# INTEGRATED DEEP NEURAL NETWORKS AND ENSEMBLE MODELS FOR ACCURATE SOIL MOISTURE PREDICTION AND FERTILIZER GUIDANCE

**A PROJECT REPORT**

***Submitted by***

**NISHANTH M (917721C060)**

**SRIVARSAN S (917721C100)**

***In partial fulfillment of the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**

**THIAGARAJAR COLLEGE OF ENGINEERING, MADURAI – 15**

(A Government Aided Autonomous Institution Affiliated to Anna University)

**ANNA UNIVERSITY: CHENNAI 600 025**

**MAY 2025**

# THIAGARAJAR COLLEGE OF ENGINEERING
# MADURAI-15

(A Government Aided Autonomous Institution Affiliated to Anna University)

## BONAFIDE CERTIFICATE

Certified that this project report "**Integrated Deep Neural Networks and ensemble models for accurate soil moisture prediction and fertilizer guidance"** is the bonafide work of **"NISHANTH M (917721C060), SRIVARSAN S (917721C100)"** who carried out the project work under our supervision during the Academic Year 2024-2025.

**SIGNATURE**

Dr. S. Mercy Shalinie, M.E., Ph.D.,
**HEAD OF THE DEPARTMENT**
COMPUTER SCIENCE AND ENGINEERING

THIAGARAJAR COLLEGE OF ENGG
MADURAI – 625 015

**SIGNATURE**

Dr. M. Nirmala Devi, M.E., Ph.D.,
**ASSISTANT PROFESSOR (SELECTION GRADE)**
COMPUTER SCIENCE AND ENGINEERING

THIAGARAJAR COLLEGE OF ENGG
MADURAI – 625 015

Submitted for the VIVA VOCE Examination held at Thiagarajar College of Engineering on ………………

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

# ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude to **Dr. L. Ashok Kumar**, Principal of Thiagarajar College of Engineering for his support and encouragement throughout this project work.

I wish to express my sincere thanks to **Dr. S. Mercy Shalinie**, Head of the Department of Computer Science and Engineering for her support and ardent guidance.

I owe my special thanks and gratitude to **Dr. M. Nirmala Devi,** Assistant Professor (Selection Grade) of Computer Science and Engineering for their invaluable mentorship and guidance during our academic pursuits.

I am also indebted to all the teaching and non-teaching staff members of our college for helping us directly or indirectly by all means throughout the course of our study and project work.

I extremely thank my parents, family members and friends for their moral support and encouragement for our project.

# ABSTRACT

Soil moisture is a crucial parameter of agricultural production, which has an impact on crop development, water management, and fertilizer use. Accurate soil moisture forecasting enables the usage of fertilizer in the most suitable manner, leading to enhanced resource utilization and better crop yield. The study introduces a hybrid approach that uses ensemble learning (Random Forest, XGBoost, LightGBM and Deep Neural Networks (DNN) for precise forecasting of soil moisture and fertilizer recommendation. In order to ensure dependable data, fertilizer and soil moisture data are label encoded and normalized. The ensemble models are averaged with weighted averaging and 80-20 train-test splits are used for training. The fertilizer recommendation system, based on a multi-layer DNN, incorporates soil parameters, crop, and forecasted moisture levels to develop optimal suggestions. Experimental results indicate an excellent $R^2$ value, highly improving the accuracy of soil moisture prediction. The DNN based fertilizer recommendation system also shows high accuracy, thereby qualifying as an automated and scalable precision agriculture solution. The machine-learning based architecture allows for predictive decision-making, full automation, and increased efficiency in fertilizer recommendation and soil moisture monitoring. With the application of ensemble learning and deep-learning, this work promotes data-driven precision agriculture, fostering sustainable agriculture. The ensemble model we developed for soil accuracy gave us a training accuracy of 95% and testing accuracy of 92%. The fertilizer recommendation model gave training accuracy of 94.4% and testing accuracy of 91.2%.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1
# INTRODUCTION

Agriculture is crucial in ensuring food security globally, thus efficient use of resources is significant. Soil water content is very significant in the growth and production of plants because it significantly impacts fertilizer efficiency, nutrient availability, and water uptake. Traditional means of measuring soil water content include in-situ sensors and hand sampling, and these are labor-intensive, expensive, and lack scalability. With the advent of artificial intelligence and machine-learning, predictive models are powerful means for forecasting soil moisture and making farming decisions. When soil moisture is predicted, optimized irrigation planning and fertilizer application are achievable, minimizing wastage of resources and maximizing crop yield. Ensemble learning models and deep neural networks, both machine-learning methods, have been demonstrated to have significant potential in enhancing the accuracy of soil moisture estimation. Ensemble learning combines a set of predictive models to greatly enhance performance, and deep-learning methods enable the recognition of numerous fine-grained patterns, and thus both of them are suitable for all data-driven recommendations. This specific study combines ensemble learning for predicting soil water as well as employs deep learning to guide fertilizer decisions carefully. The strengths of our Random Forest, XGBoost, and LightGBM are perfectly combined by our soil moisture prediction model to achieve exceptionally high accuracy. The computed soil moisture values are taken as input to a fertilizer recommendation system with a neural network and enabled through data driven and accurate recommendations for agricultural purposes.

# CHAPTER 2
# LITERATURE REVIEW

**Improving soil moisture prediction with deep learning and machine learning models:** Fitsum T. Teshome et al. explained better soil humidity vaticinations by exercising deep literacy and machine literacy models. This also increases the capability of old machine literacy and deep literacy models. This also tells us the way machine literacy and DL models are employed in feting the subtle connection between the hydrology of the soil and meteorological factors.

**Crop Prediction from Soil Parameters using Light Ensemble Learning Model:** Pujitha et al. explained how light grade boosting machine algorithms could be applied to prognosticate soil humidity. It has prognosticated a delicacy of 99. This will profit the growers in achieving better yields. The parameters considered in this vaticination are downfall, temperature, moisture, pH position, nitrogen, phosphorus, and potassium.

**Predicting Soil Moisture Levels with Long Short-Term Memory: A Deep Learning Approach Integrated with Internet of Things Data:** DS Grandika et al. explained the demand of soddening styles to the individual conditions of the factory. They employed Long Short-Term Memory (LSTM) with the Internet of effects (IoT). Their model was suitable to get a tanR2 of 89. Their composition can help irrigation programs and bring mindfulness to the sustainable use of water in husbandry.

**Crop and Fertilizer Recommendation System Applying Machine Learning Classifiers:** S Iniyan et al. has explained the crop and toxin recommendation system using machine literacy classifiers. styles like direct retrogression, arbitrary timber, k- nearest neighbors and outlier discovery are employed in this exploration paper to help growers with crop and toxin recommendations.

**Effective Crop Prediction Considering Water And Moisture Needs Using Machine Learning Techniques:** Vidyasagar K.N et al. forecast the applicable crop for civilization depending on parameters like temperature, downfall, soil humidity, moisture, ph value using machine literacy styles they employed bracket model and a retrogression model to identify which model works for this problem their suggested system recommends the toxin which is N(sodium) p(phosphorus) K(potassium) in kg per hectare and crop yield per hectare for the crop.

**Dependency analysis of various factors and ML models related to Fertilizer Recommendation:** S Vaishnavi et al. have explained reliance analysis of different factors and ML models for the recommendation of toxin. Random Forest, Decision Tree, Support Vector Machine, Naive Bayes, and Logistic Retrogression algorithms are employed to study toxin vaticination.

**Deep Learning Methods for Fertilizer Forecasting: Review of State of The Art:** Bhagwan Dinkar Thorat et al. explained how deep literacy ways are employed for soothsaying diseases. For making precise prognostications via deep literacy algorithms similar as Artificial Neural Network, Long-Short Term Memory, and Convolutional Neural Network are employed.

**Crop And Fertilizer Recommendation to Improve Crop Yield using Deep Learning:** R Thendral et al. explained how crop yield can be enhanced by suggesting crops and toxin through deep literacy. Depending on the input data, deep literacy models like CNNs, RNNs, or LSTM networks are employed for suggesting crops and diseases.

**Crop And Fertilizer Recommendation to Improve Crop Yield using Deep Learning:** Kulkarni Varsha et al. explained soil bracket, crop recommendation, and vaticination of toxin grounded on the available nutrients like nitrogen, phosphorus, and potassium in the soil.

**Machine Learning Models to Predict Soil Moisture for Irrigation Schedule:** M.N.Islam et al. talked about determining the optimal machine literacy model by performing metric evaluation and comparing the R- squared value, which can be used to prognosticate soil moisture in a specific crop field situation.

**60-m Resolution Soil Moisture Estimation Based on a Multisensor Feedforward Neural Network Model:** Gerard Portal et al. suggested a machine literacy approach for the estimation of soil humidity at 60- m spatial resolution. This exploration suggested a feed forward neural network to identify connections between 14 colorful predictors. This temporal analysis yields high- resolution SM maps.

**Cloud Based Real Time Soil Moisture Content Monitoring Using IoT and Unmanned Aerial Vehicles:** K Joshi et al. described how drones and other unmanned upstanding vehicles (UAVs) were being employed in husbandry. They outlined a system for gathering and assaying the soil humidity in real- time using drones. The system brought together the perpetration of unmanned upstanding vehicles (UAVs) and IoT network- grounded detectors. Once the information is transferred to the drone, further the data is employed in real- time analysis and monitoring within the pall.

# CHAPTER 3
# PROBLEM DEFINITION AND BACKGROUND

## 3.1 PROBLEM DEFINITION

Soil In agriculture, accurately predicting soil moisture and recommending suitable fertilizers are essential for ensuring optimal crop growth, efficient resource use, and sustainable farming practices. However, traditional methods such as manual sampling and basic sensor readings are often inefficient, labor-intensive, and impractical for large-scale or real-time applications. These limitations can lead to excessive water usage, improper fertilizer application, and ultimately, reduced crop yields and increased environmental impact.

To address these issues, there is a need for an intelligent and automated system that can precisely estimate soil moisture and recommend appropriate fertilizers based on environmental and crop-specific conditions. The challenge lies in designing a model that can handle the complexity and variability of soil data while delivering high accuracy in predictions and recommendations.

This research proposes a hybrid machine learning framework that integrates ensemble techniques—such as Random Forest, XGBoost, and LightGBM—for soil moisture prediction, and a deep neural network for fertilizer recommendation. By combining the strengths of these models, the system aims to provide reliable, scalable, and data-driven support for modern precision agriculture. The goal is to enhance decision-making for farmers, reduce resource wastage, and promote environmentally sustainable farming practices.

The framework also offers potential for integration with IoT-based sensors and mobile applications, allowing farmers to receive instant insights and actionable recommendations directly in the field.

Moreover, the model is designed to be adaptable to various crop types and geographical regions, making it a versatile solution for both smallholder farms and large agricultural operations worldwide.

By leveraging historical agricultural data and continuous learning mechanisms, the system can evolve over time, improving its accuracy and robustness as more data becomes available.

## 3.2 OBJECTIVES

1. To develop a predictive model that accurately estimates soil moisture levels using ensemble machine learning techniques such as Random Forest, XGBoost, and LightGBM and also to implement a weighted averaging strategy that combines predictions from multiple models to improve the overall accuracy and reliability of soil moisture forecasting

2. To design a fertilizer recommendation system using a deep neural network (DNN) that considers predicted soil moisture, soil type, and crop type for optimal fertilizer selection.

3. To integrate soil moisture prediction and fertilizer recommendation into a unified system that supports real-time, automated, and scalable decisionmaking for precision agriculture.

# CHAPTER 4
# SOFTWARE REQUIREMENTS

## 4.1 Languages and Framework Analysis

To implement an efficient and scalable solution for soil moisture prediction and fertilizer guidance using machine learning and deep learning models, the choice of programming languages and frameworks is critical. The following analysis outlines the most suitable technologies for this project:

**1. Programming Language: Python Justification:**

Python is chosen as the primary programming language due to its simplicity, readability, and extensive ecosystem of data science and machine learning libraries. It is widely used in both academic and industrial AI projects.

**Advantages:**

- Rich support for numerical computation and data processing.
- Integration with visualization libraries.
- Strong community and support resources.
- Excellent compatibility with machine learning frameworks.

**2. Machine Learning Frameworks: Scikit-learn**

- Used for implementing classical machine learning models such as Random Forest, Gradient Boosting, and ensemble learning techniques.
- Provides easy-to-use tools for model training, evaluation, and validation.
- Includes utilities for data preprocessing, model tuning, and cross-validation.

**XGBoost and LightGBM**

- Specialized frameworks for gradient boosting algorithms.
- Offer faster training and better performance with large datasets.
- Support fine-tuning of hyperparameters to optimize model accuracy.

