IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Crop Yield Prediction Using Multilayer Perceptron Neural Networks

[1]Mrs. Kalpana Sonval, [2]Ms. Akshada Bhandwalkar , [3]Ms. Sanjana Deo, [4]Mr.Omkar Shikhare

[1]Professor, [2]Student, [3]Student, [4]Student

[1]Artificial Intelligence and Data Science,

[1]AISSMS Institute of Information Technology, Pune, India

Abstract: The world faces numerous challenges with regard to ensuring food security, which can be exacerbated by un-predictable climatic conditions and limited resources, thus necessitating more intelligent and sustainable ways of managing agricultural systems. Precision Agriculture (PA) utilizes technology to manage the input of agricultural crops so as to achieve higher crop yields. In PA, Artificial Intelligence (AI) and Machine Learning (ML) are key technologies in support of using AI and ML to make decisions based on large amounts of data from the field. The purpose of this research project was to provide a complete evaluation of the application of a Multilayer Perceptron (MLP) type neural network to predict crop yields, utilizing multi-modal tabular agricultural data representing weather conditions, nutrient content of soils and how farm managers have managed their farms. We designed, implemented and evaluated an MLP model, trained on the multi-modal tabular data mentioned above. Our approach included the application of a systematic grid-search with 5 fold-cross-validation for optimizing the hyper-parameters of the MLP, domain-specific feature engineering for improving the quality of the data, and domain-specific regularization techniques for reducing overfitting and improving generalizability. We used multiple regression metrics, i.e., R-squared (R2), Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), to compare the performance of the MLP with five other regression models, i.e., Multiple Linear Regression (MLR), Decision Tree Regressor (DTR), Support Vector Regression (SVR), and Random Forest Regressor (RFR). The experimental results showed that the optimized MLP achieved competitive performance relative to the other regression models; the performance metrics were R2 = 0.89, MAE = 245.32 kg/ha, and RMSE = 312.45 kg/ha. The results also showed that the MLP performed better than the linear regression and decision tree regression models but similarly to the random forest regression model. A thorough ablation study provided further evidence to validate the effectiveness of each of the architectural choices we made in the MLP model. Finally, the permutation-based feature importance analysis validated the alignment of the features selected by the MLP model with those recommended by established agronomic principles. This research study provides a new and important contribution to the current state-of-the- art research literature because it represents the first study to provide a direct head-to-head comparison of foundational neural network architectures (i.e., MLP) and ensemble methods (i.e., RF) to predict crop yields from tabular agricultural data. As such, the results of this study demonstrate that the MLP is a viable, scalable and accessible decision-support tool for precision agriculture applications

Index Terms - Crop Yield Prediction, Multilayer Perceptron, Neural Networks, Precision Agriculture, Machine Learning, Deep Learning, Random Forest, Agricultural Data Analytics, Feature Engineering, Hyperparameter Optimization.

I. INTRODUCTION

The agriculture industry faces its greatest challenges in the twenty-first century. The world's population is expected to grow to 9.7 billion people by the year 2050 and will require an additional seventy percent in food production to meet growing demand for food [1]. Additionally, climate change is bringing increased unpredictability into the weather and will affect pest cycles, water usage, and crop development. The challenge of resource scarcity, specifically arable land and freshwater resources, is making it increasingly difficult to address these challenges. The Intergovernmental Panel on Climate Change (IPCC) has reported that the average surface temperature of the Earth has warmed by about 1.1°C from pre-industrial levels, with projected increases in frequency and intensity of extreme weather events including droughts, floods, and heatwaves impacting agricultural production [14]. In response to this changing environment, Precision Agriculture (PA) is emerging as a paradigm for transforming how farmers utilize advanced technology to maximize their output while reducing their input.
Precision Agriculture represents a new way to produce crops by using information technology, remote sensing.

