizers. Additionally, there is a connection between rainfall volume and nutrient loss for various fertilizer applications after each rainfall event. Rainfall that is moderate and falls at the right moment can help nutrients penetrate the soil's rooting zone and dissolve dry fertilizer. However, too much rain can increase the possibility of runoff and the pace at which nutrients like nitrogen (N) which is quintessential, phosphorus (P), and potassium (K) which are crucial, manganese (Mn), and boron (B) that are present in the soil. This research presents nutrient recommendations using an updated iteration of the random forest algorithm which is based on time-series data to forecast the required quantity of nutrients for various crops by examining rainfall patterns and crop fertility.

I. INTRODUCTION

Agriculture plays a very important role in national economic growth. Agriculture contributes 17-18% to India's GDP and ranks second worldwide in farm outputs. Plants need fertilizers and fertilizers replace the nutrients which crops take from the top layer of the soil. The absence of fertilizers can cause a drastic reduction in the volume of crop output. But fertilization requires precise action. Rainfall patterns and the amount of nutrients needed for a certain crop must be considered when using fertilizers. Machine learning is the current technology that can solve this problem by using available data for crop fertility and rainfall. Farmers can greatly benefit from the support of robust information about crops. The proposed model also uses a machine learning algorithm (random forest algorithm with k-fold cross-validation technique) and takes two inputs from the user that are crop and location.

II. LITERATURE SURVEY

A comprehensive study of the available literature presents a catalog of previous studies to address this issue. The authors show in [1] that predicting fertilizer usage can assist farmers to attain a proper yield with little waste by preventing toxicity and deficiency in plants to some extent. Paper [2] makes use of fuzzy logic systems that enable the reduction of fertilizer usage which results in an increase in crop productivity. Additionally, [10] shows that the enhanced efficiency of fertilizers is not sufficient for complications that can be caused by compaction. These issues can be prevented by improving the fertilizer recommendation which requires the establishment of a quantifiable relation under N and P for fertilizer usage, in terms of agricultural yield, nitrogen need, and nitrate remnant level which is shown in [11] and paper [4] seconds this by providing a comprehensive measure to estimate the weightage of nutrient requirements and also the role of the chemical properties of soil.

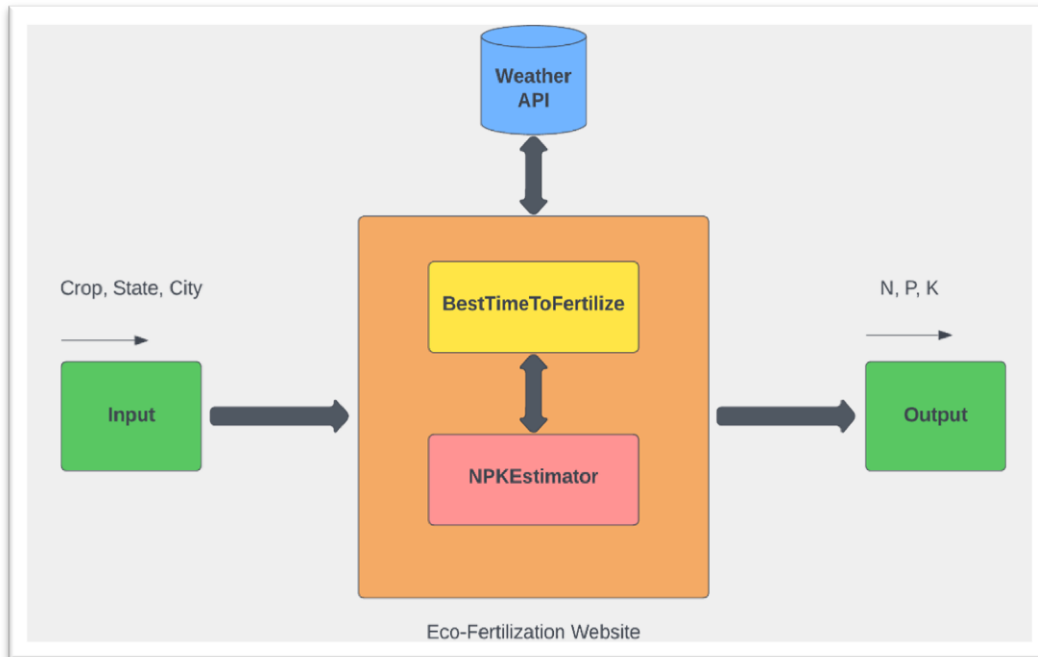
III. SYSTEM DESIGN

The process of defining a system's architecture, modules, components, interfaces, and data in order to meet predetermined requirements is known as system design. We could think of systems design as the applications of systems approach to the creation of products.

3.1 Architectural Design

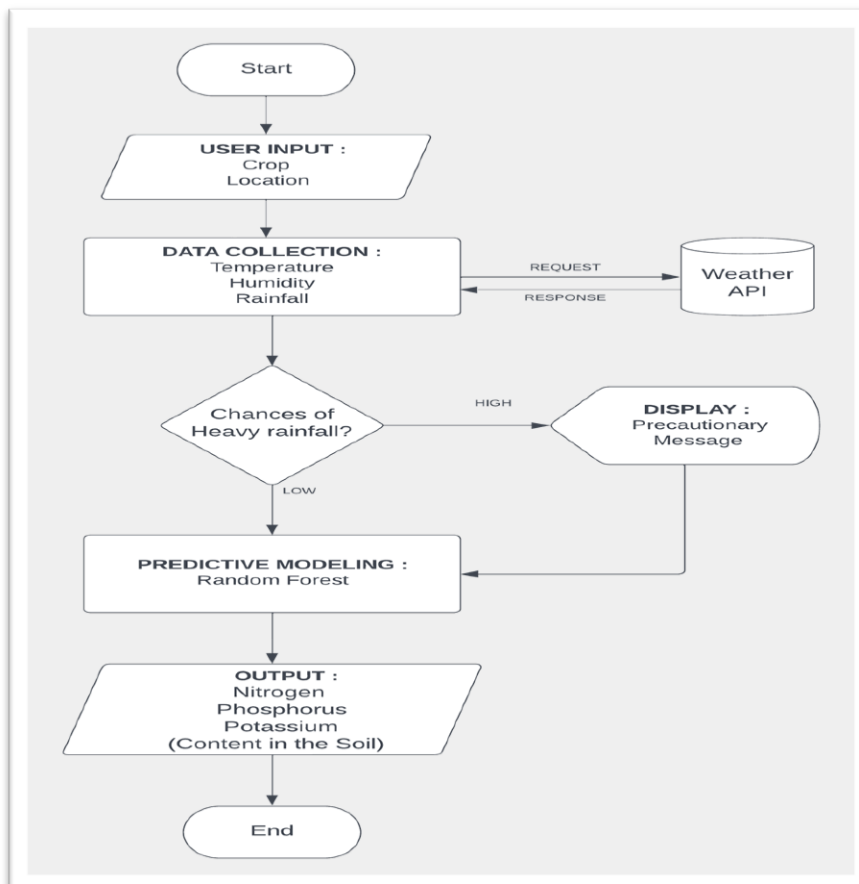
A conceptual model known as system architecture describes the structure and behavior of the system. It consists of the system's elements and the connections between them that explain how the whole system is implemented. The Fig 3.1 below shows the system's architecture and the various components added to them.