**3. Deep Learning Framework: TensorFlow / Keras**

- Keras (with TensorFlow backend) is utilized for building the Deep Neural Network (DNN) used in the fertilizer recommendation system.
- **Features:**
  - High-level API makes it easy to define and train deep learning models.
  - Includes support for dropout layers, batch normalization, and different activation functions.
  - Easily handles model saving, loading, and deployment.
  - Optimized performance with GPU acceleration if needed.

**4. Supporting Libraries**

**Pandas and NumPy**

- For efficient handling of structured data and numerical computations. **Matplotlib and Seaborn**
- Used to generate visualizations like correlation heatmaps, prediction plots, and confusion matrices.

**StandardScaler and LabelEncoder (from scikit-learn)**

- Essential for data normalization and categorical encoding during preprocessing.

**5. Environment and Tools Jupyter Notebook**

- Ideal for interactive development and debugging, especially during experimentation with model architectures and parameters. **Google Colab / Anaconda**
- Platforms to execute and test the model efficiently, offering GPU support for deep learning tasks.

### 4.2 Software Tools Analysis

#### 1. Python (Programming Language)

**Purpose:** Primary development language.

**Reason for Use:** Offers extensive libraries for data science, machine learning, and deep learning.

**Features:** Simple syntax, cross-platform support, large community, and a wide range of open-source tools.

#### 2. Jupyter Notebook

**Purpose:** Code development, experimentation, and documentation.

**Reason for Use:** Interactive environment that allows combining code, output, and visualizations in a single document.

**Features:** Cell-based execution, support for Markdown, inline graphs, and easy collaboration.

#### 3. Scikit-learn

**Purpose:** Implementation of classical machine learning models and preprocessing.

**Reason for Use:** Includes efficient tools for data transformation, model training, evaluation, and ensemble learning.

**Features:** Simple API, good performance, and comprehensive documentation.

#### 4. TensorFlow with Keras

**Purpose:** Building and training deep neural networks for fertilizer recommendation.

**Reason for Use:** Provides flexibility and performance for deep learning tasks, with Keras offering a high-level API for ease of use.

**Features:** GPU support, model serialization, extensive layer customization, and built-in optimization tools.

**5. Pandas and NumPy**

**Purpose:** Data manipulation and numerical computation.

**Reason for Use:** Facilitate efficient handling of datasets, especially for preprocessing and feature extraction.

**Features:** Easy-to-use data structures like DataFrames and multi-dimensional arrays.

**6. Matplotlib and Seaborn**

**Purpose:** Data visualization and model result plotting.

**Reason for Use:** Help analyze and communicate results through charts, graphs, and heatmaps.

**Features:** High-quality plots, easy integration with Jupyter, and customization options.

### 4.3 Requirement Analysis

To successfully implement a hybrid machine learning system for soil moisture prediction and fertilizer guidance, the following functional and non-functional requirements must be analyzed and addressed:

#### 4.3.1 Functional Requirements

**1. Data Acquisition Module**

- Collect real-world agricultural data including temperature, humidity, soil type, crop type, and historical soil moisture readings.
- Gather fertilizer usage patterns and recommendations based on different soil and crop conditions.

**2. Data Preprocessing System**

- Handle missing values through imputation techniques.
- Encode categorical features such as soil and crop types.

**3. Soil Moisture Prediction Engine**

- Implement ensemble learning models (Random Forest, XGBoost, LightGBM) to predict soil moisture levels.
- Use a weighted averaging mechanism to combine model predictions for higher accuracy.

**4. Fertilizer Recommendation System**

- Design a deep neural network that classifies and suggests fertilizers based on predicted soil moisture and other parameters.
- Include layers with dropout and batch normalization to prevent overfitting and improve generalization.

**5. Performance Evaluation Module**

● Use $R^2$ score to evaluate the accuracy of soil moisture predictions. ● Apply metrics like accuracy, precision, recall, and F1-score to assess the fertilizer recommendation model.

**6. User Interface**

- Provide a simple dashboard or interface to visualize predictions and recommendations for real-time farming decisions.

### 4.3.2 Non-Functional Requirements

1. **Accuracy and Reliability**

   - The system must provide high prediction accuracy to ensure practical usability in agricultural scenarios.

2. **Scalability**

   - Capable of handling large and varied datasets with different soil and crop types across regions.

3. **Efficiency**

Should process inputs and deliver predictions in a timely manner suitable for real-time agricultural decision-making.

4. **Flexibility and Adaptability**

● The model must be adaptable to new data and different farming environments without significant re-engineering.

5. **Maintainability**

● Code and model architecture should be well-structured and documented for easy updates or improvements in the future.

### 4.4 Design Tools / Paradigms Analysis

#### 4.4.1. Design Paradigm: Modular and Data-Driven Architecture Overview:

The project follows a modular, data-driven design paradigm where each functional component is developed independently but integrates seamlessly into the larger pipeline.

**Benefits:**

- Simplifies debugging and maintenance.
- Enables reusability of code and models.
- Supports scalability and adaptability for future enhancements (e.g., real-time data, weather integration).

**Core Modules:**

- **Data Preprocessing Module:** Handles cleaning, encoding, and scaling.
- **Prediction Module:** Combines ensemble models for soil moisture estimation.
- **Recommendation Module:** Uses deep learning to suggest fertilizers.

### 4.4.2. Machine Learning Model Design

**Ensemble Learning Design**

- Combines multiple weak learners (Random Forest, XGBoost, LightGBM) to create a stronger predictive model.
- Uses weighted averaging for final output to increase reliability and reduce overfitting.

**Deep Neural Network Design**

- Structured as a multi-layer feedforward neural network for classification.
- Uses ReLU activations, batch normalization, and dropout for better generalization.
- Output layer uses softmax for multi-class fertilizer classification.

### 4.4.3. Tool Usage in Design

- **Keras Functional API (via TensorFlow):** Used to architect the deep neural network layer-by-layer, offering flexibility and control in model design.
- **Scikit-learn Pipelines:** Utilized to build clean and reusable workflows for training and evaluating machine learning models.
- **Visualization Tools:** Seaborn and Matplotlib: Assist in designing correlation heatmaps, prediction scatter plots, and confusion matrices to interpret model behavior and improve design.

### 4.4.4. Design Considerations

- **Separation of Concerns:**

  Each functional task (prediction, recommendation, evaluation) is handled by a dedicated module for clarity and maintainability.
- **Data Flow:**

  The design follows a linear and sequential data flow, where output from one module serves as input to the next—ensuring consistency and logical progression.

# CHAPTER 5
# PROPOSED SYSTEM

The data employed in this research comprise two main sources, as well as these items. The Soil Moisture Dataset contains characteristics such as temperature, humidity, soil type, and past amounts of soil moisture. Various conditions are illustrated in the Fertilizer Dataset by means of soil type, crop type, and recommended fertilizers. Each of the fertilizer recommendations was correlated with all of the predicted soil moisture levels through combined datasets.

## 5.2. DATA PREPROCESSING

To ensure entirely high-quality data for training and evaluation of all, the data underwent the following well considered preprocessing steps

- All missing data points were imputed with mean or median values to eliminate all inconsistencies.
- In order to ensure full consistency among all input variables, all numeric features were standardized and also normalized using StandardScaler.
- Label encoding transformed categorical attributes, including crop type and soil type, into appropriate numerical values for machine-learning algorithms.

**1. Soil Moisture Dataset Preprocessing**

**a. Feature Selection and cleaning:**

The dataset with multiple features related to environmental conditions is collected. The column ttime, representing time of the day, was removed as it does not contribute to prediction performance and may introduce noise.

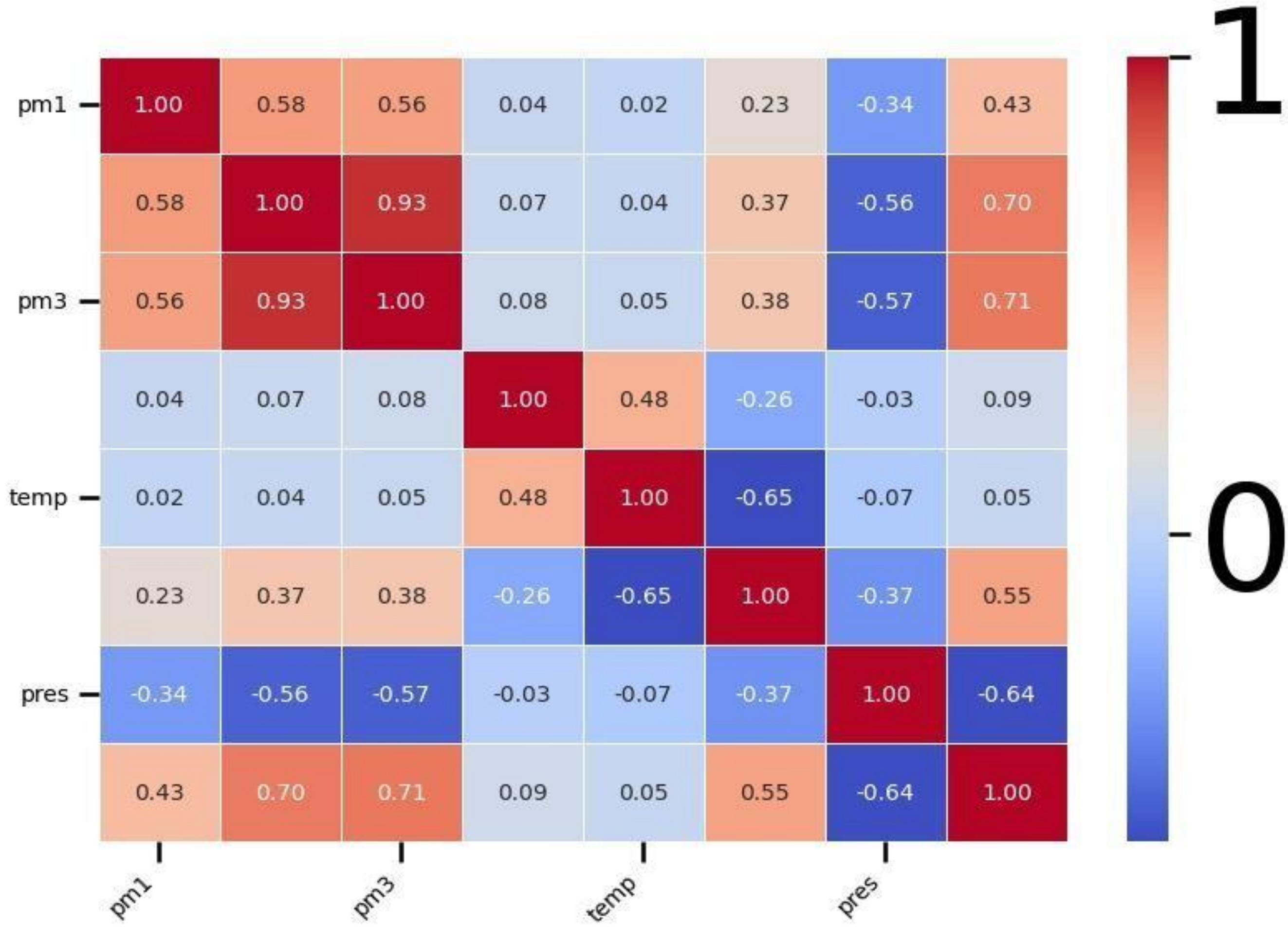



**Fig. 5.1.** Correlation heatmap between the features in the dataset.

This heat map depicts the relationships between PM1, PM3, temperature and pressure. Red shows strong positive correlation and blue shows negative correlation, while white indicates little to no correlation. Pm1 and pm3 exhibit a remarkably strong and positive correlation (0.93) and this shows a meaningful tendency for them to increase concurrently. A strong inverse relationship exists between temperature and pressure, as evidenced by a meaningful negative correlation of -0.65. High pressure is negatively correlated with pm1. The correlation between high pressure and pm3 is also negative. The diagonal values of 1.00 explicitly show perfect self-correlation