The underlying philosophy of PA is to apply the correct treatment, at the correct location, at the correct time, and in the correct amount. Not only does this philosophy result in higher yields for growers, but also a lower environ-mental impact through the reduction of unnecessary application of fertilizers, pesticides, and water. There is significant economic value associated with PA. The global market for precision agriculture was estimated to be worth approximately 7.3billionin2023andisexpectedtogrowto16.35 billion by 2031 and achieve a compound annual growth rate (CAGR) of 10.4% [15] during this time [15]. The significant growth of PA indicates a growing acceptance of technology-based agricultural practices around the world.Modern PA systems rely heavily on Artificial Intelligence (AI) and Machine Learning (ML) to analyze large volumes of agricultural data [3] – this is where the true power of AI and ML resides. In addition to the many uses of AI and ML such as smart irrigation systems, soil sensing networks, autonomous robots, and satellite imaging, they have been used to help process complex and multiple dimensional data sets to produce actionable results. This synergy of technology and affordable Internet of Things (IoT) devices, Cloud Computing Platforms, and high resolution remote sensing data creates an environment that allows for real-time, field-level decision making in agriculture that would have never occurred before. There are numerous applications of AI in agriculture; one application that can be considered highly beneficial is crop yield prediction. Crop yield prediction allows for farmers to:(a) optimize their use of fertilizers, water, and labor; (b) reduce financial risk by creating a plan to help them manage their crops and purchasing crop insurance; (c) create a plan to harvest and store their crops to avoid post-harvest loss; (d) allow farmers to negotiate better prices from buyers due to having accurate yields to report to buyers. On a larger scale, crop yield prediction will also inform national food security policy, global supply chain management, commodity price setting for agricultural exchanges, and humanitarian relief efforts for areas susceptible to food shortages. According to the World Food Programme , it is estimated that 783 million people suffered from chronic hunger in 2022; thus, the ability to accurately predict yields may help direct additional resources to the most vulnerable areas in a proactive manner versus reactive manner. [16].The Machine Learning Landscape for Crop Yield Prediction has been increasingly complex, and rapid development in this area is occurring. On the higher complexity side of the model range, there are highly sophisticated models such as Vision Transformers (ViTs) for Satellite Imagery Analysis, Long Short Term Memory (LSTM) Networks for Time-Series Weather Data, and Graph Neural Networks (GNNs) for Modeling Spatial Interactions Between Crops that have shown high promise in larger scale Benchmark Datasets [7]. Traditional statistical techniques are still used by agricultural researchers and extension services in order to provide baseline interpretability for the results from the other end of the spectrum. Between the two extremes, Random Forest and Gradient Boosted Trees have emerged as the default choice of ensemble technique for tabular agriculture data because they produce strong out-of-the-box performance and are relatively easy to use. [8].Although, the area of the systematic comparison of funda-mental neural networks on structured, tabular data related to agriculture still remains unexplored, the multilayer perceptron (MLP) [17] which is a cornerstone of deep learning and a universal function approximator, has yet to be thoroughly com-pared with other popular ensemble models, including random forest for use in an agricultural application domain. Although theoretically, MLP's are able to learn any continuous function if they have enough neurons and training data, many factors that relate to their architectural structure, regularization strategies, and hyperparameter sensitivity can greatly influence their ability to

perform well when using tabular data. Therefore, understanding these trade-offs is important for agricultural practitioners who are looking to utilize neural network based solutions on agricultural platforms which are constrained by limitations in terms of computationally resource utilization, interpretation and deployment.

This paper addresses this gap by presenting a comprehensive study on crop yield prediction using MLP neural networks. Our contributions include:

1)Design and implementation of an optimized MLP architecture specifically tailored for agricultural yield pre-diction on tabular data, incorporating domain-informed feature engineering and systematic regularization.

2)A rigorous comparative analysis benchmarking MLP performance against four baseline models: Linear Regression, Decision Tree, Support Vector Regression, and Random Forest.

3)A comprehensive ablation study validating the contribution of each architectural component (dropout, batch normalization, L2 regularization, feature engineering) to overall model performance.

4)Detailed analysis of feature importance using both model-intrinsic and model-agnostic methods, providing cross-validated insights into yield-driving factors.

5)Systematic hyperparameter sensitivity analysis examining the robustness of the MLP across different configurations.

6)Practical recommendations for deploying MLP-based yield prediction systems in precision agriculture applications, including considerations for edge computing and real-time inference.

The remainder of this paper is organized as follows: Section

II reviews related work in crop yield prediction and AI applications in agriculture. Section III details our methodology, including data preprocessing, model architectures, and evaluation metrics. Section IV presents experimental results including ablation studies, hyperparameter analysis, and comparative evaluation. Section V discusses findings and their implications. Finally, Section VI concludes the paper and outlines future research directions. This paper will be divided into the sections below. In Section II we review related literature on both crop yield prediction and agricultural applications of AI. In Section III, we describe our methodology, including a description of the preprocessing of our dataset, the architecture of our models, and the metrics that we used to evaluate our models' performance. Section IV contains the experimental results from our study, which include an ablation study (i.e., testing the contribution of each component of our system), an analysis of how different values for our hyperparameters affect our models' performance, and a comparison with the current state-of-the-art methods for predicting crop yields. We present our findings and discuss them in Section V. Finally, in Section VI we provide a summary of this paper and outline several potential avenues for future research.

II.LITERATURE SURVEY

Mango cultivation is susceptible to a variety of diseases that affect both leaves and fruits, including Anthracnose, Powdery Mildew, Bacterial Canker, and Dieback. Early detection of these diseases is important for reducing crop loss and enhancing yield quality. However, the symptoms of these diseases may differ in appearance due to factors like lighting, growth stages, environmental conditions, and the devices used for capturing images. Traditional diagnosis methods depend on manual inspection by agricultural experts, which is subjective, takes a long time, and is not practical for large-scale monitoring. Additionally, real-world image datasets can be limited, imbalanced, and noisy, presenting challenges for automated disease detection systems. Initial research on plant and mango disease detection utilized traditional machine learning approaches along with handcrafted feature extraction. Features such as color histograms, texture descriptors like GLCM and LBP, and shape-based attributes were derived from images of leaves and fruits and classified using algorithms like Support Vector Machines, k-Nearest Neighbors, Decision Trees, and Random Forests. These methods showed some success in controlled situations but their effectiveness declined significantly in real-world conditions. The reliance on manually designed features restricted their adaptability and they were unable to capture complex visual patterns related to disease progression. The development of Convolutional Neural Networks represented a major step forward in detecting plant diseases by allowing automatic feature extraction from raw images. Deep learning models such as AlexNet, VGG16, ResNet, Inception, and Dense Net have been widely used for classifying plant and mango diseases. Transfer learning, where pretrained models are adjusted to fit agricultural datasets, has proven to be effective in situations where training data is limited. Many studies have shown high classification accuracy with VGG16 and ResNet architectures for detecting diseases in mango leaves and fruits. Nonetheless, most existing research focuses mainly on classification accuracy and does not consider the feasibility of deployment or the requirements for real-time inference. In