3.2 Data Flow Diagram



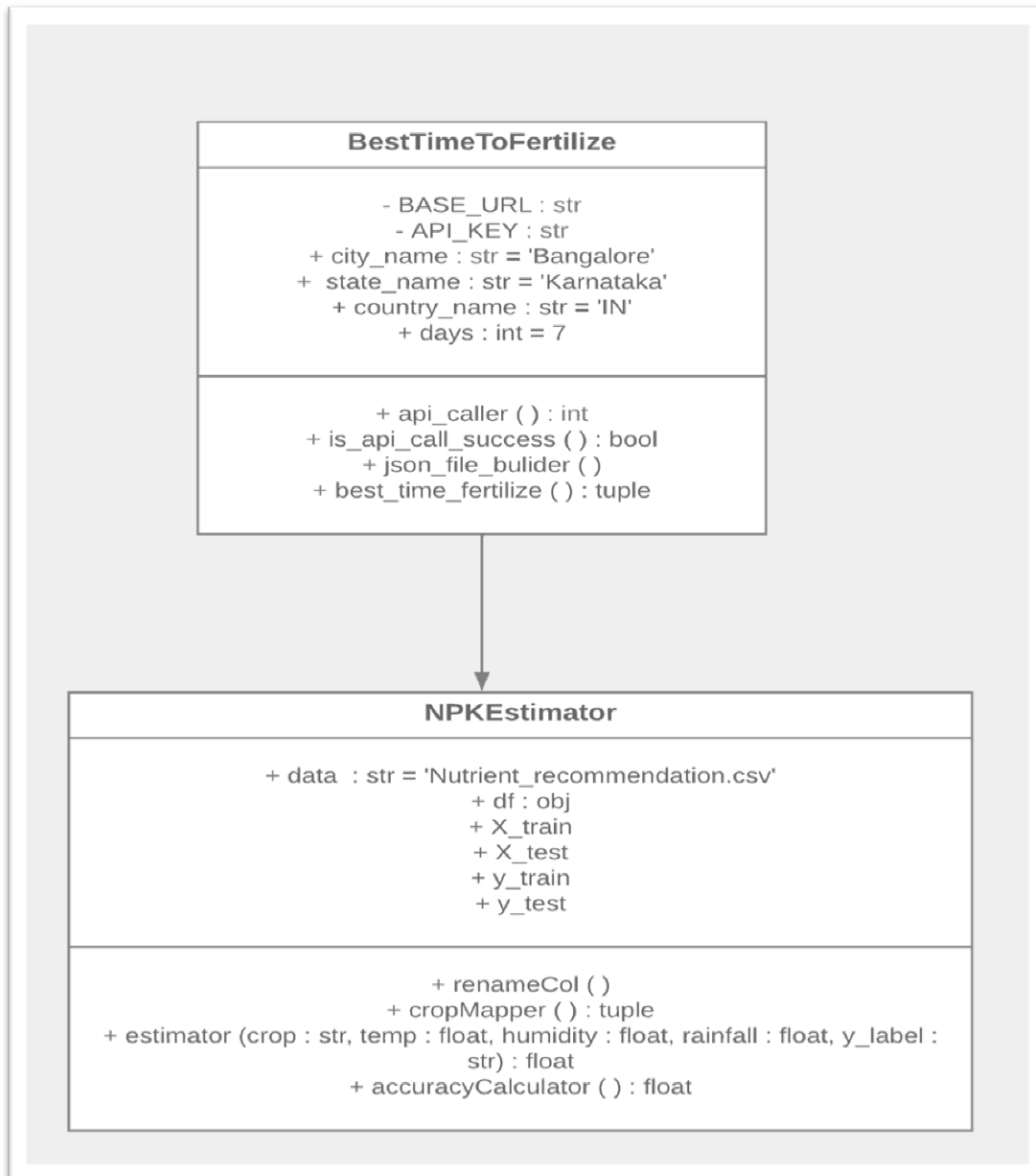
A data flow diagram is a visual representation of how data "flows" throughout a data system, simulating certain features of its operation. It is frequently used as an initial stage to develop, without going into great depth, an overview of the system that may then be expanded upon. They may also be utilized to display data processing.

3.3 Class Diagram



Static diagrams include class diagrams. It represents the application's static view. Class diagrams are used to create executable code for software applications as well as for visualizing, explaining, and documenting various elements of systems.

A collection of classes, interfaces, affiliations, collaborations, and constraints are displayed in a class diagram. A structural diagram is another name for it.



IV. IMPLEMENTATION

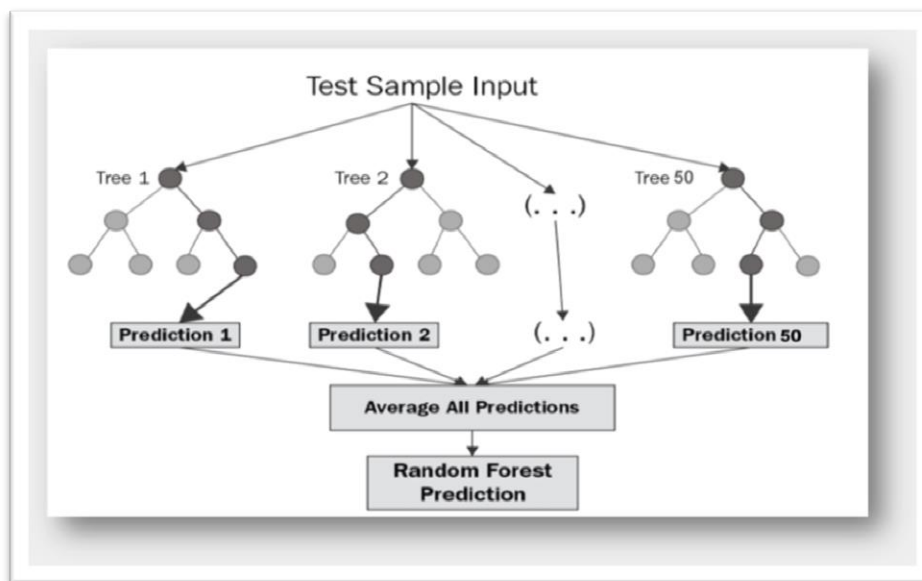
System implementation builds system pieces that adhere to user requirements of the system requirements founded in the early life cycle stages using the framework generated throughout architectural design and the outcomes of system analysis. These system components are then combined to create intermediate aggregates, which ultimately result in the entire system-of-interest (SoI). The system hierarchy's lowest-level system components are really produced by the implementation process (system breakdown structure). System components are created, purchased, or recycled. Production includes the forming, removing, connecting, and finishing processes used in hardware fabrication, the coding and testing processes used in software realization, or the processes used to build operating procedures for the duties of operators.