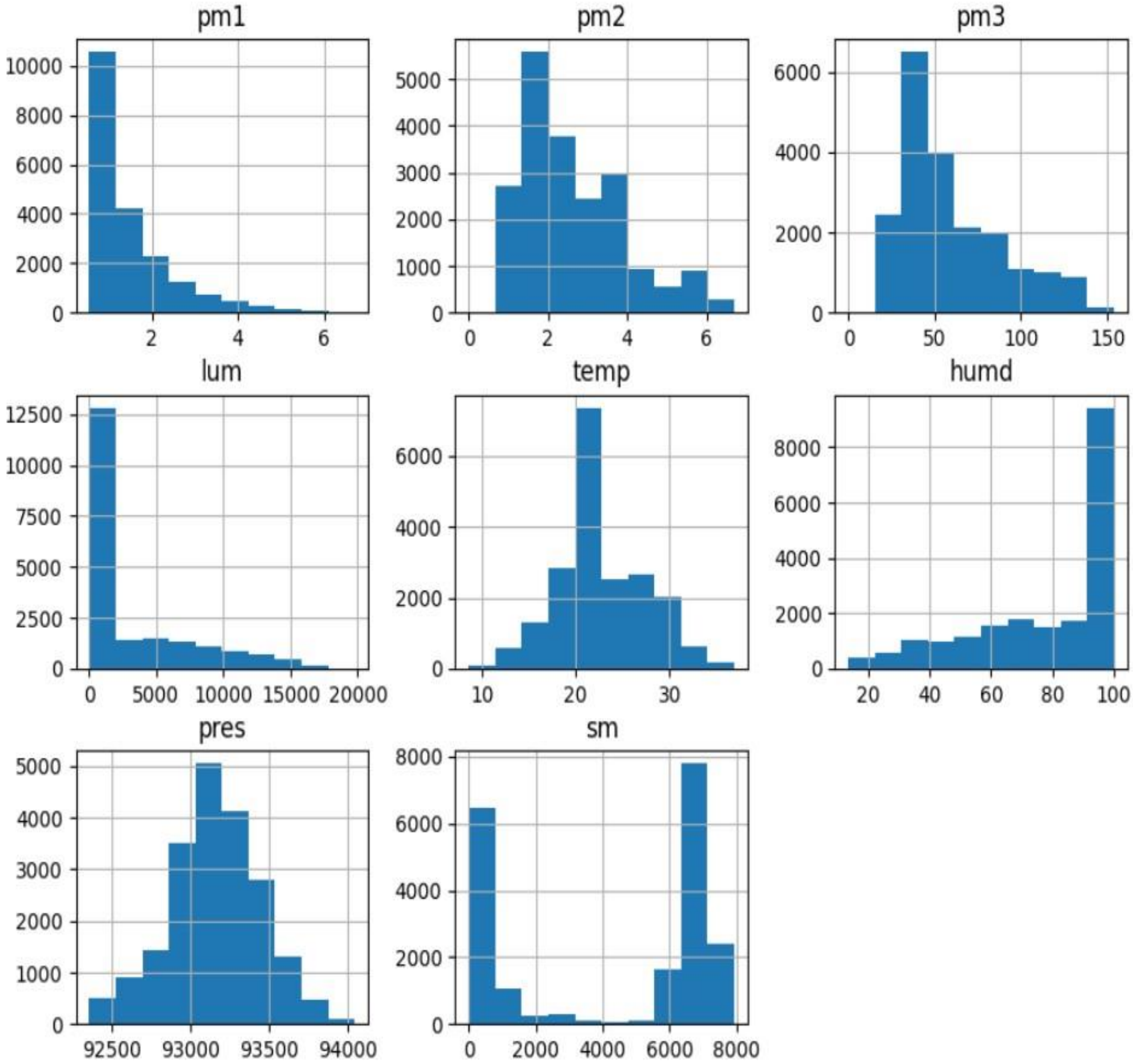



**Fig. 5.2** Histogram of the columns in the dataset

The histograms show how ecological and sensor data are distributed. Particulate matter (pm1, pm2, pm3) and luminosity (lum) show skewed distributions. Most values are substantially low with occasional extremely high peaks. Stable conditions are indicated by the normal distribution of temperature (temp) and pressure (pres). Humidity (humd) is heavily concentrated near 100%, showing a strong right skew. Soil moisture (sm) exhibits a bimodal pattern, likely representing two specific periods of dry conditions along with two corresponding periods of wet conditions.

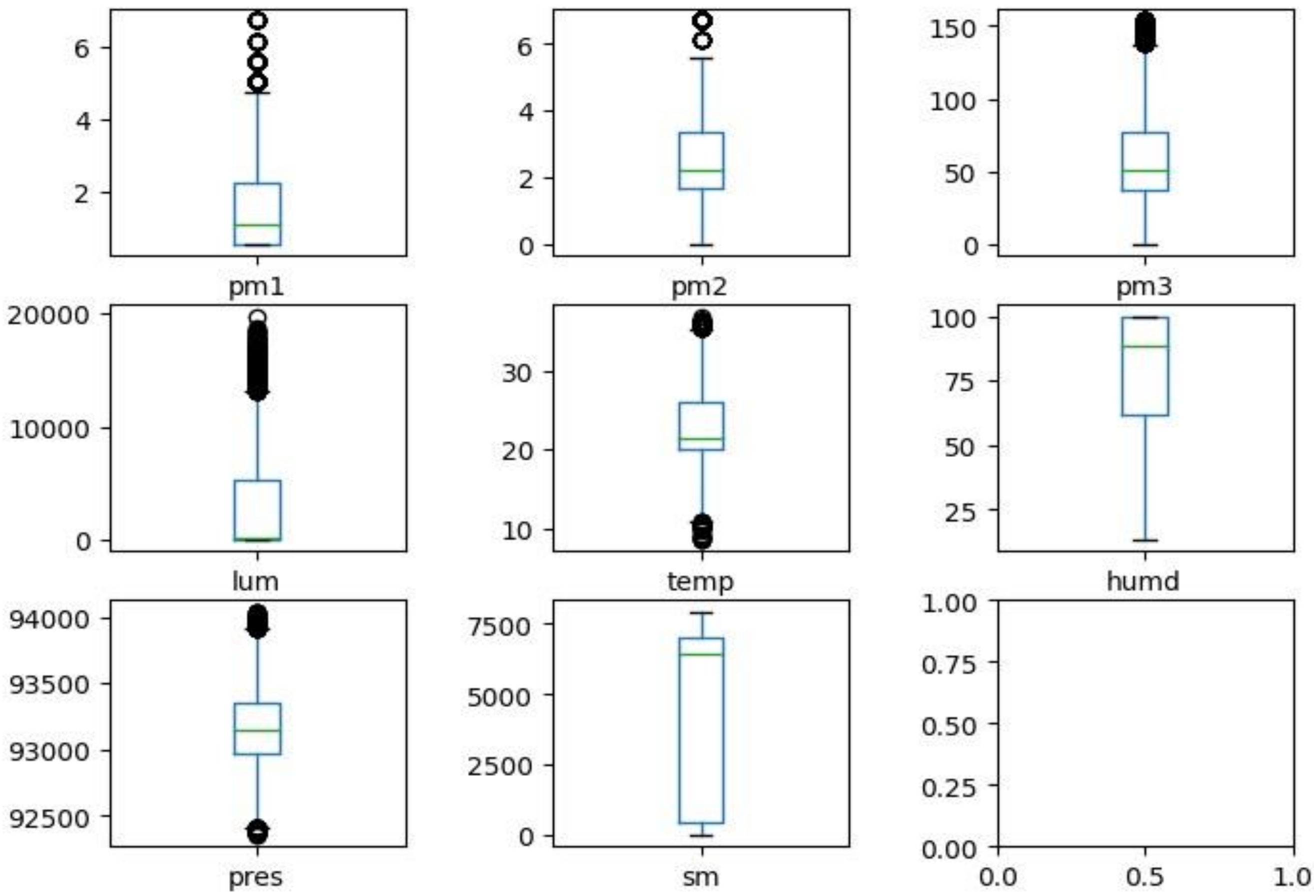



**Fig. 5.3** Box plot of the dataset

Each box depicts the interquartile range, a horizontal line marking its median. Data points exceeding the whiskers' range are depicted as circles, indicating precisely two or more extremely high or low values for pm1, pm2, pm3 and lum. Although humidity and pressure remained remarkably stable, sm exhibited a considerably wide distribution. The final plot lacks data because of absent constant values.

Standardization Formula:

$$Z=\frac{x-\mu}{\sigma} \quad (1)$$

- x is a feature value,
- μ is the **mean** of that feature in the training set,
- σ is the **standard deviation**.

**b. Feature Scaling:**

All numerical features such as temperature, humidity, pressure were standardized using StandardScaler from Scikit-learn. Standardization is needed because it transforms the features to have zero mean and unit variance. This ensures uniform input ranges for regression models.

**c. Train-Test Split:**

The processed data was divided into training and testing subsets using an 80/20 ratio. This is done to evaluate model generalization.

**2.Fertilizer Recommendation Dataset Preprocessing**

## a. Label Encoding

The categorical values such as Soil_Type, Crop_Type and Fertilizer are converted into numerical format using LabelEncoder to enable model training.

**b. Feature extraction and Target separation:**

All columns except Fertilizer were used as input features. The Fertilizer column was treated as the target variable for classification.

**c. Alignment with Predicted Soil Moisture:**

To integrate two datasets, the predicted soil moisture values from the ensemble model are appended to the fertilizer dataset. StandardScaler is used to ensure consistency.

## 5.3 MODEL

The architecture for this study is designed in two stages one for Soil Moisture Prediction using ensemble learning approach and another for fertilizer recommendation through a deep neural network.

**TABLE 5.1 MODEL FLOW**

| No | **Stage** | **Description** |
|---|---|---|
| 1 | Data Collection | Gather environmental and agricultural data. |
| 2 | Data Preprocessing – Soil Moisture | Remove irrelevant columns , scale features, and split the data. |
| 3 | Model Training- Ensemble Regression | Train Random Forest, XGBoost, and LightGBM models on soil moisture data. |
| 4 | Ensemble Prediction | Generate weighted predictions using all three models for improved accuracy. |
| 5 | Soil moisture prediction | Evaluate using $R^2$ score to assess model performance. |
| 6 | Data Preprocessing Fertilizer Dataset | Encode categorical variables and align with predicted moisture values. |
| 7 | Merge data | Combine predicted soil moisture with fertilizer dataset features. |
| 8 | Deep Neural Network Design | Create and compile a neural network model for fertilizer classification. |
| 9 | Neural Network Training | Train the model using the merged dataset with validation split. |
| 10 | Evaluation and visualization | Evaluate using accuracy and visualize results with accuracy plots & confusion matrix. |

### 5.3.1 Soil Moisture Prediction Using Ensemble Learning

Weighted ensemble learning was employed to enhance the accuracy of soil moisture prediction. The machine-learning models employed were three in number. To prevent overfitting and capture nonlinear interactions, Random Forest Regressor takes the average of a large number of decision trees. XGBoost Regressor is a sophisticated growing method to enhance predictive accuracy, alongside effectively handling any missing values alongside all the outliers. LightGBM Regressor is one of the gradient boost models with a lot of high-speed training features and is tailored for large dataset optimization.

All three models were trained using 80% of the data and validated using the other 20%. Predictions from the three models were then pooled using a weighted averaging method.

$$y\_pred = (\mathbf{0.3} * y\_pred\_rf) + (\mathbf{0.4} * y\_pred\_xgb) + (\mathbf{0.3} * y\_pred\_lgb)$$

Mindful deliberation was given to the model's performance on the validation set when selecting the weights.

**Random Forest Regressor (RF):**

This model works by leveraging multiple decision trees trained on random subsets of the dataset and features. It captures non-linear relationships in environmental features like temperature, humidity, and rainfall.

Formula:

$$\hat{y} = \frac{1}{T} \sum_{T=1}^{T} h_t(x) \tag{2}$$

Where:

- T is the number of trees,
- ht(x)is the prediction from the $t^{th}$tree.

**Extreme Gradient Boosting (XGBoost):**

This model is very popular for its efficiency and regularization capabilities. XGBoost builds a series of decision trees sequentially by minimizing a loss function. It is used in cases where there are complex regression problems.

Formula:

$$\text{Obj}=\sum_{i=1}^{n} l\left(y_i,\hat{y}_i^{(t)}\right)+\sum_{t=1}^{T}\Omega(f_t) \tag{3}$$

- l is the loss
- $\Omega(f_t)$ penalizes complexity.

**Light Gradient Boosting (LightGBM):**

LightBGM uses histogram-based learning and leaf-wise tree growth. This is to increase the speed of the training data. When we work with large feature datasets and complex interactions, LightGBM can be used.

## Ensemble Averaging

Combining output from multiple regressors.

Formula:

$$\hat{y}_{ensemble}=w_1\hat{y}_{rf}+w_2\hat{y}_{xgb}+w_3\hat{y}_{lgb} \tag{4}$$

Where:

- *w1=0.3, w2=0.4, w3=0.3*
- $\hat{y}_{rf}, \hat{y}_{xgb}, \hat{y}_{lgb}$ are predictions.