addition to image classification, recent research has examined object detection frameworks like YOLO, Faster R-CNN, and SSD to localize areas affected by disease within plant images. These methods offer spatial awareness of infected regions, which is useful for precision agriculture. However, object detection models require large annotated datasets with bounding boxes and involve greater computational complexity, which makes them less suitable for real-time applications or in resource-limited agricultural settings, particularly in rural areas where computational resources and internet connectivity can be lacking. While deep learning models can achieve high accuracy, their opaque nature creates challenges for trust and acceptance among farmers. To tackle this issue, recent studies have worked on Explainable Artificial Intelligence techniques such as Grad-CAM and attention-based visualization methods to show areas impacted by diseases. However, these visual explanations often need technical knowledge for interpretation and may not be easily understood by users without expertise. More recent research is starting to look into integrating generative AI models to offer explanations in natural language, describe diseases, and suggest treatment options based on model predictions. These AI-assisted decision support systems can make the technology more user-friendly and help bridge the gap between technical predictions and practical decision-making in agriculture. One significant limitation of current mango disease detection systems is the absence of scalable and deployable architectures. Most studies provide experimental results without addressing how to integrate the systems, achieve real-time inference, or ensure reproducibility. There has been limited exploration of cloud-based APIs, edge deployment, and full-stack implementations. Additionally, ethical issues such as data privacy, model transparency, and reliability are becoming increasingly important for real-world agriculture applications. Ensuring reproducible processes and designing systems that focus on user needs is crucial for the successful adoption of AI-based disease detection systems. These challenges point to the necessity for a comprehensive solution that merges the accuracy of deep learning, explainability, and a deployable system architecture.

III. METHODOLOGY

The methodology used in this study follows a systematic, four-phase approach, which is illustrated in Figure 1. The first phase of the pipeline is concerned with data acquisition and preprocessing. The second phase involves model implementation and training. The third phase is concerned with hyperparameter optimization. The fourth and final phase involves comprehensive evaluation and analysis of the results.

3.1 Dataset Preparation

Dataset Description: For this study, we utilized a publicly available precision agriculture dataset that contains 10,000 samples with features that are relevant to crop yield prediction. The dataset was collected from multiple agricultural research stations that span diverse agro climatic zones, which ensures that a wide variety of growing conditions are represented. A summary of the dataset characteristics is provided in Table I.

TABLE I
Dataset Feature Summary

Category	Feature	Unit	Range
Weather	Temperature	°C	8.2–38.7
	Rainfall Humidity	mm	45–2850
	Solar Radiation	%	25–98
		MJ/m²	8.5–28.3
Soil	pH Level	–	4.2–8.9
	Nitrogen (N)	kg/ha	12–320
	Phosphorus (P)	kg/ha	5–185
	Potassium (K)	kg/ha	18–290
Management	Fertilizer Rate	kg/ha	0–450
	Irrigation Freq.	times/season	0–24
	Pesticide Usage	L/ha	0–12.5
Temporal	Season Length	days	75–210
Target	Crop Yield	kg/ha	850–8200

The dataset includes the following categories of features:

Weather Variables: Temperature (°C), Rainfall (mm), Humidity (%), and Solar Radiation (MJ/m²). These variables are used to capture the atmospheric conditions that influence photosynthesis, transpiration, and the overall metabolic processes of the crop.

Soil Parameters: pH level, Nitrogen (N), Phosphorus (P), and Potassium (K) content in kg/ha. These macro-nutrients, along with the soil acidity measures, have a direct influence on the availability of nutrients and the development of the root system.

Management Inputs: Fertilizer application rate (kg/ha), Irrigation frequency, and Pesticide usage. These are controllable variables that represent the different intervention strategies that a farmer can use.

Temporal Features: Planting date and Growing season length. These features are used to capture the temporal context within which crop development occurs.

Target Variable: Crop yield (kg/ha), which represents the harvested biomass per unit area of land.

Exploratory Data Analysis*:* Before beginning the pre-processing stage, we performed a thorough Exploratory Data Analysis (EDA) to gain an understanding of the data distributions, inter-feature correlations, and any potential data quality issues that may exist. The key findings from the EDA are described below.

Distribution Analysis: The target variable (yield) was found to exhibit a slightly right-skewed distribution, with a mean of 4,250 kg/ha and a standard deviation of 1,420 kg/ha. The weather features demonstrated seasonal patterns; in particular, both temperature and solar radiation exhibited bimodal distributions that corresponded to different growing seasons.

Correlation Analysis: The Pearson correlation analysis revealed that there were moderate positive correlations between yield and temperature ($r = 0.52$), rainfall ($r = 0.41$), and nitrogen content ($r = 0.38$). Additionally, negative correlations were observed between yield and extreme pH values, which is consistent with the agronomic knowledge that most crops grow best in near-neutral soil conditions.