A design method known as "modular design," sometimes known as "modularity in design," separates a system

into smaller components known as modules or skids that may be independently produced and then used in multiple systems. Functional division into distinct, scalable, reusable modules; strict usage of very well modular interfaces; and adherence to industry norms for interfaces are characteristics of a modular system.

Random Forest Regression

A group of several decision trees called a random forest (RF) are trained using different subsets of data and have changeable hyper-parameters. In our project, we are going to take crop and location as input, and based on it, we will predict the value of N, P, and K. First, we will divide our dataset into training and test datasets, where the training dataset is 80% of the original data and the rest 20% is test data. Then we will create three different random forests of size 50 (decision tree) for each N, P, and K and produces the average of the classes as the overall tree projection



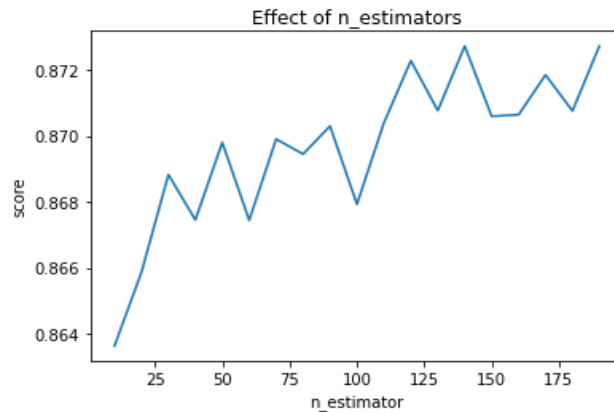
Random Forest Regression

BEGIN:
Step 1: The dataset of size $n = 2200$ is divided into training and test dataset (where the training set is 80% and the test set is 20% that is training set=1,760 and the test set=240).
Step 2: Apply random forest regression to each N, P and K (Nitrogen, Phosphorus & Potassium) value with n estimators=50 (n estimators is the number of decision trees).
Step 3: Train the N Label, P Label and K Label with the training dataset and dependent variable (Where the dependent variable is N for N Label, P for P Label and K for K Label).
Step 4: Each N Label, P Label and K Label generates a 50-decision tree as an output based on the training dataset.
END

Random Forest Regression Algorithm

Why we selected 50 decision trees ($n_{estimator} = 50$) for each label?

We have tested for different $n_{estimator}$ values, but the upmost accuracy achieved for N_Label is 0.87 for two decimal digit precision. As shown in below figure.



Effect of n_estimator

Why is a Random Forest chosen instead of a Decision Tree for this project?

Decision trees are trees that show all possible consequences of a selection using a branched technique, which is a key distinction between them and the random forest algorithm. In contrast, a set of decision trees that follow the output are produced by the random forest method.

In general, adding more trees will increase performance and predictability while decreasing calculation speed. The end solution for regression issues is the mean of all the trees. The samples in the tree target cell is the initial level of means in a random forest method regression model, followed by all trees. In contrast to linear regression, it estimates values beyond the observed range using prior observations.

The accuracy in the decision tree depends on the number of right predictions made divided by the total number of predictions, since it uses huge value attributes at each node, and it produces less accurate results when we apply an algorithm to handle the regression problem in a random forest. Decision trees are greedy and may be deterministic, meaning they produce different answers if we add or remove any additional rows. So, compared to decision trees, random forest forecasts outcomes with higher accuracy.

4.1 Cross-Validation

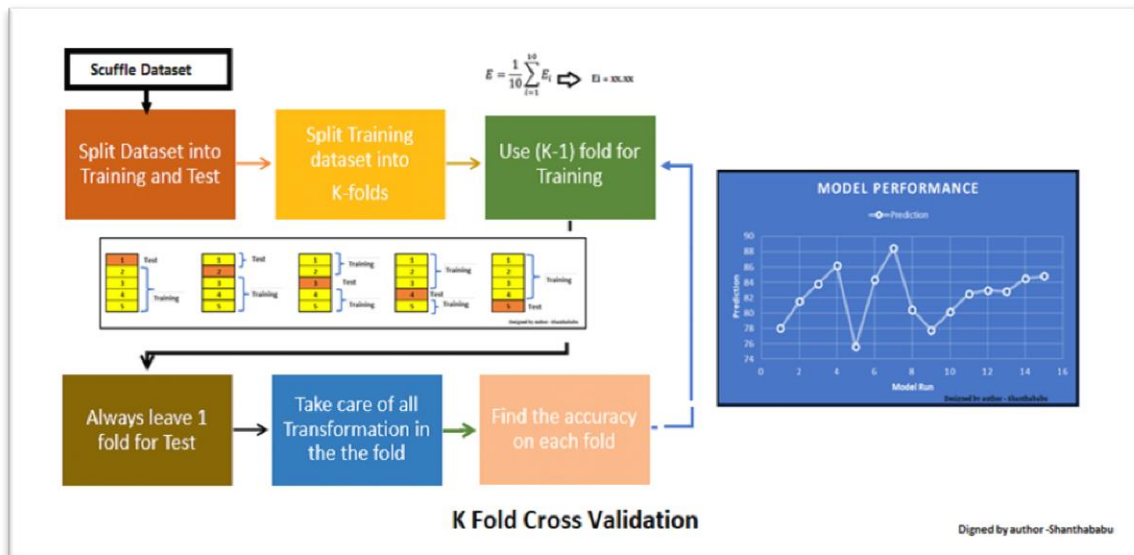
In order to evaluate machine learning algorithm on a small set of data, cross-validation is a re - sampling technique. The algorithm's sole parameter, k, indicates how many groups should be formed from a given data sample. As a result, the technique is frequently referred to as k-fold cross-validation. When a precise value for k is given, it can be substituted for k in the model's regard, such as k=4 for cross-validation that is performed four times.