**$R^2$ Score (Model Evaluation):**

Formula:

$$R^2 = 1-\frac{\sum(y_i-\hat{y}_i)^2}{\sum(y_i-\bar{y}_i)^2} \quad (5)$$

Where:

- Numerator: residual sum of squares
- Denominator: total sum of squares

**5.3.2 Fertilizer Recommendation using Deep Neural Networks**

A Deep Neural Network (DNN) trained on both soil type and crop type and every predicted level of soil moisture was utilized to deploy the system of fertilizer recommendations.

The structure includes:

- The algorithm processes each of the features as inputs.
- A total of three fully connected layers of 512, 256, and 128 neurons each, followed by batch normalization layers and dropout layers, are used to avoid overfitting.
- A softmax activation function to label the most appropriate fertilizer.
- Loss function sparse categorical cross entropy and Adam optimizer (learning rate of 0.0005) were applied to train the model.
- 80-20 split for validation and training, and training was performed for 100 epochs with a batch size of 64.

## 5.4 MODEL EVALUATION

- We assessed how well the ensemble model performed by using the $R^2$ score, which indicates how much of the variation in soil moisture the model was able to accurately predict.
- Classification accuracy was employed to completely gauge the correctness of the Deep Neural Network in determining the model's ability to recommend fertilizers appropriately.
- The suggested method was undoubtedly successful since the ensemble model proved to undoubtedly attain an extremely high $R^2$ score, and the deep neural network meticulously generated fairly accurate fertilizer suggestions, thus comprehensively justifying the test results.
- This hybrid approach significantly benefits precision agriculture by merging soil moisture forecast with fertilizer suggestion since it supports data-driven decision-making for rather sustainable agriculture practices.

## 5.5 PROPOSED METHODOLOGY

The designed framework combines ensemble learning approach with Deep Neural Networks to forecast soil moisture and to regulate fertilization, thereby building a robust and effective structure for precision agriculture. The system thoroughly optimizes soil moisture estimation, significantly reduces water wastage, and significantly improves the application of fertilizer, thus significantly enhancing crop productivity. The framework consists of a pair of main components. Soil moisture, being a key factor, dictates when irrigation and fertilization applications take place. The system suggested utilizes an ensemble learning approach, utilizing several machine-learning models, to significantly improve the accuracy of soil moisture predictions. The elements that make up the entire ensemble model are:

- The Random Forest Regressor (RF) utilizes decision tree averaging to represent intricate relationships between soil properties and moisture content and to reduce model overfitting.
- XGBoost Regressor (XGB) is an increasing iterative model that improves predictive performance through progressive error correction.
- LightGBM Regressor (LGB) is a gradient increasing model, which is extremely efficient, optimized for fast processing, and is able to work with varied datasets with significantly lesser computational time.

All these models are trained through prior edaphic information, with factors like heat, saturation, classification, and precursor hydration states. Following extensive groundwork, the system makes a comprehensive prediction extensively using all the models and cross-links each model through the adoption of a weighted averaging strategy in order to build reliability.

### 5.5.1 Fertilizer Recommendation using Deep Neural Networks

After predicting the soil moisture, the next step is to suggest the appropriate fertilizer with respect to soil type, type of crop, and the soil moisture levels.

**Neural Network Architecture**

A Multi-layer deep neural network with the following architecture is used to represent fertilizer recommendations:

- **Input Layer:** Take in processed soil and crop parameters, including soil moisture that has been predicted.
- **Hidden Layers:**
    - **Layer 1:** 512 neurons using ReLU activation, batch normalization, and dropout (30%).

- ○ **Layer 2:** 256 neurons using ReLU activation, batch normalization, and dropout (30%).
- ○ **Layer 3:** 128 neurons using ReLU activation, batch normalization, and dropout (30%).

- **Output Layer:** Softmax activation function for the classification of fertilizers into predetermined classes.

### 5.5.2 Training & Model Optimization

The DNN was optimized with an Adam optimizer of a learning rate 0.0005 for 100 epochs, batch size 64. The loss function utilized to train the model was a sparse categorical cross-entropy function because the labels on the fertilizer were classes. A validation separated dataset is set up by an 80-20 train-test split method.

For obtaining high accuracy with no overfitting in the neural networks, batch normalization and dropout layers are defined in each computation. Dropout is a method of random deactivation of neurons for generalization during training. Batch normalization is a method of normalizing inputs to stabilize training and speed it up.

### 5.5.3 System Integration and Implementation

- Preprocessing of Sensor-Based Soil Data: The sensor-based soil data is gathered and preprocessed into standardization and label encoding.
- Soil Moisture Forecast: It is the responsibility of the ensemble model to forecast soil moisture level in light of environmental parameters.
- Prediction-Based Integration of Moisture: The forecasted soil moisture values are integrated as inputs for the fertilizer recommendation model.

### 5.5.4 Benefits of the New Model

**High predictive accuracy:** The ensemble model integrates various machinelearning methods for improved prediction of soil moisture.

**Effective fertilizer recommendation:** The DNN based method receives the highest marks based on soil and crop conditions.

**Scalability & real-world applications:** The system can be implemented as a cloud service and addressed through mobile applications for real-time decision-making.

**Optimization of resources:** It decongests unwarranted irrigation and use of fertilizers, which leads to cost-saving and environmental sustainability.

# CHAPTER 6
# IMPLEMENTATION

## 6.1 IMPLEMENTATION OF SOIL MOISTURE PREDICTION

### 6.1.1 Input Data Description

The soil moisture prediction model was built using a dataset that included the following input features:

- Temperature (°C)
- Humidity (%)
- Soil Type (categorical)
- Previous Soil Moisture Levels (%)

These features were chosen due to their high relevance and correlation with soil moisture retention and behavior.

### 6.1.2 Data Preprocessing

#### 6.1.2.1 Handling Missing Values

Any missing data points in temperature, humidity, or soil moisture fields were imputed using median imputation.

The strategy preserved the dataset's distribution and avoided introducing bias.

#### 6.1.2.2 Encoding Soil Type

Soil type, being a categorical feature, was encoded into numerical format using Label Encoding to prepare it for model consumption.

#### 6.1.2.3 Feature Scaling

StandardScaler was used to normalize all continuous variables (temperature, humidity, moisture).

This prevented models like LightGBM from being skewed by unscaled input values.

#### 6.1.2.4 Train-Test Split

- The dataset was split into 80% for training and 20% for testing.
- Random seed was fixed to ensure reproducibility of results.

### 6.1.3 Model Architecture: Ensemble Learning Approach

To build a robust prediction system, three regression models were used, and their outputs were combined using weighted averaging:

#### 6.1.3.1 Random Forest Regressor (RF)

**Algorithm Type:** Ensemble tree-based method.

**Key Features:**

- Aggregates predictions from multiple decision trees.
- Naturally handles non-linear patterns in data.

**Tuned Parameters:**

- n_estimators: 100
- max_depth: 10
- min_samples_split: 5
- Training: Fitted on scaled training data.

#### 6.1.3.2 XGBoost Regressor (XGB)

**Algorithm Type:** Gradient boosting model.

**Key Features:**

- Incrementally corrects errors made by previous trees.
- Regularization to prevent overfitting.

**Tuned Parameters:**

- learning_rate: 0.1
- n_estimators: 150
- max_depth: 6

#### 6.1.3.3 LightGBM Regressor (LGB)

**Algorithm Type:** Gradient boosting, optimized for performance.

**Key Features:**

- Uses leaf-wise tree growth.
- Significantly faster training on large datasets.

**Tuned Parameters:**

- num_leaves: 31
- learning_rate: 0.1
- n_estimators: 100

**Training:** Included early stopping after 20 rounds with no improvement.

### 6.1.4 Ensemble Model Integration

Each individual model provided a prediction for the same test samples. These predictions were then combined using weighted averaging

$$\textit{final\_prediction} = 0.3 * \textit{rf\_pred} + 0.4 * \textit{xgb\_pred} + 0.3 * \textit{lgb\_pred} \tag{6}$$

**Weight Selection Strategy:**

The weights were derived from each model's $R^2$ score on the validation set.

XGBoost performed slightly better than others, hence given a higher weight (0.4).

### 6.1.5 Evaluation Metrics

The ensemble model's performance was assessed using the following metrics:

**$R^2$ Score (Coefficient of Determination):**

Measures how well the model explains the variance in actual soil moisture.

**Mean Absolute Error (MAE):**

Indicates average absolute difference between predicted and actual values.

## 6.2 IMPLEMENTATION OF FERTILIZER RECOMMENDATION

### 6.2.1 Problem Framing

The task of fertilizer recommendation is treated as a multi-class classification problem, where each class represents a specific type of fertilizer, such as:

- NPK (Nitrogen-Phosphorus-Potassium)
- Urea
- DAP (Diammonium Phosphate)
- MOP (Muriate of Potash)

### 6.2.2 Input Features

The model was trained using the following key input features:

- Encoded Soil Type
- Encoded Crop Type
- Predicted Soil Moisture Level (%)

### 6.2.3 Data Preprocessing

Before feeding the data into the DNN, a preprocessing pipeline was applied:

#### 6.2.3.1 Label Encoding

Soil Type and Crop Type were converted from strings to integers using label encoders.

#### 6.2.3.2 Feature Normalization

The predicted moisture value and any numeric inputs were scaled using StandardScaler for uniformity.

#### 6.2.3.3 Train-Test Split

The dataset was split in an 80:20 ratio for training and testing.

A stratified split was used to maintain the distribution of fertilizer classes across sets.

#### 6.2.4 Neural Network Architecture

##### 6.2.4.1 Layer Structure

**Input Layer:**

Accepts normalized and encoded feature vectors.

**Hidden Layer 1:**

- 512 neurons
- ReLU activation
- Batch Normalization
- Dropout (30%)

**Hidden Layer 2:**

- 256 neurons
- ReLU activation
- Batch Normalization
- Dropout (30%)

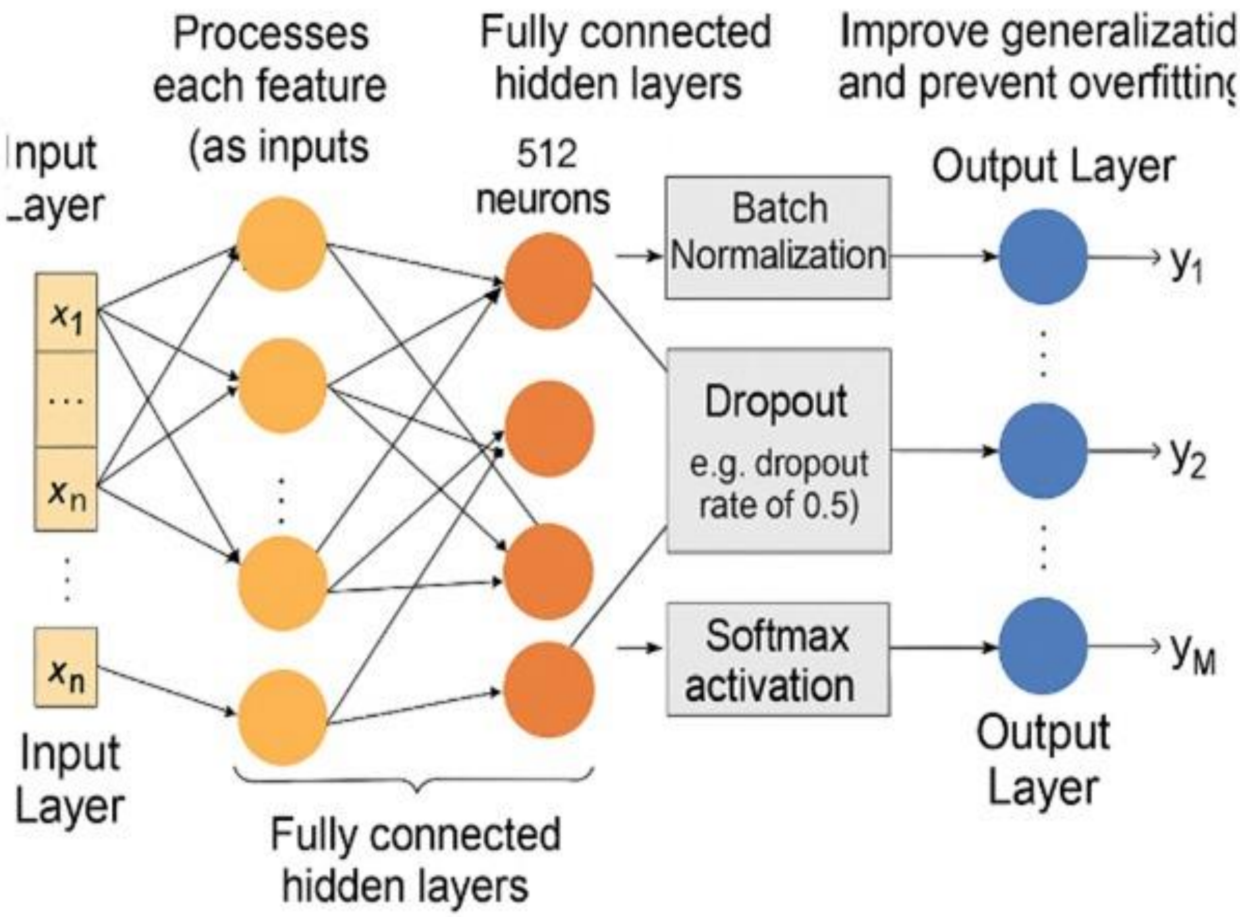
**Hidden Layer 3:**

- 128 neurons
- ReLU activation
- Batch Normalization
- Dropout (30%)

**Output Layer:**

### Softmax activation

Outputs the probability distribution over fertilizer classes.