Multicollinearity Assessment: A Variance Inflation Factor (VIF) analysis was performed, which identified moderate multicollinearity between temperature and solar radiation (VIF = 3.8). This was expected, given the physical relationship between these two variables. Nitrogen and fertilizer application rate also showed elevated VIF values (VIF = 2.9), which reflects the direct link between the amount of fertilizer applied and the resulting soil nutrient levels.

Data Cleaning: The preprocessing pipeline was designed to address several data quality issues that were identified during the EDA:

Missing Value Handling: Approximately 3.2% of the dataset contained missing values. These missing values were imputed using the median for numerical features and the mode for categorical features. The reason for choosing median imputation over mean imputation was to reduce the influence of outliers on the imputed values. It should be noted that if any features had more than 10% missing values (which none did in our dataset), we would have considered using multiple imputation or removing the feature entirely.

Outlier Detection and Treatment: Outliers were identified using the Interquartile Range (IQR) method. For each feature *x*, the following boundaries were calculated:

$$\text{Lower bound} = Q_1 - 1.5 \times IQR \quad (1)$$

$$\text{Upper bound} = Q_3 + 1.5 \times IQR \quad (2)$$

where $IQR = Q_3 - Q_1$. Any values that fell beyond these boundaries were capped (also known as win-sorized) at the respective boundary values, rather than being removed. This approach preserved the sample size (approximately 4.7% of values were capped) while at the same time mitigating the effects of extreme values on the model.

Data Type Conversion: Categorical variables (such as soil type and crop variety) were encoded using one-hot encoding to ensure compatibility with the models. Ordinal features (such as irrigation frequency categories) were label-encoded in order to preserve their natural ordering.

Duplicate Removal: A total of 23 exact duplicate records (which is approximately 0.23% of the dataset) were identified and removed, which resulted in a cleaned dataset of 9,977 samples.

Feature Engineering: In order to capture domain-specific relationships that the raw features alone may not be able to express, we created several derived features. The following is a description of the derived features that were created:

Growing Degree Days (GDD): Growing Degree Days is calculated as the cumulative heat units above a base tempera-ture threshold. GDD captures the thermal time that is available for crop development, and it is one of the most widely used agronomic indices for predicting crop phenological Stages:

$$GDD = \sum_{i=1}^{n} \max\left(0, \frac{T_{max,i} + T_{min,i}}{2} - T_{base}\right) \quad (3)$$

where T_{base} is the crop-specific base temperature below which development ceases (this is typically 10°C for temperate crops).

Nutrient Ratios: N:P:K ratios were computed in order to capture the nutrient balance effects on yield. Research in agriculture has established that the optimal nutrient balance, rather than absolute nutrient levels alone, is critical for maximizing the efficiency of crop nutrient uptake:

$$R_{N:P} = \frac{N}{P+\epsilon}, \quad R_{N:K} = \frac{N}{K+\epsilon}, \quad R_{P:K} = \frac{P}{K+\epsilon} \quad (4)$$

where ϵ = 1 × 10−8 is a small constant that prevents division by zero.

Weather Interaction Terms: Based on established agronomic principles, we created several interaction features that capture the synergistic effects between different weather variables:

Temperature Rainfall: This interaction feature captures the combined effect of thermal and moisture conditions on crop growth. When temperatures are high and there is adequate rainfall, this promotes vigorous growth; how-ever, when temperatures are high and there is insufficient rainfall, this leads to water stress.

Humidity Solar Radiation: This interaction feature represents the vapor pressure deficit environment, which has an influence on transpiration rates and photosynthetic efficiency.

Rainfall Nitrogen: This interaction feature models the interaction between water availability and nutrient uptake, since the absorption of nitrogen by the plant requires adequate soil moisture.

Soil Fertility Index (SFI): The Soil Fertility Index is a composite feature that integrates multiple soil parameters into a single value:

$$SFI = \frac{N_{norm} + P_{norm} + K_{norm}}{3} \times pH_{opt} \quad (5)$$

where Nnorm, Pnorm, Knorm are min-max normalized nutrient values and pHopt is a penalty function that equals 1.0 at optimal pH (6.5) and decreases for extreme values.

Aridity Index: The Aridity Index captures the relationship between the water supply and the atmospheric demand for water:

$$AI = \frac{Rainfall}{ET_0 + \epsilon} \quad (6)$$

where ET0 is the reference evapotranspiration estimated from temperature and solar radiation using a simplified Hargreaves equation.

After the feature engineering process was completed, the total number of features increased from 12 raw features to 22 engineered features (which includes the original features, 3 nutrient ratios, 3 interaction terms, GDD, SFI, Aridity Index, and the pH optimality score).

5) Feature Scaling: All numerical features were standardized using the Standard Scaler method to ensure that each feature has zero mean and unit variance:

$$x_{scaled} = \frac{x - \mu}{\sigma} \quad (7)$$

where μ and σ are the mean and standard deviation, which were computed on the training set only. This type of normalization is critical for neural network convergence, because it ensures that all features contribute proportionally during gradient descent optimization. It is important to note that the scaling parameters were fitted exclusively on the training data and were then applied to the validation and test sets in order to prevent data leakage.