In applied machine learning, cross-validation is mostly used to gauge how well a machine learning model performs on untrained data. That is, to use a small sample to assess how the model will generally perform when used to generate predictions on data that was not utilized during the model's training.

It is a well-liked technique since it is easy to comprehend and typically yields a less biased or overly optimistic assessment of the model ability than other techniques, including a straightforward train/test split.

Following is the general process:

1. Randomly shuffle the dataset.
2. Create k groups from the dataset.
3. For every distinct group:
 - a. The group should be used as a holdout or test data set.
 - b. As a training dataset contains, use the remaining groupings.
 - c. Adapt a model to the training set, then evaluate it against the test set.
 - d. Keep the evaluation result, but discard the model.
4. Using a sample of quality assessment ratings, summaries the model's skill.



K-Cross Fold Validation

4.2 Input Features

Below are the input features of our system:

- **Crop:** rice, cotton, mango, orange, lentil, etc.
- **Temperature:** temperature measured in Celsius
- **Humidity:** measured relatively in percentages
- **Rainfall:** rainfall in mm

4.3 Output Features

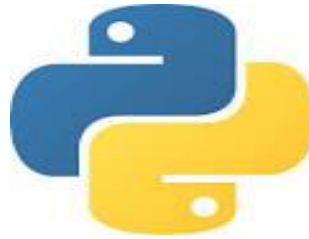
Below are the output features of our system:

- **Label_N:** ratio of soil Nitrogen content
- **Label_P:** ratio of soil Phosphorus content
- **Label_K:** ratio of soil Potassium content

4.4 Python

Python is a high-level, interpreted, general-purpose programming language. It supports a range of programming paradigms, such as functional, object-oriented, and structured programming (particularly procedural). Python is a scripting language that may be used for web applications, including through Apache's mod wsgi. The Web Server Gateway Interface has become a standardised API for these applications. Web

frameworks like Pylons, Django, web2py, Pyramid, Flask, Zope, Tornado, TurboGears, and Bottle enable programmers to create and maintain sophisticated applications. IronPython and Pyjs can be used to create the client-side of Ajax-based applications. One option for a relational database data mapper is SQLAlchemy.



Python logo

4.5 Flask

A micro-web framework built on Python is called Flask. Because it doesn't need any particular tools or libraries, it is known as a microframework. It doesn't have a form validation layer, database abstraction layer, or any other elements that depend on already-built third-party libraries to carry out basic functions. But at the other hand, extensions can be utilized to increase application capabilities as if they had been created in Flask. The following standard framework-related tools all have extensions: upload handling, form validation, object-relational mappers and numerous open authentication protocols.

Python is used to create the Flask web application framework. It is created by Armin Ronacher, the founder of Pocco, a global community of Python fans. The Jinja2 template engine, Werkzeug and WSGI toolkit serve as the foundation for Flask. They're both Pocco projects.

WSGI

Python web application development now adheres to the Web Server Gateway Interface (WSGI) standard. A uniform gateway between both the web applications and web server is described by the WSGI protocol.

Werkzeug

This is a WSGI toolkit that carries out utility operations like requests and response objects. As a result, a web framework may be built on top of it. Werkzeug serves as one of the foundations for the Flask framework.

Jinja2

A well-liked Python templating engine is Jinja2. To create dynamic web pages, a web templating method integrates a template with a specific data source. It's common to hear Flask referred to as a micro framework. It strives to keep an application's core straightforward but flexible. Both a form validation support and an established abstraction-layer for database processing are absent from Flask. Instead, Flask permits the use of extensions to give the application such capability.

4.6 HTML

- Hyper Text Markup Language is what HTML stands for.
- Website and applications are created using HTML.
- Hypertext: "Text within Text" is what hypertext is. A hypertext link can be found within a text. A link is considered a hypertext when it sends users to a new webpage when user click on it. Hypertext can be used to connect two or more web content.
- Markup language: A text document can be formatted and designed using this computer language. It can convert text into pictures, tables, and other formats.
- A web page is typically an HTML document that a web browser translates. Entering the URL will take you to the website. Both static and dynamic web pages are possible.
- Static Pages can be produced using HTML.
- HTML main tags:
 - <!DOCTYPE>: It describes the type of document or informs the browser of the HTML version.
 - <html>: This element informs the browser that the document is an HTML one.

- <head>: The metadata should be contained in the first element of the html element. The body tag must open before it can be closed.
- <title>: It is employed to include the HTML page's title, which is displayed just at top of the web browser.
- <body>: The HTML's primary material is included in here.
- <h1>: That the very first level heading of the page is described in the text inside the <h1> tag.

4.7 CSS

- Cascading style sheets, or CSS for short, is a language that governs how web pages are displayed, including their colors, fonts, and layout, making them more user-friendly.
- The primary use of CSS is to create style sheets for the internet. Even HTML is unrelated to it.
- Let's break down the CSS (acronym):
 - Falling into Style: Cascading
 - Imbuing our HTML tags: Style
 - Using style in a variety of publications' writing: Sheets
- Our HTML file can contain CSS in one of three ways:
 - Inline CSS
 - Internal CSS
 - External CSS
- Considering Priority:
 - Inline (then) Internal (then) External
- Inline CSS:
 - The only way to apply a style.
 - Independence
 - Adheres to each piece clearly.
 - The concept of separating issues has been forgotten.
- Internal CSS:
 - We can use style tags to apply styles in HTML files.
 - Redundancy is eliminated.
 - But the concept of separating issues is still misunderstood.
 - Particularly used on a single document.
- External CSS:
 - We can apply styles with the aid of the link tag in the head tag.
 - References are included.
 - Documents end with a .css (extension).
 - Redundancy is eliminated.
 - The concept of distinct concerns is maintained.
 - Particularly used for each document.
- CSS characteristics:
 - A selector element and a declaration block element make up a style rule.
 - The HTML component that we want to have its style applied to is indicated by the selector.
 - One or more assertions are contained in the declaration block, separated by semicolons.
 - Every declaration that is entered has a semicolon, a value, and a name for the CSS attribute. For instance, if color is the property, then red is the value. The property is font size, and the value is 15 pixels.
 - These blocks are enclosed in curly braces, and the CSS declaration is completed with a semicolon.
 - The ones that are used to find HTML elements dependent on the element id, class, name, attribute, and more

are known as CSS selectors.