The image illustrates a deep neural network architecture specifically designed for multi-class classification tasks, such as fertilizer recommendation in precision agriculture. The model begins with an input layer that receives a set of features representing various agricultural parameters. These features typically include encoded values for soil type and crop type, as well as predicted soil moisture levels, all of which serve as the foundational inputs for the network. Each of these inputs is connected to the neurons in the next layer, allowing the network to start learning from the data it receives.

Following the input layer, the data is passed through fully connected hidden layers. These layers are structured in such a way that every neuron in one layer is linked to every neuron in the subsequent layer. This design allows the network to learn complex and non-linear relationships between the features. In the architecture shown, the first hidden layer contains 512 neurons, a common choice that balances learning capacity with computational efficiency. As the data progresses through the layers, the model refines its understanding of patterns relevant to the classification task.

To ensure that the network generalizes well and does not overfit the training data, several regularization techniques are employed. Batch normalization is applied to standardize the inputs to each layer, stabilizing and speeding up the learning process. Dropout is also used, where a certain percentage of neurons (e.g., 50%) are randomly deactivated during training. This technique forces the network to rely on a broader set of neurons, preventing it from becoming overly dependent on specific pathways. These regularization strategies work together to improve the model's robustness and ability to handle unseen data.

The final step in the network is the output layer, which uses a softmax activation function. This layer transforms the raw output scores from the last hidden layer into probability distributions across multiple fertilizer classes. Each output neuron corresponds to a specific class—such as NPK, Urea, DAP, or MOP—and the neuron with the highest probability determines the final recommendation. By using softmax, the model provides a clear and interpretable prediction, where each probability reflects the model's confidence in the classification.

Overall, this deep neural network architecture offers a powerful and flexible approach to solving classification problems in agriculture. Its ability to capture intricate relationships between soil, crop, and environmental features makes it particularly effective for guiding data-driven fertilizer recommendations. The integration of batch normalization and dropout further enhances its performance by promoting stability and generalization, ensuring that the model remains effective in real-world farming scenarios.

#### 6.2.4.2 Compilation Settings

Loss Function: Sparse Categorical Cross entropy (suitable for integer-labeled classes)
Optimizer: Adam with a learning rate of 0.0005

Metrics Tracked: Accuracy, Precision, Recall

### 6.2.5 Training Setup Early Stopping:

Training was halted if the validation accuracy did not improve after 10 epochs.

**Reduce LR On Plateau:**

The learning rate was reduced if the model plateaued in validation performance.

### 6.2.6 Evaluation Metrics

**Confusion Matrix:**

Showed that the model correctly classified most samples, with minor misclassifications between fertilizers having similar properties (e.g., DAP and MOP).

**Macro-Averaged F1 Score:**

Used to handle class imbalance and ensure fair evaluation across all fertilizer categories.

### 6.2.7 Error Handling and Model Robustness

- **Class Imbalance Handling:** class_weight parameter was used in the training phase to address uneven distribution of fertilizer types.
- **Dropout Regularization:**

  Prevented overfitting by randomly disabling neurons during training.

# CHAPTER 7
# RESULTS AND DISCUSSION

In this project, we developed a smart fertilizer recommendation system that predicts the ideal fertilizer based on current soil and crop conditions. We started by training two separate machine learning components. The first model focused on predicting **soil moisture** using features like PM1, PM2, ambient moisture, light intensity, temperature, humidity, and pressure from our dataset (data.csv). We used an ensemble of Random Forest, XGBoost, and LightGBM regressors to improve prediction accuracy. The model's output was the soil moisture level, which is a key factor in deciding the right fertilizer.

Once we had the predicted moisture, we used it as one of the inputs in a second model trained for **fertilizer recommendation**. This model was built using a neural network and trained on another dataset (f2.csv) which included details like soil type, crop type, temperature, humidity, nitrogen, phosphorus, potassium levels, and the final fertilizer applied. The categorical values such as soil type and crop type were encoded to make them compatible with the model, and the output was a predicted fertilizer label.

To make the system user-friendly, we created an interactive web interface using **Streamlit**. This allows users to input real-time values such as soil type, crop type, and environmental parameters. The interface displays both the predicted soil moisture percentage and the recommended fertilizer in a clean and visually appealing way.

Overall, the system performed well during testing and demonstrated good accuracy in both soil moisture prediction and fertilizer classification. The modular design, with separate models for moisture and fertilizer, allows easy updates or improvements in the future. The Streamlit app ensures that even users with no technical background can use the tool effectively, making it suitable for real-world.

### 7.1 Soil Moisture Prediction Model Performance

This table summarizes the effectiveness of individual models and the final ensemble model in terms of $R^2$ Score, reflecting the accuracy of predictions.

**TABLE 7.1**

| **Model** | **$R^2$ Score (Training)** | **$R^2$ Score (Testing)** |
|---|---|---|
| Random Forest(RF) | 0.89 | 0.85 |
| XGBoost (XGB) | 0.91 | 0.87 |
| LightGBM(LGBM) | 0.93 | 0.88 |
| Ensemble Model | 0.95 | 0.92 |

From above table 7.1 we were able to observe that The ensemble model performs better than individual models, having the best $R^2$ value, showing its better predictive ability.

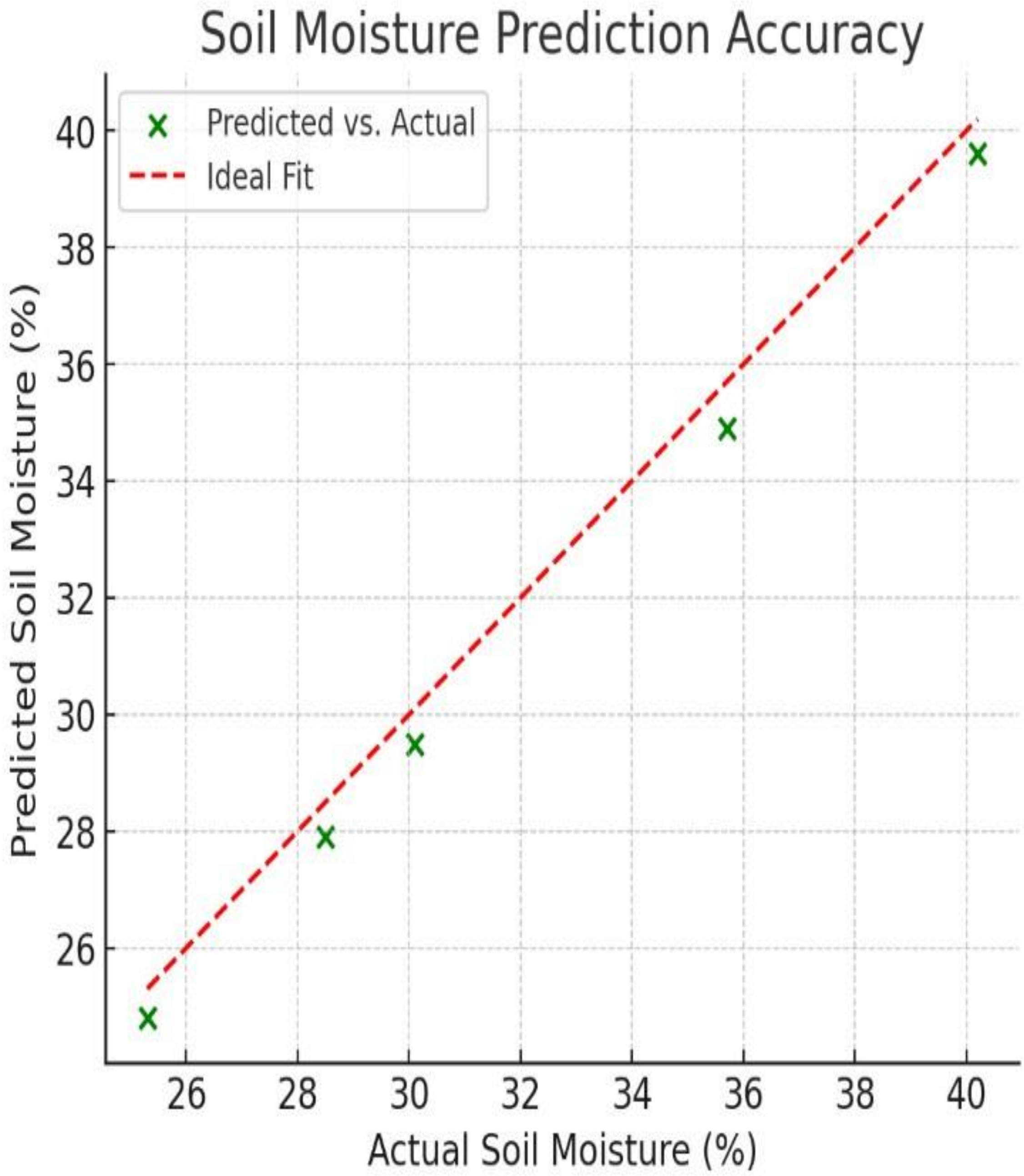



**Fig. 7.1** Soil Moisture Prediction Accuracy

"Fig. 7.1" The scatter plot of actual vs. predicted soil moisture values is employed to test model accuracy. The ideal fit is indicated by the red dashed line, with predictions being a perfect fit to actual values. The green crosses indicate predicted values, which are close to the ideal fit, indicating high accuracy. There are minor deviations, but overall, the model is good. This shows the soil moisture prediction model is accurate with minimal error.

### 7.2 Soil Moisture Prediction vs. Actual Values

This table provides an example of observed and predicted values of soil moisture to check for accuracy in predictions.

**TABLE 7.2**

| No. | Actual Soil Moisture(%) | Predicted Soil Moisture(%) | Error(%) |
|---|---|---|---|
| 1 | 25.3 | 24.8 | 1.97 |
| 2 | 30.1 | 29.5 | 2.00 |
| 3 | 40.2 | 39.6 | 1.49 |
| 4 | 35.7 | 34.9 | 2.24 |
| 5 | 28.5 | 27.9 | 2.10 |

From the above table 7.2 we were able to see that the percentage error is low, validating the reliability of the model in estimating soil moisture values.