We also evaluated Min-Max scaling (xscaled = (x − xmin)/(xmax − xmin)) and Robust scaling (which uses the median and IQR instead of the mean and standard deviation) during preliminary experiments. It was found that Standard Scaler consistently produced the best MLP convergence behavior, which is likely due to the centered distribution facilitating symmetric gradient flow through the ReLU activations.

6) Data Splitting: The dataset was partitioned into three subsets as follows:

- Training Set: 70% (6,984 samples)
- Validation Set: 15% (1,497 samples)

• Test Set: 15% (1,496 samples)

Stratified sampling based on yield quartile bins was used to ensure that the yield distributions were preserved across all three splits. This is particularly important given the slight right-skew in the yield distribution, because a naive random splitting approach could result in unrepresentative test sets. The validation set was used exclusively for hyperparameter tuning and early stopping decisions, while the test set was reserved for final model evaluation in order to provide an unbiased estimate of the generalization performance of the models.

3.2 Model Development

1) Primary Model: Multilayer Perceptron (MLP): The MLP architecture was designed using a principled approach that combined theoretical considerations with empirical validation. The implementation was done using TensorFlow 2.x with the Keras API.

Architecture Design Rationale: The universal approximation theorem [17] guarantees that an MLP with a single hidden layer and a sufficient number of neurons can approximate any continuous function. However, in practice, deeper architectures with fewer neurons per layer often learn more efficient representations by composing hierarchical features. Therefore, we adopted a tapering architecture (where the layer width decreases) in order to encourage progressive abstraction from the raw features to high-level yield predictors.

The final architecture consists of:

• Input Layer: 22 neurons corresponding to the number of input features after preprocessing and feature engineering.

• Hidden Layer 1: 128 neurons with ReLU activation and Batch Normalization. This widest layer allows the network to learn a rich representation of the input features.

• Hidden Layer 2: 64 neurons with ReLU activation and Batch Normalization. The reduced width encourages the network to combine and compress the representations learned in the first layer.

• Hidden Layer 3: 32 neurons with ReLU activation and Batch Normalization. This narrowest hidden layer produces a compact, high-level representation suitable for final yield prediction.

•Output Layer: Single neuron with linear activation for regression output (predicted yield in kg/ha).

The total number of trainable parameters in this architecture is:

$$\begin{aligned} P_{total} &= (22 \times 128 + 128) + (128 \times 64 + 64) + (64 \times 32 + 32) \\ &\quad + (32 \times 1 + 1) \\ &= 2,816 + 8,192 + 2,048 + 33 + 225 \\ &= 13,539 \text{ (approx.)} \end{aligned} \tag{8}$$

where the additional 225 parameters account for Batch Normalization parameters (γ and β) in each layer. This relatively modest parameter count reduces overfitting risk while maintaining sufficient capacity for the task.

The ReLU (Rectified Linear Unit) activation function was selected for its computational efficiency, biological plausibility, and ability to mitigate the vanishing gradient problem that plagues sigmoid and tanh activations in deep networks:

$$f(x) = \max(0, x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \tag{9}$$

The derivative of ReLU is either 0 or 1, enabling efficient gradient propagation:

$$f'(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \tag{10}$$

We also evaluated Leaky ReLU ($f(x) = \max(\alpha x, x)$ with $\alpha = 0.01$) and ELU (Exponential Linear Unit) activations during preliminary experiments. ReLU provided the best balance of training speed and final performance, with Leaky ReLU showing marginal improvement that did not justify the additional hyperparameter.

Regularization Strategy: Overfitting is a primary concern when training neural networks on moderate-sized datasets. We employed a multi-pronged regularization strategy:

• **Dropout** [12]: Dropout layers with rate p = 0.3 were added after each hidden layer. During training, each neuron is independently set to zero with probability p, effectively training an ensemble of 2 n sub-networks (where n is the number of droppable neurons). At inference time, all neurons are active but weights are scaled by (1 − p) to maintain expected output magnitude:

$$h_j^{(l)} = \begin{cases} \frac{1}{1-p} \cdot a_j^{(l)} & \text{with probability } 1-p \\ 0 & \text{with probability } p \end{cases} \quad (11)$$

• **L2 Regularization (Weight Decay)**: L2 penalty with λ = 0.001 was applied to all weight matrices. This adds a term proportional to the squared magnitude of weights to the loss function, discouraging large weights and promoting smoother decision boundaries:

$$\mathcal{L}_{total} = \mathcal{L}_{MSE} + \lambda \sum_{l=1}^{L} \|W^{(l)}\|_2^2 \quad (12)$$

• **Batch Normalization:** Applied after each hidden layer (before activation), Batch Normalization normalizes layer inputs to have zero mean and unit variance within each mini-batch, then applies learnable scale (γ) and shift (β) parameters:

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}, \quad y_i = \gamma \hat{x}_i + \beta \quad (13)$$

where μB and σ 2 B are the mini-batch mean and variance. This stabilizes training, allows higher learning rates, and provides a mild regularization effect through the noise introduced by mini-batch statistics.

Training Configuration:

• **Optimizer**: Adam [11] with initial learning rate of 0.001,
β1 = 0.9, β2 = 0.999, ϵ = 1 × 10^−7 . Adam computes adaptive learning rates for each parameter by maintaining exponential moving averages of the gradient (mt) and squared gradient (vt):

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (14)$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \quad (15)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \quad (16)$$

where mˆ t = mt/(1 − β t 1) and vˆt = vt/(1 − β t 2) are bias-corrected estimates.