4.8 Parallax Effect

- The parallax effect is one of the little tricks that can make our landing page great and get our visitors' attention.
- CSS has matured considerably in the last few years. In short, there is a lot of flexibility in how things are done.
- The parallax effect is when two slices moving at a constant speed move at a constant speed.
- We have seen a parallax effect app in front of a 2D game. In those apps, the background changes slowly compared to the front.
- We have noticed the parallax effect when the object is sitting in a car that is moving slower than other cars on the road.
- Simply put, the conditions that must be met for a parallax effect to occur are:
 - The parallax layer's perception changes, so it appears to move relatively fast or slowly without deviating from its original size.
 - Do not change the position or speed.
- To achieve both of these conditions, you need to increase the perceived distance between the user and the plane before moving along the negative Z axis.
- The longer the apparent distance, the smaller the aircraft may look. To counter this size reduction, you can reduce it to its original size.
- This mainly means using two CSS properties, Perspective and Transform. Use `translateZ ()` and `scale ()` transformation.

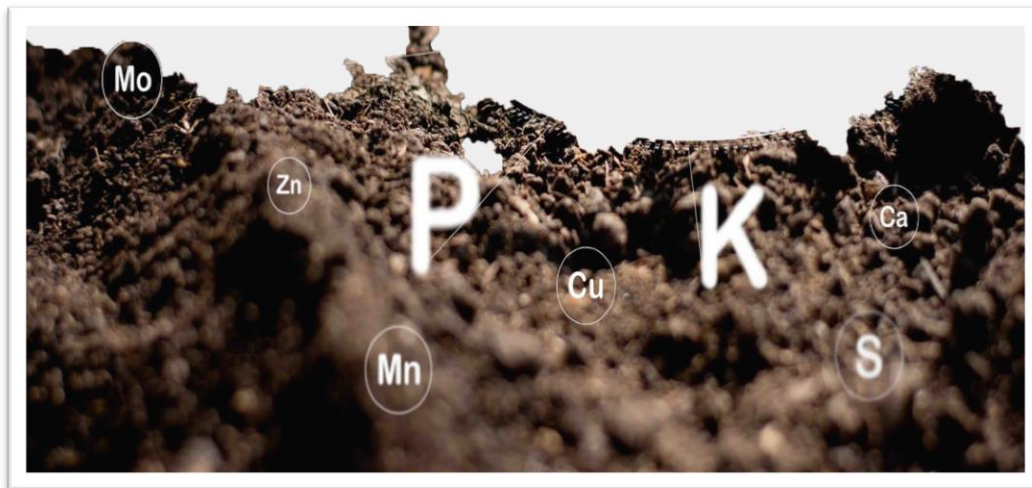
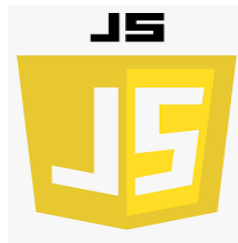


Fig: Parallax Effect Top

4.9 JavaScript

- Website programming languages include JavaScript.
- Recognized for creating websites that are also often used outside of browsers.
- JavaScript is a declarative as well as an imperative language. There is a standard library of objects in JavaScript, including arrays, dates, math, and other types.
- Client-side: Provides objects to manipulate the browser's Document Object Model (DOM).
- For instance, client-side extensions let your program include HTML components.
- React to user events including form submission, mouse clicks, page navigation, etc. as well as forms and user events.
- Convenient client-side libraries include AngularJS, ReactJS, and VueJS.

- Server-side: Provides objects related to the execution of JavaScript on the server.
- Imperative Languages-In this type of language, we are primarily interested in how to do it.
- Controls only the calculation flow. The object-oriented approach, which is a procedural programming approach, is equivalent to asynchronous wait when considering what to do next after an asynchronous call.
- Declarative programming: I'm worried about how to do that in this kind of language. Basically, it requires logical operations. The main goal here is to explain the desired result without directly defining the acquisition method like the arrow function.
- There are two ways to include JavaScript in HTML:
 - Internal JS: As necessary, tags can be inserted inside of other tags..
 - External JS: The JavaScript code can be written in a separate file with a.js extension and linked within the tag of the HTML file we wish to add it to.



V. RESULTS

Eco-Fertilization, a user-friendly system, has been implemented in the form of a website to provide cross-platform functionality and suggest appropriate timings and amount of nutrients required for an inputted crop with alert system for heavy rainfall (as shown in Fig 5.1-5.5).