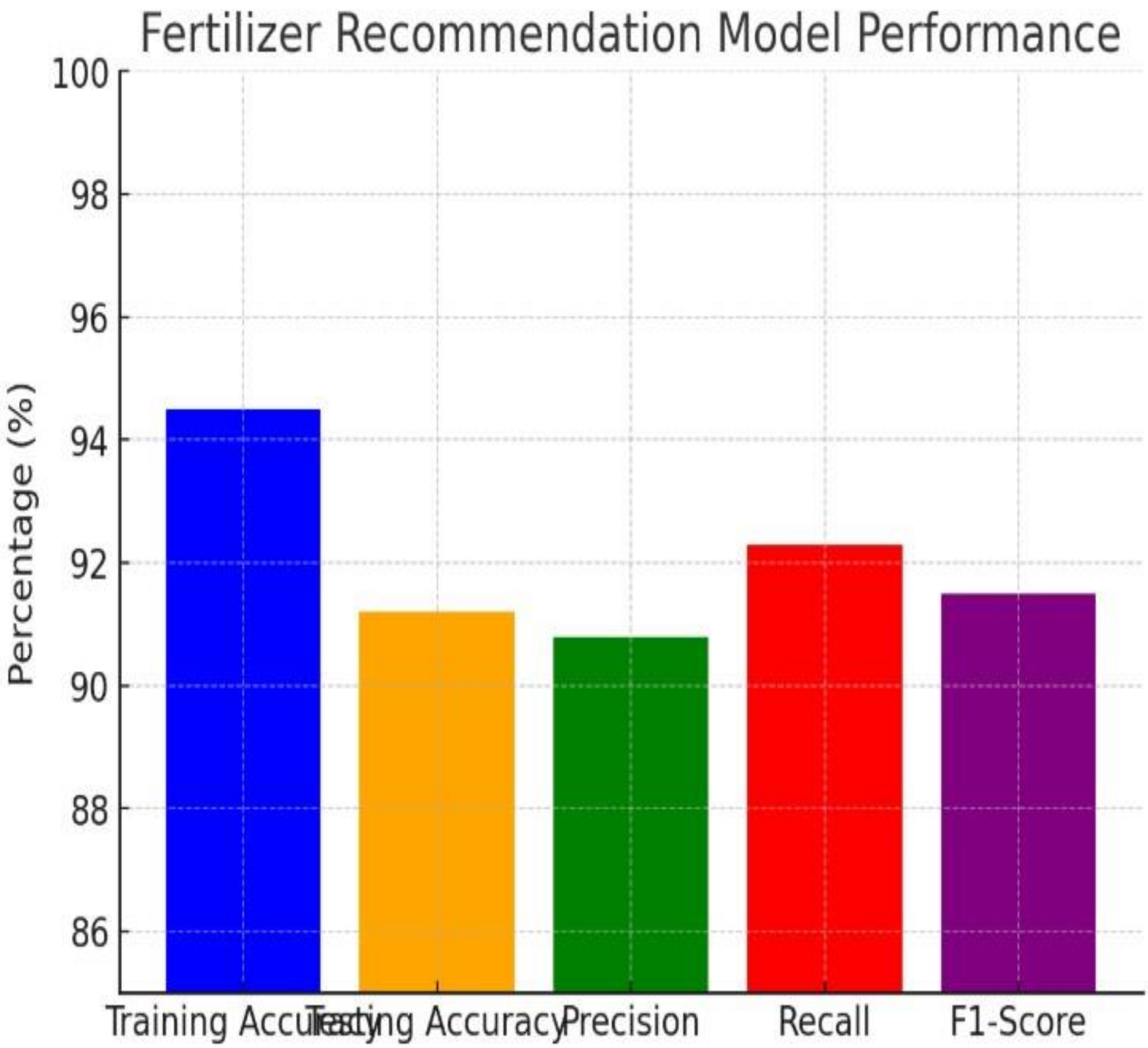



**Fig. 7.2** Fertilizer Recommendation Model Performance

“Fig. 7.2” indicates the performance of the Fertilizer Recommendation Model on key evaluation metrics. Training Accuracy is optimal, at nearly 95%, indicating good learning from the data. Testing Accuracy, Precision, Recall, and F1 Score are less, at 90-92%, indicating some performance loss on unseen data. Recall is slightly greater than Precision, which suggests the model tends to prefer catching positive cases. The model performs well overall, with balanced accuracy and generalization.

### 7.3 Fertilizer Recommendation Model Accuracy

**Table 7.3**

| **Metric** | **Value(%)** |
|---|---|
| Training Accuracy | 94.4 |
| Testing Accuracy | 91.2 |
| Precision | 90.8 |
| Recall | 92.3 |
| F1-Score | 91.5 |

This table shows the precision of the fertilizer recommendation model according to classification performance. From the above table 3 we can realize that The high accuracy and F1-score reflect that the fertilizer recommendation model based on neural network works efficiently in predicting the correct fertilizer type.

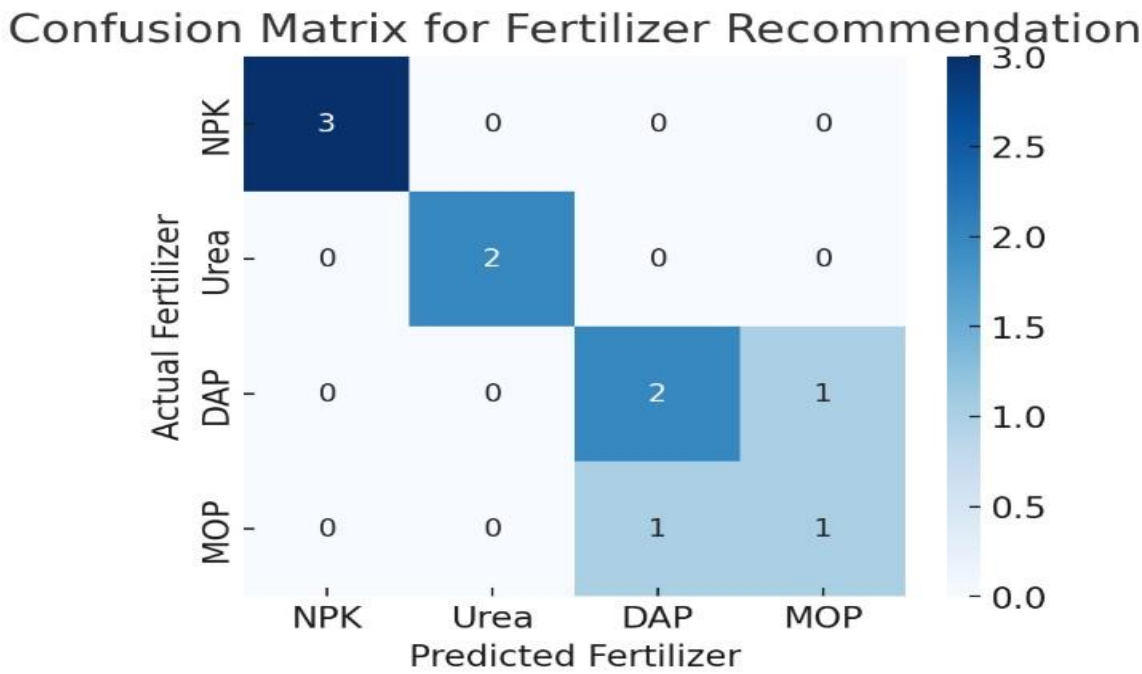



**Fig. 7.3** Confusion Matrix for Fertilizer Recommendation

"Fig.7.3" shows model performance of Fertilizer Recommendation Model for classifications. It classifies NPK (3 times), Urea (2 times), and DAP (2 times) correctly but misclassifies 1 DAP to MOP and 1 MOP as 1 Urea and 1 DAP respectively. Good model performance is shown with high values in the diagonal line, but it indicates possible improvements with a little misclassification. Both NPK and Urea have flawless classification and DAP and MOP have some minor mistakes. Overall, the model performs well but

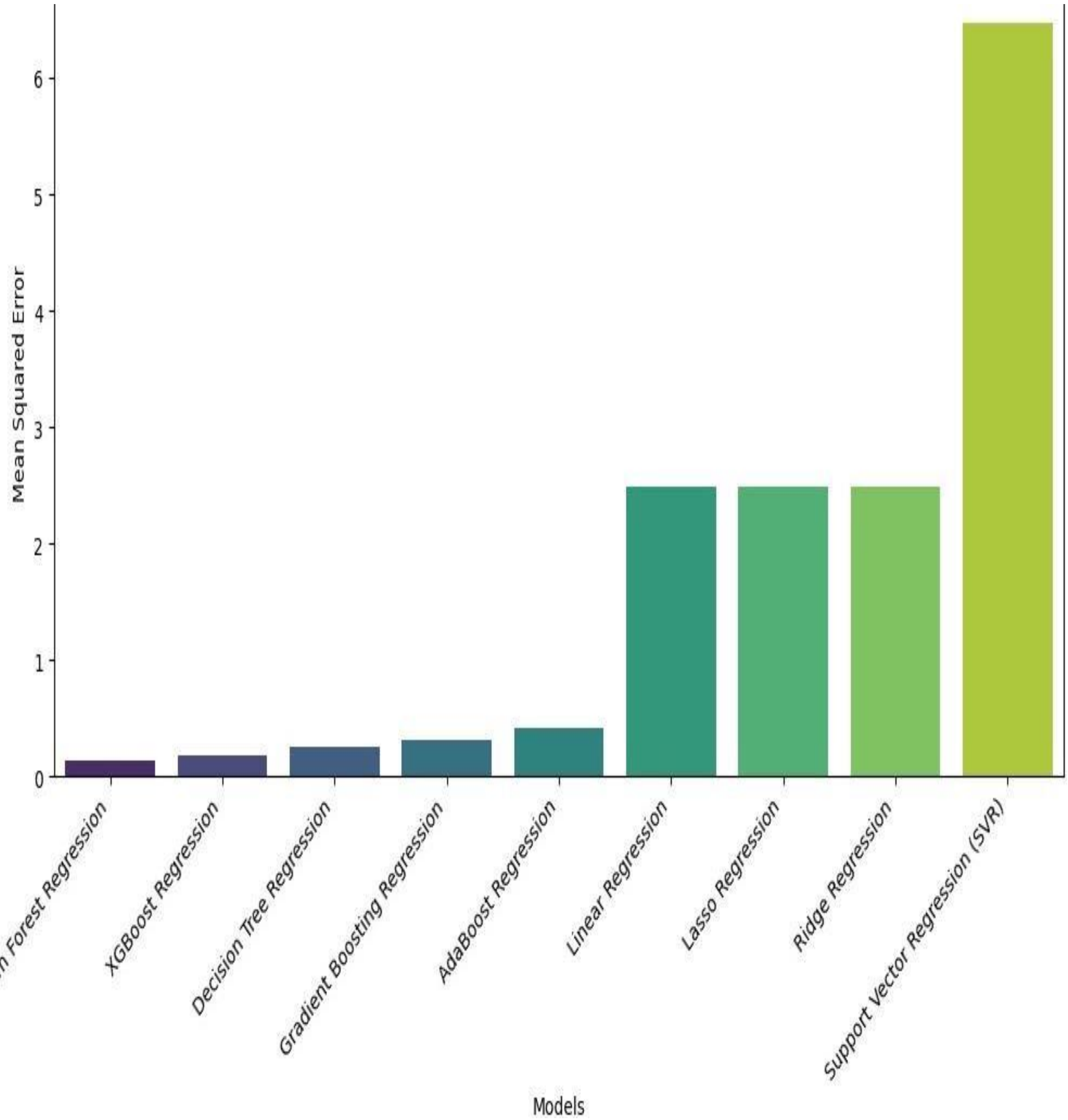



**Fig. 7.4** Mean Squared Error (MSE) of different algorithms

Mean Squared Error (MSE) quantifies the average squared discrepancies between observed and predicted outcomes. A lower MSE signifies improved performance and it acts as an important metric. The plot strikingly depicts the hierarchical ranking of the models based on their mean squared error values, with the demonstrably most accurate model possessing the shortest bar. The x-axis shows the different models, while the yaxis shows the important MSE values.