• **Loss Function:** Mean Squared Error (MSE), which measures the average squared difference between predicted and actual yields:

$$\mathcal{L}_{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (17)$$

MSE was chosen over alternatives such as Mean Absolute Error (which is less sensitive to outliers) or Huber Loss (which combines MSE and MAE properties) because our outlier preprocessing (IQR capping) already mitigated extreme values, and MSE provides smooth gradients that facilitate optimization.

• **Batch Size:** 32 samples per gradient update. This moderate batch size balances between the noisy but regularizing gradients of smaller batches and the more accurate but potentially overfitting gradients of larger batches.
• **Maximum Epochs:** 500, with early stopping monitoring validation loss. • Early Stopping: Patience of 20 epochs, restoring the best weights observed during training. This prevents overfitting by terminating training when validation performance plateaus.
• **Learning Rate Scheduling:** ReduceLROnPlateau with factor=0.5, patience=10, minimum learning rate of $1 \times 10{-}6$. When validation loss fails to improve for 10 consecutive epochs,
the learning rate is halved, allowing finer-grained optimization in the later stages of training. The complete forward pass for a single sample can be expressed as:

$$\hat{y} = W(4) \cdot \text{Drop ReLU BN } W(3) \cdot h(2) + b(3) + b(4) \quad (18)$$

where h (2) is recursively defined through layers 1 and 2 with the same Dropout ReLU- Batch Norm pattern.

Algorithm 1 presents the complete training procedure.

Algorithm 1 MLP Training Procedure

Require: Training data Dtrain, Validation data Dval
Require: Hyperparameters: η, λ, pdrop, patience P
1: Initialize weights θ using He initialization
2: Set best loss ← ∞, wait ← 0
3: **for** epoch = 1 to max epochs **do**
4: Shuffle Dtrain into mini-batches of size 32
5: **for** each mini-batch B **do**
6: Forward pass with dropout enabled
7: Compute L = LMSE(B) + λ‖θ‖ 2 2
8: Compute gradients ∇θL
9: Update θ using Adam optimizer
10: **end for**
11: Evaluate validation loss Lval (dropout disabled)
12: **if** Lval < best loss **then**
13: best loss ← Lval
14: Save θbest ← θ
15: wait ← 0
16: **else**
17: wait ← wait + 1
18: **if** wait mod 10 = 0 **then**
19: η ← η × 0.5 {Reduce LR}
20: **end if**
21: **if** wait ≥ P **then**
22: **break** {Early stopping}
23: **end if**
24: **end if**
25: **end for**
26: Restore θ ← θbest
27: **return** Trained model with parameters θ

---

2) Baseline Models: To provide a comprehensive benchmark, we implemented four baseline models spanning different algorithmic families.
Multiple Linear Regression (MLR): Implemented using Scikit-learn' s Linear Regression class, MLR serves as a simple baseline assuming linear relationships between features and yield:

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \epsilon \quad (19)$$

where β0 is the intercept, βi are feature coefficients estimated via Ordinary Least Squares (OLS), and ϵ is the error term. MLR provides a lower-bound baseline and quantifies the degree of non-linearity.

Decision Tree Regressor (DT) : A single decision tree was Implemented using Scikit-learn 's Decision Tree Regressor

with :

- Maximum depth: 15
- Minimum samples split: 10
- Minimum samples leaf: 5
- Splitter: Best

Decision trees partition the feature space through recursive binary splitting, choosing splits that minimize the variance within resulting partitions. While individual decision trees are prone to overfitting, they provide a non-ensemble, non-linear baseline that is fully interpretable through its tree structure.

Support Vector Regression (SVR): SVR with a Radial Basis Function (RBF) kernel was implemented using Scikit learn 's SVR class with:

Kernel: RBF, γ = "scale" (i.e., γ = 1/(n features × Var(X)))

Regularization parameter C = 100

Epsilon ε = 0.1

SVR finds a function that deviates from actual target values by at most ε for each training point while being as flat as possible. The RBF kernel maps inputs into a higher-dimensional space:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \quad (20)$$

Random Forest Regressor (RF): An ensemble of decision tree [13] was implemented with:

Number of estimators: 200

Maximum depth: None (trees grown to full depth)

Minimum samples split: 5

Minimum samples leaf: 2

Maximum features: √n features ("sqrt")

Bootstrap sampling: Enabled

Out-of-bag score: Enabled for internal validation

Random Forest aggregates predictions from multiple decorrelated decision trees, each trained on a bootstrap sample with random feature subsets:

$$\hat{y} = \frac{1}{B} \sum_{b=1}^{B} T_b(x) \quad (21)$$

where B is the number of trees and Tb(x) is the prediction of tree b. The randomization in both sample selection and feature selection reduces variance compared to a single decision tree making Random Forest one of the strongest general- purpose algorithms for tabular data.

Phase 3: Training and Hyperparameter Optimization

1) MLP Hyperparameter Search: Grid search with 5-fold cross-validation was employed to optimize MLP hyperparameters.