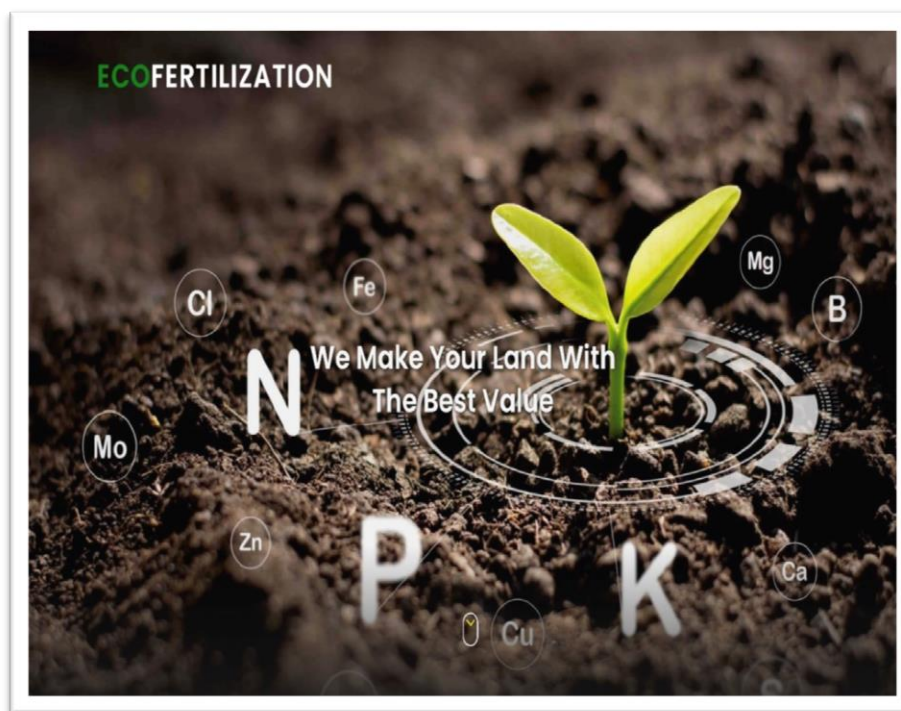
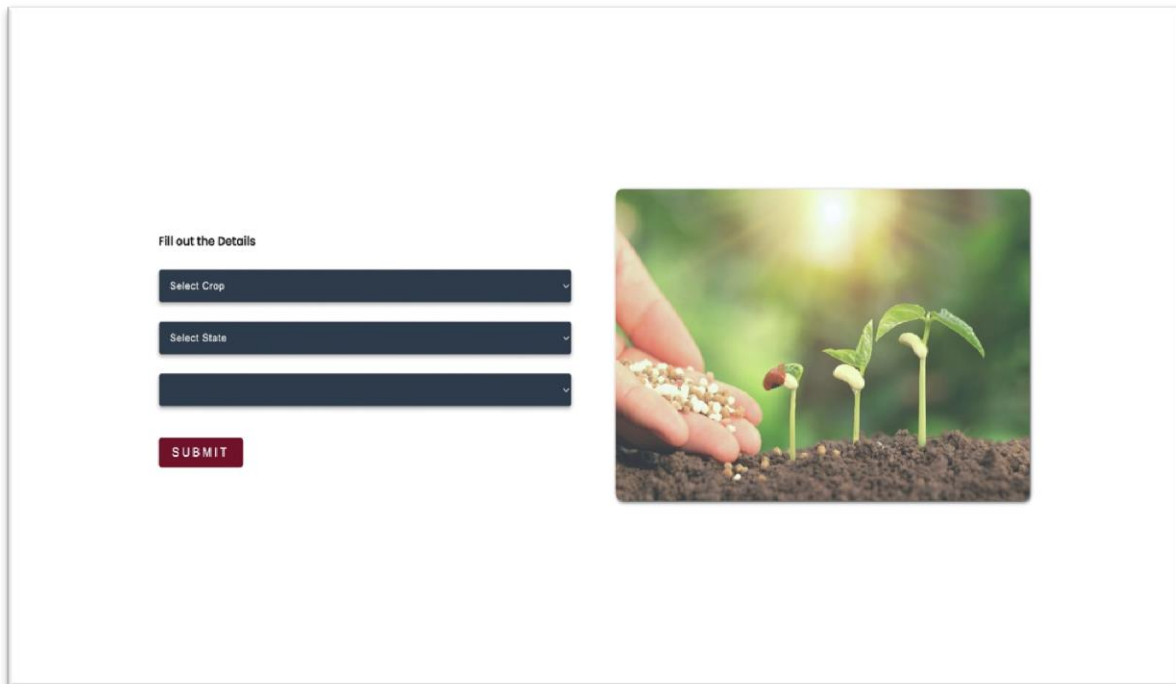