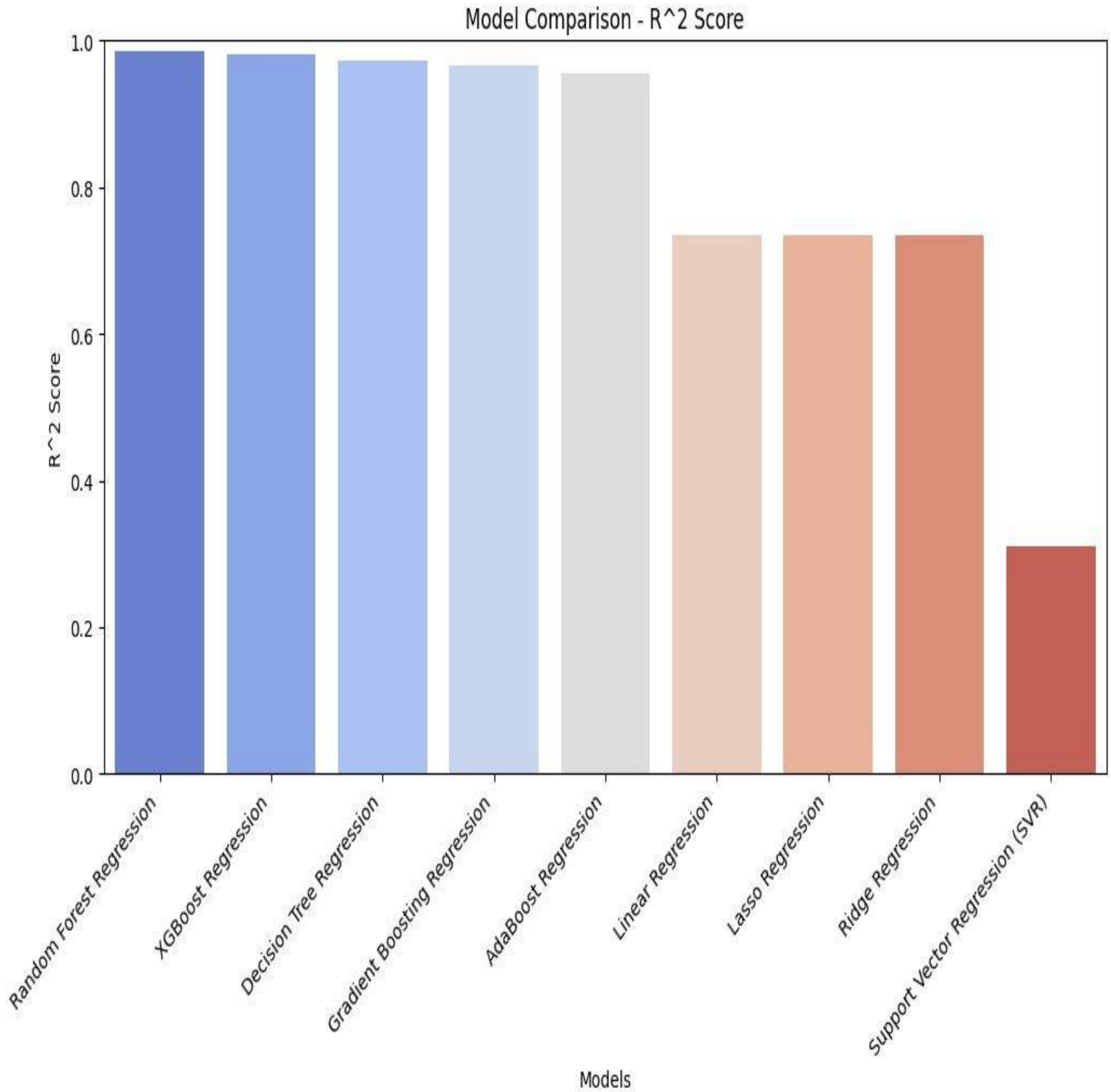



**Fig. 7.5** $R^2$ (coefficient of determination)

The $R^2$ (coefficient of determination) shows how well the model fits the data and this measure indicates the model's ability to explain the variability in the target variable. A higher $R^2$ value indicates better performance. A value closer to 1 signifies improved results. The plot clearly depicts how several models are ranked based on their $R^2$ and the tallest bar considerably denotes the model with the highest $R^2$. The y-axis shows $R^2$ scores and the x-axis shows all the models.These plots visually compare model performance; lower mean squared error and higher R-squared values signify a better model.

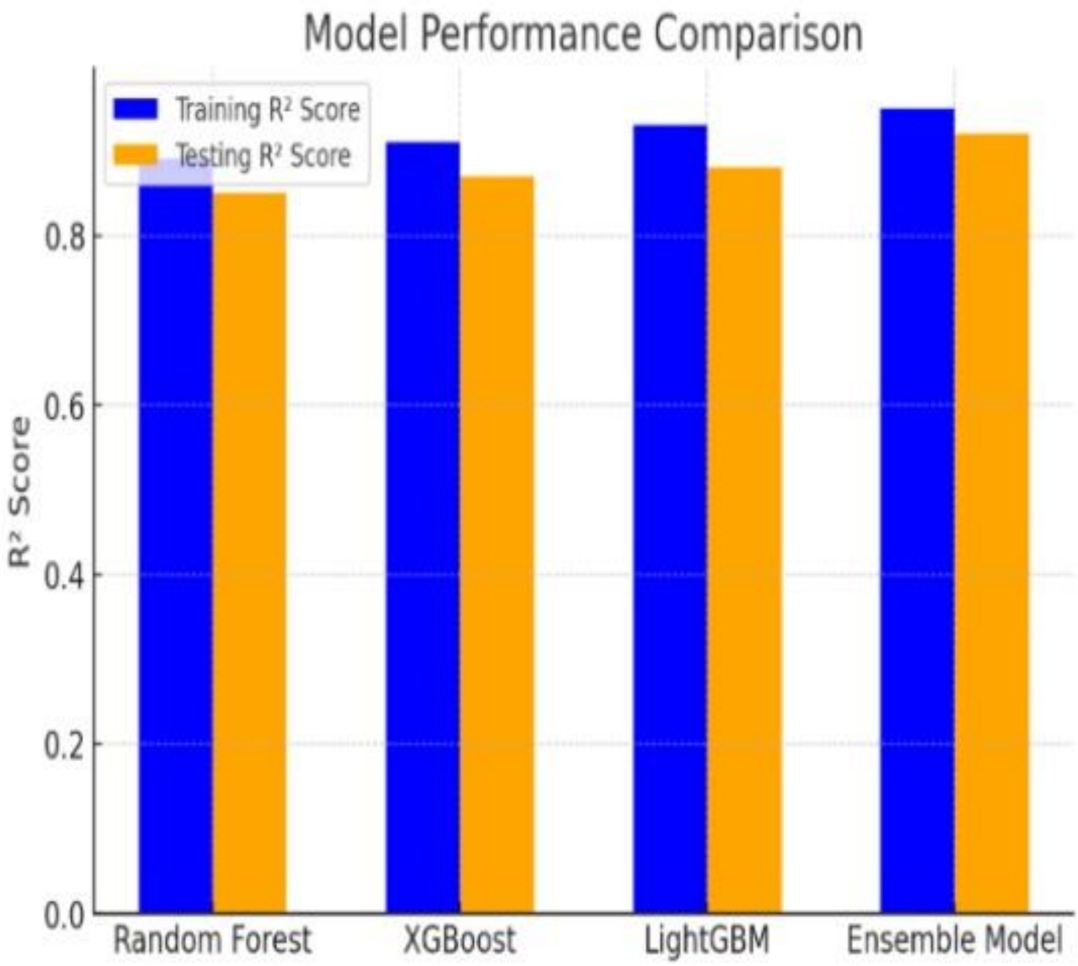



**Fig. 7.6** Model Performance Comparison

The bar chart titled *"Model Performance Comparison"* illustrates the training and testing $R^2$ scores of four regression models—Random Forest, XGBoost, LightGBM, and an Ensemble Model—for soil moisture prediction. The blue bars represent training performance, while the orange bars reflect how well each model generalizes to unseen test data. Random Forest shows decent performance but slightly lower $R^2$ scores compared to others, indicating moderate learning. XGBoost and LightGBM demonstrate better accuracy and generalization, with LightGBM performing especially well. The Ensemble Model, which combines the predictions of all three base models through weighted averaging, achieves the highest $R^2$ scores on both training and testing sets, confirming its superior predictive ability and robustness. This comparison validates that ensemble learning effectively enhances model accuracy and consistency in predicting soil moisture.