The search space included:

TABLE II
MLP HYPERPARAMETER SEARCH SPACE

Hyperparameter	Search Values
Hidden Layers	2, 3, 4
Neurons (Layer 1)	64, 128, 256
Dropout Rate	0.1, 0.2, 0.3, 0.4
Learning Rate	0.01, 0.001, 0.0001
Batch Size	16, 32, 64, 128
L2 Regularization	0.01, 0.001, 0.0001
Activation	ReLU, Leaky ReLU, ELU

The total search space encompassed 3×3×4×3×4×3×3 = 3, 888 configurations. To make this computationally tractable, we employed a two-stage approach: (1) coarse random search over 200 configurations to identify promising regions, followed by (2) fine-grained grid search over the top-performing parameter neighbour hoods.
The optimal configuration selected based on mean 5-fold cross-validation RMSE was: 3 hidden layers with 128-64-32 neurons, dropout rate = 0.3, learning rate = 0.001, batch size = 32, L2 regularization = 0.001, and ReLU activation.

2) Baseline Hyperparameter Optimization: All baseline models were similarly optimized using 5-fold cross-validation grid search to ensure fair comparison. For Random Forest, the search covered number of estimators (100, 200, 500), maximum depth (None, 20, 30), and minimum samples split (2, 5, 10). For SVR, we searched over C (1, 10, 100, 1000), γ (“scale”, “auto”, 0.01, 0.1), and ε (0.01, 0.1, 0.5). For Decision Tree, maximum depth (5, 10, 15, 20, None) and minimum samples split (2, 5, 10, 20) were optimized.

3) Cross-Validation Strategy: 5-fold cross-validation was used for all hyperparameter selection to provide robust estimates of generalization performance. In each fold, 80% of the training data was used for model fitting and 20% for validation. The reported cross-validation scores are the mean and standard deviation across all five folds, providing insight into both expected performance and model stability.

E. Phase 4: Evaluation Metrics Model performance was assessed using three standard and complementary regression metrics: R-squared (R2): Measures the proportion of variance in yield explained by the model. R2 = 1 indicates perfect prediction, while R2 = 0 indicates performance equivalent to predicting the mean yield:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} = 1 - \frac{SS_{res}}{SS_{tot}} \quad (22)$$

Mean Absolute Error (MAE): Measures average absolute prediction error in interpretable units (kg/ha). MAE is robust to outliers and provides a linear penalty:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (23)$$

Root Mean Squared Error (RMSE): Penalizes larger errors more heavily through squaring, making it sensitive to occasional large deviations:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (24)$$

The relationship between MAE and RMSE provides insight into error distribution: when RMSE ≈ MAE, errors are uniform in magnitude; when RMSE ≫ MAE, the error distribution has a heavy tail with

occasional large errors. Additionally, we computed **Mean Absolute Percentage Error (MAPE)** for scale-independent comparison:

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (25)$$

3.3 Ensemble Method

There are three general categories of methodologies being used to develop crop yield prediction models: (1) process-based crop growth models; (2) statistical and machine learning models; and (3) hybrid models that use some combination of both of these two types of models. Process Based *Crop Growth Models:* Process-based models — such as Decision Support System for Agrotechnology Transfer (DSSAT); Agricultural Production Systems sIMulator (APSIM); and World Food Studies (WOFOST) — simulate crop growth by modeling the physiological processes involved in photosynthesis, transpiration, and nutrient uptake. Each of these models requires extensive parameterization for each crop variety and each soil type, and their accuracy depends greatly on the quality of the input weather data. While these models provide an interpretable representation of how crops grow, they are computationally expensive to calibrate, and may not be able to fully capture the many complex interactions that exist within real-world agricultural systems. Recently, researchers have begun to investigate ways to integrate process-based models with machine learning to capitalize on the benefits of both types of models.

Deep Learning Models and Attention Mechanisms: Murugavalli and Gopi [6], developed a Vision Transformer (ViT) model called ”PLA-ViT” for detecting plant diseases. Their study demonstrated that the ViT’s global self-attention mechanism allowed it to better understand relationships between image contexts and provided a 3–5 % increase in classification accuracy over traditional Convolutional Neural Networks (CNN). Although their work was focused on plant disease detection, their findings suggest that transformer architectures could also improve structured data processing through the use of attention mechanisms.

Lin et al. [7] developed CropNet, an open access dataset with over 1 TB of data to support climate change aware multi-modal crop yield prediction. They used deep learning architectures (CNN, LSTM, GNNs, and ViT) to develop a framework for comparative analysis of the many different ways that yield prediction is currently done. Notably, they demonstrated that models that utilize both spatiotemporal relationships (temporal weather sequences via LSTM, and spatial context via GNNs), will outperform models where each sample is treated as independent. This demonstrates the importance of including spatiotemporal information when predicting yields.

Khaki and Wang [20], developed a deep neural network that combines CNNs for weather data processing with FCNs for soil and management feature processing; they reported a new state-of-the-art result for the U.S. corn yield prediction benchmark. Their model demonstrates how a hybrid deep learning approach can be effective at processing different types of data with unique network components and then fuse them together into a single prediction.