Fig 5.1: Homepage of Eco-Fertilization



Fill out the Details

Select Crop

Select State

SUBMIT


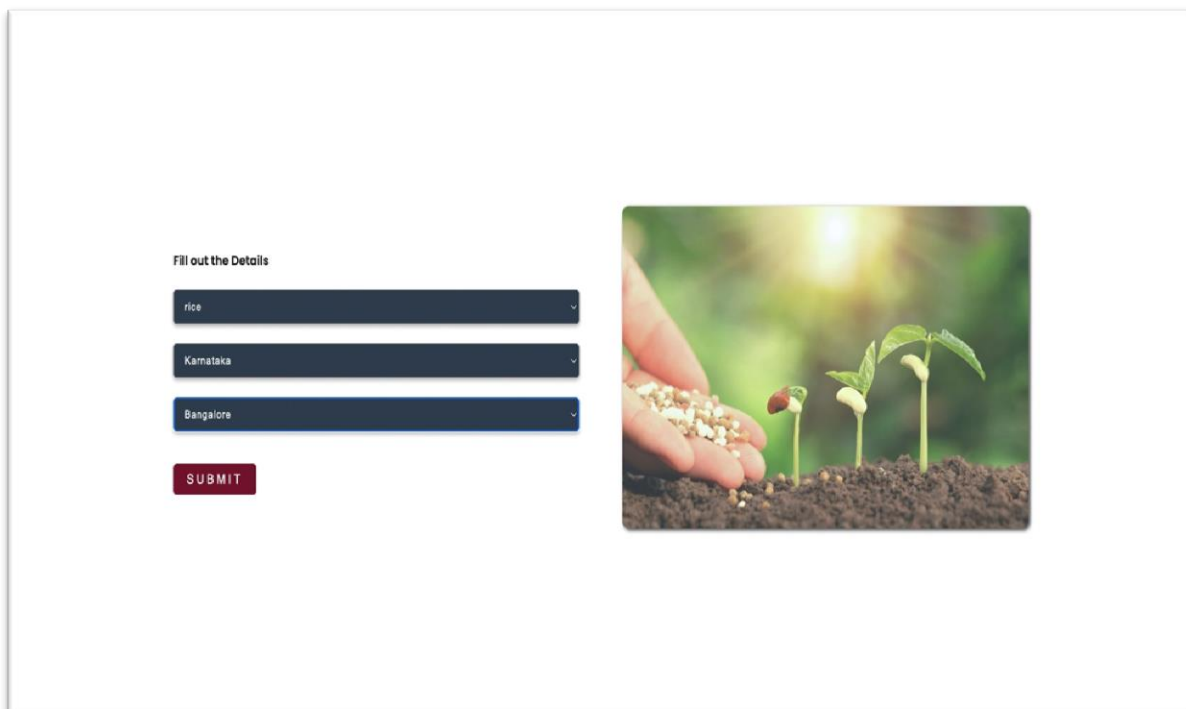


Fig 5.2: Input Form



Fill out the Details

rice

Karnataka

Bangalore

SUBMIT




Fig 5.3: Details filled using the drop-down menu

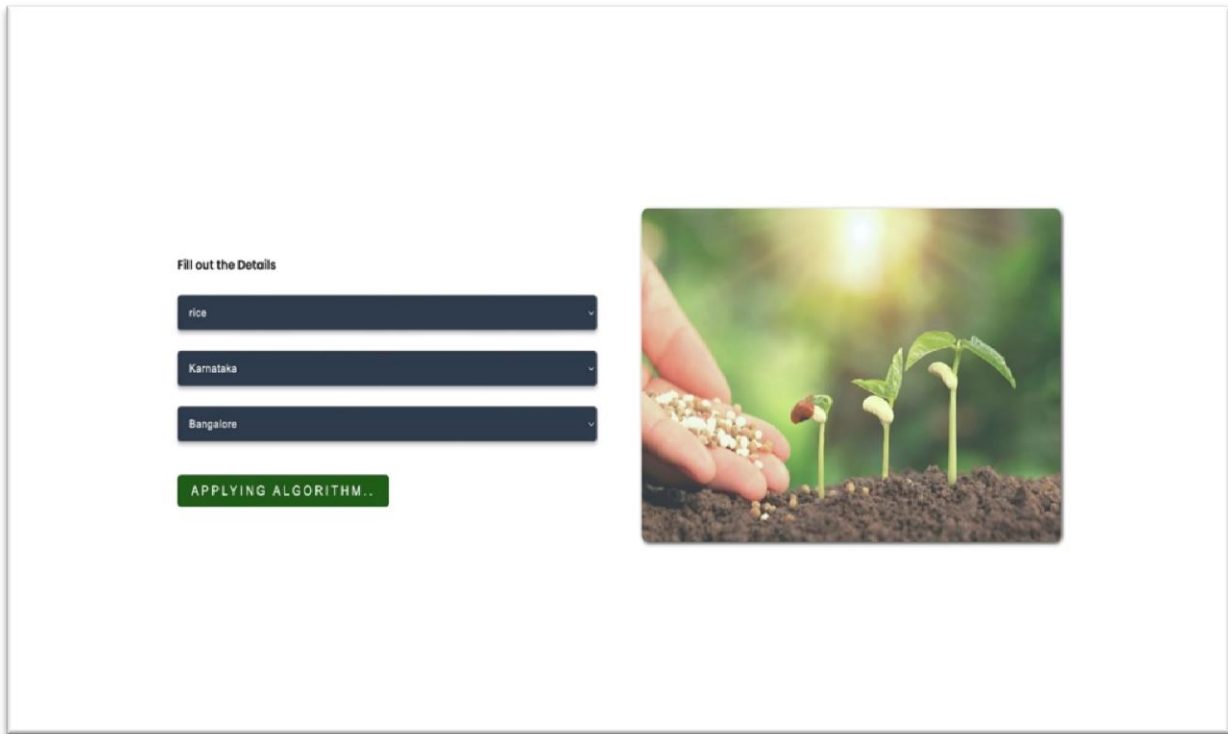


Fig 5.4: Applying Algorithm to inputted details

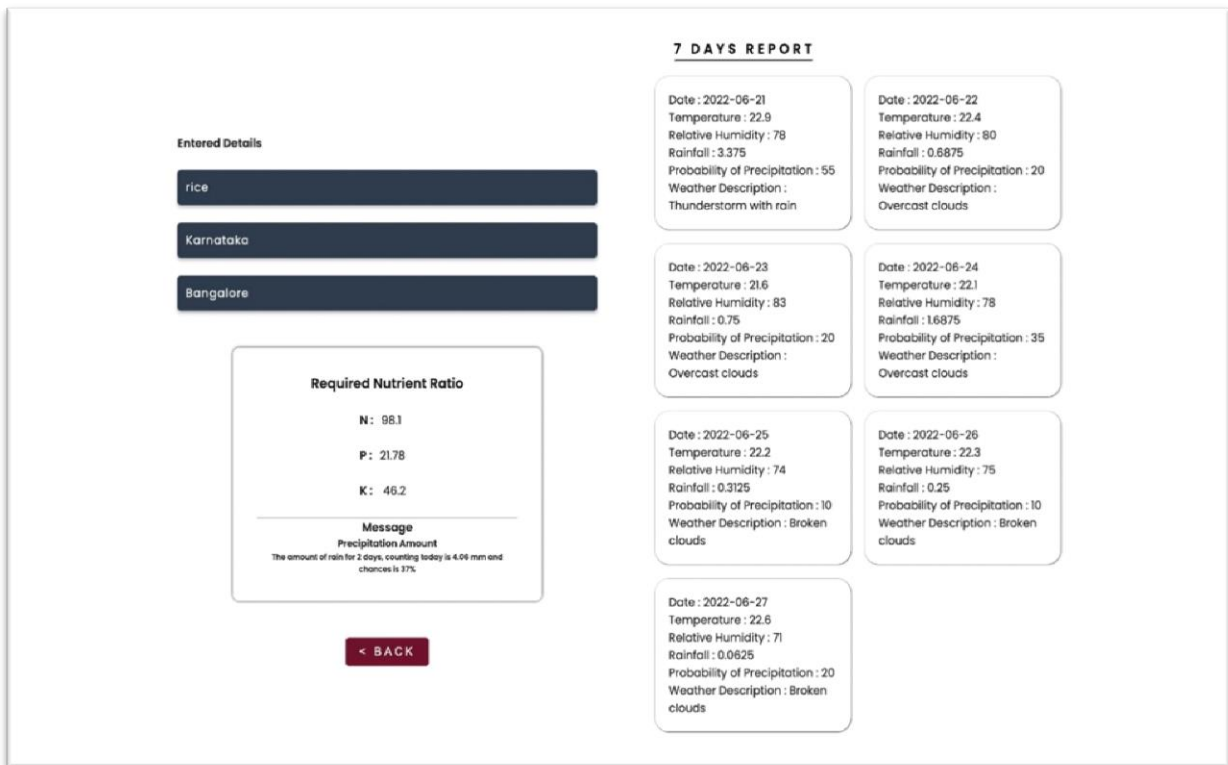


Fig 5.5: Output with seven days of weather forecasts & alerts/messages

VI. APPLICATIONS AND FUTURE WORKS

6.1 Applications

- This project is useful in the agriculture sector.
- Can be used to reduce the wastage of fertilizers.

- Used to suggest nutrient recommendations for the crops.
- It reduces water pollution by slowing down the process of soil leaching as fertilizers can reach the water table and contaminate shallow groundwater and deep aquifers.
- Reduce leaching and runoff potential.
- It provides weather alerts and messages. Alerts are displayed in the output of this application in case of bad weather conditions.
- Seven-day weather forecasts to timely plan the fertilization.

6.2 Future Scope

The proposed system provides a helping hand to our farmers. It gives information about the use and quantity of nutrients required by the crops. There is scope for improvement in the system by providing user interface in the native language, so that the user can operate the system easily if he or she is unfamiliar with the English language. In addition, speech recognition systems can be added to handle illiterate users.