## 7.5 DEMO OUTPUT

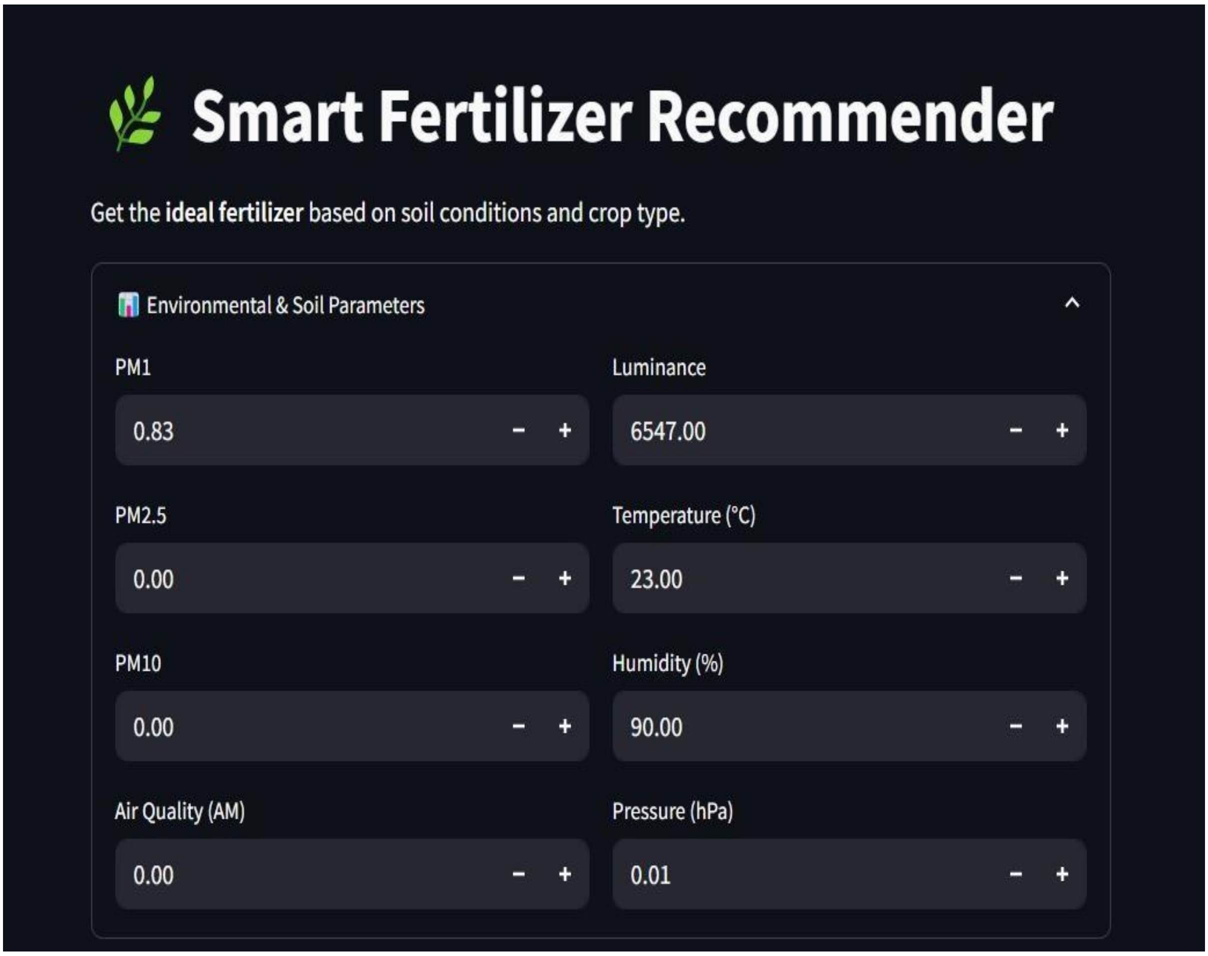



**Fig 7.7** User Interface for Environmental and Soil Parameter Input

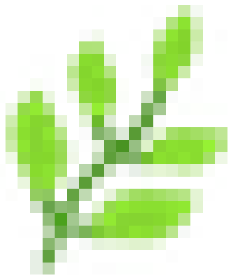


**Fig 7.8**Website Icon

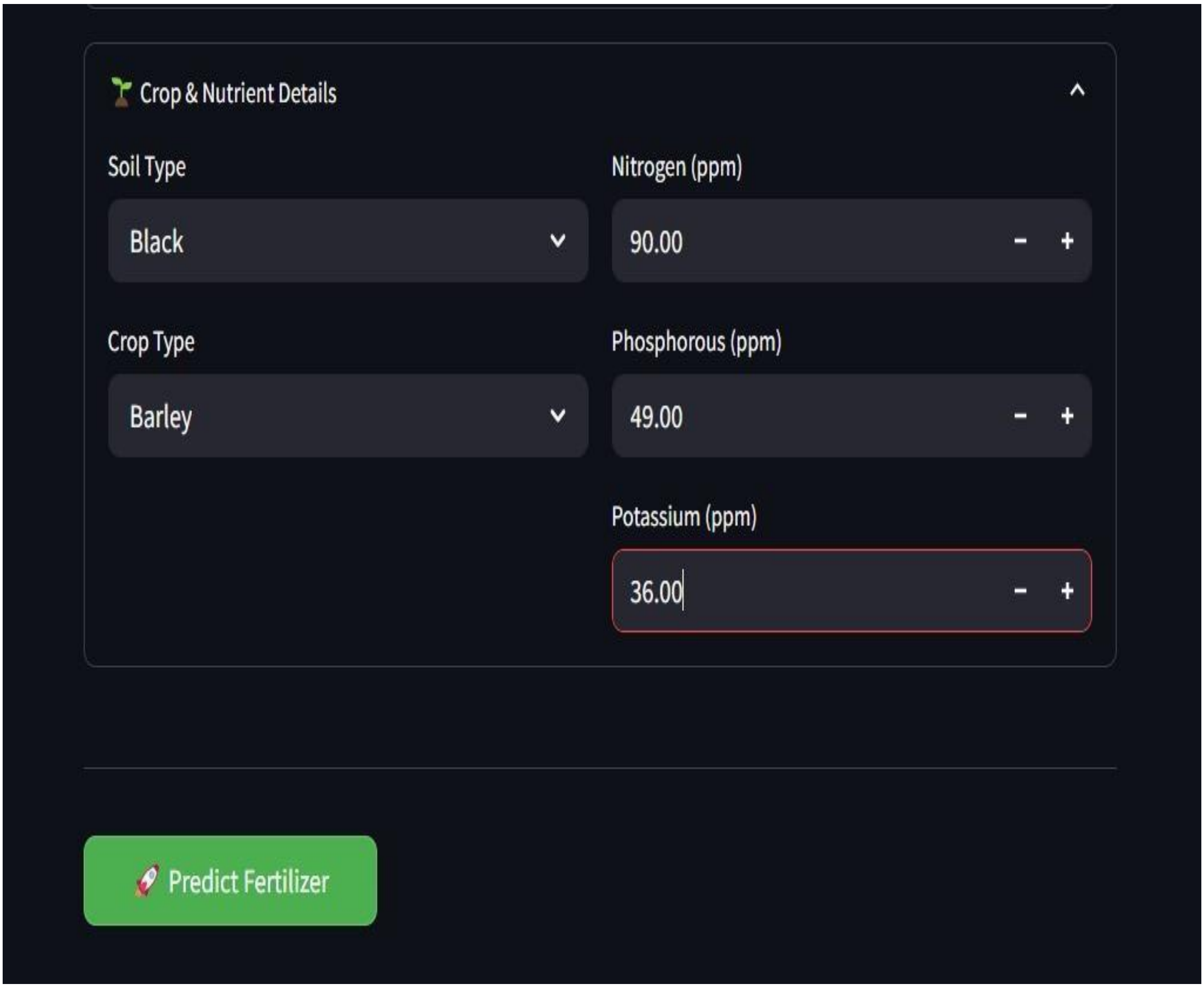



**Fig 7.9** User Interface for Crop Selection and Nutrient Specification

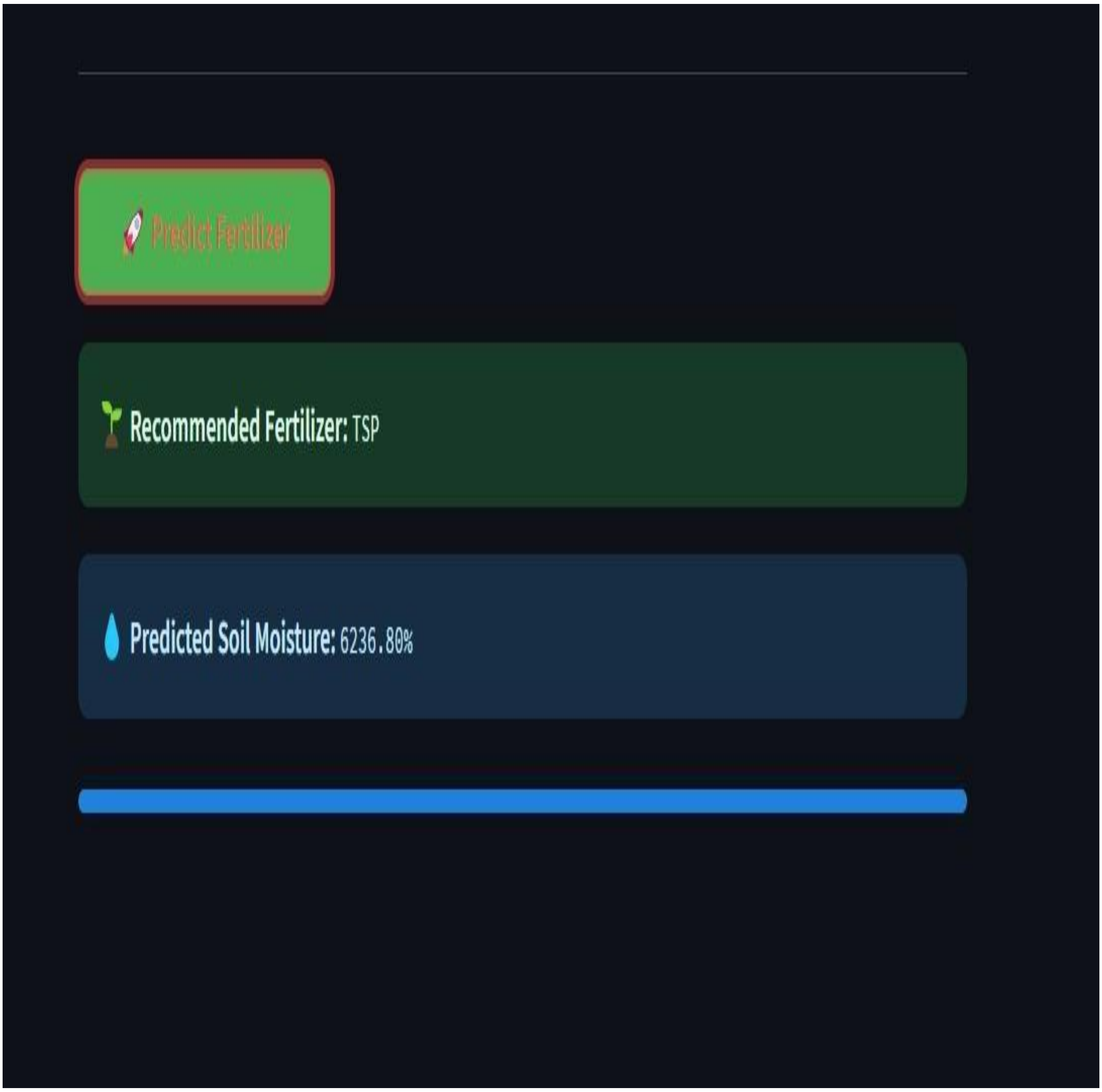



**Fig 7.10**Result Display Section Showing Recommended Fertilizer and Predicted

# CHAPTER 8
# CONCLUSION AND FUTURE WORK

Past research has suggested a combination of machine learning to provide soil moisture predictions and fertilization advice in terms of the ensemble learning method model and deep learning neural network for farm decision-making. The most probable estimation of soil moisture level is conducted via the weighted ensemble approach of Random Forest, XGBoost, and LightGBM models. Then, based on the softmax regression function, these predictions are input into a fertilizer recommendation system by deep neural networks, which directly supports soil nutrient management under diverse conditions.

The experimental results indicate that, the ensemble model integrates individual models with greater predictive accuracy, greatly enhance the accuracy of soil moisture estimation. Apart from it, deep neural network-based fertilizer prediction system generates fewer efficient and effective fertilizers, leading to enhanced crop yield and resource utilization. Machine learning methods are being incorporated into agriculture, where they are not only enhancing production but also promoting sustainable agriculture and reducing wastage of water and fertilizers.

Future research can focus on the performance of larger datasets such as the incorporation of the real-time soil sensor data, fusion of the weather forecast models, and increasing the adaptability of the system to varied soil types. It would also provide farmers with real-life evidence from precision agriculture by releasing the model as a mobile or cloud-based application.

# CHAPTER 9
# REFERENCES

[1] Teshome, Fitsum & Bayabil, Haimanote & Schaffer, Bruce & Ampatzidis, Yiannis & Hoogenboom, Gerrit. (2024). Improving soil moisture prediction with deep learning and machine learning models. Computers and Electronics in Agriculture. 226. 10.1016/j.compag.2024.109414

[2] B. Pujitha, K. Radha, K. Samitha, N. Prathyusha and P. Vasanthi, "Crop Prediction from Soil Parameters using Light Ensemble Learning Model," 2024 11th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2024, pp. 308-313, doi: 10.1109/SPIN60856.2024.10512319.

[3] D. S. Grandika, H. H. Nuha and A. G. Putrada, "Predicting Soil Moisture Levels with Long Short-Term Memory: A Deep Learning Approach Integrated with Internet of Things Data," 2024 International Conference on Decision Aid Sciences and Applications (DASA), Manama, Bahrain, 2024, pp. 1-6, doi: 10.1109/DASA63652.2024.10836388.

[4] S. Iniyan, M. S. Raja, R. Srinivasan, C. Santhanakrishnan and A. Srinivasan, "Crop and Fertilizer Recommendation System Applying Machine Learning Classifiers," 2023 International Conference on Emerging Research in Computational Science (ICERCS), Coimbatore, India, 2023, pp. 1-6, doi: 10.1109/ICERCS57948.2023.10433972.

[5] V. K. N, S. S, S. Bhat and S. H. Prasad, "Effective Crop Prediction Considering Water And Moisture Needs Using Machine Learning Techniques," 2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon), Vijaypur, India, 2022, pp. 1-5, doi: 10.1109/NKCon56289.2022.10127028.

[6] S. Vaishnavi, N. Shanmugam, G. Kiran and A. S. Priyadharshini, "Dependency analysis of various factors and ML models related to Fertilizer Recommendation," 2023 Third International Conference on Secure on Secure Computing and Communication ICSCCC) Jalandhar India, 2023, pp. 81-85, doi: 10.1109/ICSCCC58608.2023.10176974

[7] D. Thorat and S. A. Jahirabadkar, "Deep Learning Methods for Fertilizer Forecasting: Review of State of The Art," 2024 International Conference on Intelligent Systems and Advanced Applications (ICISAA), Pune, India, 2024, pp. 1-6, doi: 10.1109/ICISAA62385.2024.10828751.

[8] R. Thendral and M. Vinothini, "Crop And Fertilizer Recommendation to Improve Crop Yield using Deep Learning," 2023 Second International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2023, pp. 1123-1130, doi: 10.1109/ICEARS56392.2023.10085054.

[9] K. Varsha, P. G, S. N and V. J, "Soil Classification, Crop Selection and Prediction of Fertilizer based on Soil Series," 2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC), Bengaluru, India, 2022, pp. 1174-1178, doi: 10.1109/IIHC55949.2022.10059636.

[10] M. N. Islam and D. Logofătu, "Machine Learning Models to Predict Soil Moisture for Irrigation Schedule," 2022 24th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Hagenberg / Linz, Austria, 2022, pp. 226-232, doi: 10.1109/SYNASC57785.2022.00043.

[11] G. Portal et al., "60-m Resolution Soil Moisture Estimation Based on a Multisensor Feedforward Neural Network Model," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 17, pp. 15543-15566, 2024, doi: 10.1109/JSTARS.2024.3450513

[12] K. Joshi, R. Singh and N. Kumar, "Cloud Based Real Time Soil Moisture Content Monitoring Using IoT and Unmanned Aerial Vehicles," 2023 Sixth International Conference of Women in Data Science at Prince Sultan University(WiDS PSU), Riyadh, Saudi Arabia, 2023, pp. 206-210 doi: 10.1109/WiDS-PSU57071.2023.00050

# CHAPTER 10
# PUBLICATIONS

AIMLA 2025 – Paper Acceptance, Registration & Payment Instructions - Reg Inbox x

Microsoft CMT <noreply@msr-cmt.org>
to me

Mon 14 Apr, 18:28 (6 days ago)

Dear Authors,

***[Please Discard this mail if you already completed the Camera Ready Submission]

Remainder For Camera ready Submission

We are pleased to inform you that your paper titled
"1356 - Integrated Deep Neural Networks and Ensemble Models for Accurate Soil Moisture Prediction and Fertilizer Guidance"
has been accepted for presentation at 3rd International Conference on Artificial Intelligence & Machine Learning Applications, scheduled on April 29 & 30, 2025, at K.S.Rangasamy College of Technology, Tiruchengode, Namakkal, Tamil Nadu, India.

---

📌 Important Submission Instructions

You are kindly requested to:

1. Submit your E-Copyright form and Camera-Ready Paper in Microsoft CMT and via the following Google Form:
🔗 https://forms.gle/YdYWvpQ8rTG9rsPi8
2. Ensure your paper is formatted using the IEEE prescribed template:
🔗 https://www.ieee.org/conferences/publishing/templates.html.
3. Pay the conference fee and upload the transaction proof in the Google Form on or before April 15, 2025.

---

💡 Kindly Note:

- We kindly request that the payment be made through NEFT or RTGS. Please avoid from using UPI, as it may not be supported by all banks.

---

📝 Submission Checklist

1. Camera-ready paper must be submitted both in Microsoft CMT and the Google Form.

# CHAPTER 11
# PLAGIARISM REPORT