VII. CONCLUSION

The proposed system is able to achieve 92% of accuracy, which is quite good for any predictive model. It provides information about the use and the amount of nutrients required by the crops for satisfactory crop growth and production with respect to weather conditions. It provides weather alerts and messages. Alerts are displayed in the output of this application in case of bad weather conditions. The accuracy can be improved further with development in technologies.

ACKNOWLEDGEMENT

We express gratitude to our institution and management for providing us with good infrastructure, laboratory, facilities and inspiring staff, and whose gratitude was of immense help in completion of this project successfully.

We express our sincere gratitude to **Dr. T N Sreenivasa**, Principal, Atria Institute of Technology, for providing us the required environment and for his valuable suggestion.

Our sincere thanks to **Dr. Aishwarya P**, Head of the Dept. Computer Science and Engineering, Atria Institute of Technology, for her valuable support and for rendering us resources for this project work.

We express our gratitude to **Dr. Manash Sarkar**, Associate Professor, Dept. of Computer Science and Engineering, Atria Institute of Technology, who guided us with valuable suggestions in completing this project at every stage.

Last but not the least, the project would not have been a success without the support of our **parents** and **friends**. Our sincere thanks should be rendered to everyone who helped us in all possible ways.

VIII. REFERENCES

- [1] Krutika Hampannavar, Vijay Bhajantri, Shashikumar G. Totad "Prediction of Crop Fertilizer Consumption," Fourth International Conference on Computing Communication Control and Automation (ICCUBEA),2018, PP.1-5
- [2] G. Prabakaran, D. Vaithyanathan, Madhavi Ganesa, "Fuzzy decision support system for improving the crop productivity and efficient use of fertilizers," Computers and Electronics in Agriculture, vol-150, 2018, PP. 88-97
- [3] Shital Bhojani, Nirav Bhatt, "Data Mining Techniques for Crop Yield Prediction," Computers and Electronics in Agriculture, vol-6, 2018, PP. 357-358
- [4] Yulong Yin, Hao Ying, Huifang Zhen, Qingsong Zhang, Yanfang Xue, Zhenling I, "Estimation of NPK requirements for rice production in diverse Chinese environments under optimal fertilization rate," Agricultural and Forest Meteorology, vol-279, 2019, PP. 1-6
- [5] Laura J.T. Hess, Eve-Lyn S. Hinckley, G. Philip Robertson, Pamela A. Matson, "Rainfall intensification increases nitrate leaching from tilled but not no-till cropping systems in the U.S. Midwest," Agriculture, Ecosystems & Environment, vol-290, 2020, PP. 1-10
- [6] Potnuru Sai Nishant, Pinapa Sai Venkat, Bollu Lakshmi Avinash, B. Jabber, "Crop Yield Prediction Based on Indian Agriculture using Machine Learning," 2020 International Conference for Emerging

- Technology (INCET), 2020, PP. 1-4
- [7] Tony Yang, Kadambot H.M., Siddique, Kui Liu, "Cropping systems in agriculture and their impact on soil health," *Global Ecology and Conservation*, vol-23, year, PP. 1-13
- [8] János Kátai, Ágnes Oláh Zsuposné, Magdolna Tállai, Tarek Alshaal, "Would fertilization history render the soil microbial communities and their activities more resistant to rainfall fluctuations? ," *Ecotoxicology and Environmental Safety*, vol-201, 2020, PP. 1-11
- [9] Usman Ahmed, Jerry Chun-Wei Lin, Gautam Srivastava, Youcef Djenouri, "A nutrient recommendation system for soil fertilization based on Evolutionary Computation," *Computers and Electronics in Agriculture*, vol-189, 2021, PP. 1-7
- [10] A.Hussein, Diogenes L. Antille, Shreevatsa Kodur, GuangnanChen, Jeff N.Tullberg, "Controlled traffic farming effects on productivity of grain sorghum, rainfall and fertilizer nitrogen use efficiency," *Journal of Agriculture and Food Research*, vol-3, 2021, PP. 1-17
- [11] Zujiao Shi, Donghua Liu, Miao Liu, Muhammad Bilal Hafeez, Pengfei Wen, Xiaoli Wang, Rui Wang, Xudong Zhang, Jun Li, "Optimized fertilizer recommendation method for nitrate residue control in a wheat-maize double cropping system in dryland farming," *Field Crops Research*, vol-271, 2021, PP. 1-10.
- [12] Janmejy Pant, R.P. Pant, Manoj Kumar Singh, Devesh Pratap Singh, Himanshu Pant, "Analysis of agricultural crop yield prediction using statistical techniques of machine learning," *Materials Today: Proceedings*, vol-46, 2021, PP. 1-10
- [13] Benny Antony, "Prediction of the production of crops with respect to rainfall,," *Environmental Research*, vol-202, 2021, PP. 1-5
- [14] Akash Manish Lad, K. Mani Bharathi, B. Akash Saravanan, R. Karthik, "Factors affecting agriculture and estimation of crop yield using supervised learning algorithms," *Materials Today: Proceedings*, 2022, PP. 1-10
- [15] Raves Akhtar, Shabbir Ahmad Sofi, "Precision agriculture using IoT data analytics and machine learning," *Journal of King Saud University - Computer and Information Sciences*, 2021, PP. 1-17
- [16] Saheed Garnaik, Prasanna Kumar Samant, Mitali Mandal, Tushar Ranjan Mohanty, Sanat Kumar Dwibedi, Ranjan Kumar Patra, Kiran Kumar Mohapatra, R.H. Wanjari, Debadatta Sethi, Dipaka Ranjan Sena, Tek Bahadur Sapkota, Jagmohan Nayak, Sridhar Patra, Chiter Mal Parihar, Hari Sankar Nayak, "Untangling the effect of soil quality on rice productivity under a 16-years long-term fertilizer experiment using conditional random forest," *Computers and Electronics in Agriculture*, vol-197,2022, PP. 1-10
- [17] Rubby Aworka, Lontsi Saadio Cedric, Wilfried Yves Hamilton Adoni, Jérémie Thouakesseh Zoueu, Franck Kalala Mutombo, Charles Lebon Mberi Kimpolo, Tarik Nahhal, Moez Krichen, "Agricultural decision system based on advanced machine learning models for yield prediction: Case of East African countries," *Smart Agricultural Technology*, vol-3, 2022, PP. 1-9
- [18] Senthil Kumar Swami Durai, Mary Divya Shamili, "Smart farming using Machine Learning and Deep Learning techniques," *Decision Analytics Journal*, vol-2, 2022, PP. 1-30.