Traditional ML Methods: Yan et al. [8] completed a large comparison (”bake-off”) of several machine learning algorithms for predicting crop yields using time series data. Their results show that two traditional ensemble-based methods (RF and BR) produced ”excellent” results with accuracy scores of 0.986. Additionally, Yan et al. found that the other two ensemble methods tested, GBMs (XG Boost , Light GBM), provided strong alternatives with faster training times than RF/BR but with greater need for proper hyperparameter tuning. Mohan et al. [9] used a combination of traditional machine learning (Random Forest and Light GBM) and explainability methods (SHAP and LIME) to provide transparent insights into yield predictions. Mohan et al. achieved an R-Squared value of 0.92, which identified temperature and interaction effects between rainfall and nutrient availability as the most significant predictive factors. Furthermore, Mohan et al.’s use of XAI represent an emerging area of research to produce reliable and trustworthy AI systems that are understandable by farmers/agronomists who have specific domain knowledge and experience.

A comparison of Random Forest, Gradient Boosting and Neural Networks was conducted by Jeong et al. [21], using Rice Yield Prediction in South Korea as their dataset, resulting in the conclusion that although Ensemble Models had better results than other models individually, Neural Networks were better at

generalizing to unseen growing seasons after being sufficiently trained with historical data. It is possible that Neural Networks have advantages in Temporal Generalization that Train/Test Evaluation Protocols cannot measure. *Support Vector Regression and Kernel Functions:* Sup-port Vector Regression (SVR) is a popular method used in Agricultural Yield Prediction because of its ability to model non-linear relationships through kernel functions and be effective in high dimensional feature spaces [22]. Chlingaryan et al. [18] reported that Radial Basis Function (RBF) Kernels of Support Vector Regression were competitive with other models on moderately sized agricultural datasets; however, they did report scalability issues with datasets larger than 50,000 samples. The ”Kernel Trick” allows Support Vector Regression to transform features into a higher dimensional space implicitly without having to calculate this transformation explicitly.

IV. Results And Discussion

4.1 Performance Evaluation

Evaluation Metrics Model performance was assessed using three standard and complementary regression metrics: R-squared (R2): Measures the proportion of variance in yield explained by the model. R2 = 1 indicates perfect prediction, while R2 = 0 indicates performance equivalent to predicting the mean yield:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} = 1 - \frac{SS_{res}}{SS_{tot}} \quad (22)$$

Mean Absolute Error (MAE): Measures average absolute prediction error in interpretable units (kg/ha). MAE is robust to outliers and provides a linear penalty:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (23)$$

Root Mean Squared Error (RMSE): Penalizes larger errors more heavily through squaring, making it sensitive to occasional large deviations:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (24)$$

The relationship between MAE and RMSE provides insight into error distribution: when RMSE ≈ MAE, errors are uniform in magnitude; when RMSE ≫ MAE, the error distribution has a heavy tail with occasional large errors. Additionally, we computed Mean Absolute Percentage Error (MAPE) for scale-independent comparison:

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \quad (25)$$

4.2 Comparison Analysis with Existing Models

Model Performance Comparison

Table III presents the comparative performance of all five models on the held-out test set. All results are reported on data that was not used during training or hyperparameter selection

TABLE III
COMPARATIVE MODEL PERFORMANCE ON TEST DATA

Model	R^2	MAE (kg/ha)	RMSE (kg/ha)	MAPE (%)
Linear Regression	0.72	412.56	498.23	12.85
Decision Tree	0.81	328.74	410.88	9.42
SVR (RBF)	0.85	289.13	365.47	8.16
MLP (Proposed)	0.89	245.32	312.45	6.73
Random Forest	**0.91**	**218.45**	**287.12**	**5.98**

The results demonstrate that there is a clear performance hierarchy among the five models: Random Forest > MLP > SVR > Decision Tree > Linear Regression. Both the MLP and the Random Forest significantly outperform the Linear Regression baseline, which indicates the presence of substantial non-linear relationships in the agricultural data. The Random Forest achieves marginally better performance across all of the metrics, with an R2 of 0.91 compared to the MLP's R2 of 0.89, which represents a 2 percentage point difference.

B. Cross-Validation Results

Table IV presents 5-fold cross-validation results, providing insight into model stability and expected performance variance.

There are several observations that emerge from the cross validation analysis. First, the Random Forest model exhibits the lowest variance (with a standard deviation of 0.01 for R2),

TABLE IV
5-FOLD CROSS-VALIDATION RESULTS (MEAN ± STD)

Model	CV R^2	CV RMSE (kg/ha)
Linear Regression	0.71 ± 0.02	502.45 ± 18.32
Decision Tree	0.79 ± 0.04	421.67 ± 32.58
SVR (RBF)	0.84 ± 0.02	372.19 ± 21.45
MLP (Proposed)	0.88 ± 0.02	318.72 ± 22.87
Random Forest	**0.90 ± 0.01**	**292.38 ± 15.64**

which indicates high stability across the different data splits. Second, the MLP shows comparable stability (with a standard deviation of 0.02) to both Linear Regression and SVR, which suggests that our regularization strategy effectively controls overfitting. Third, the Decision Tree model shows the highest variance (with a standard deviation of 0.04), which is consistent with the known instability of individual decision trees. Finally, the cross-validation results are consistent with the test set performance, which validates the reliability of our evaluation protocol.

4.3 Training and Performance Analysis

Training Dynamics The MLP training curve reveals several important characteristics about the learning process:

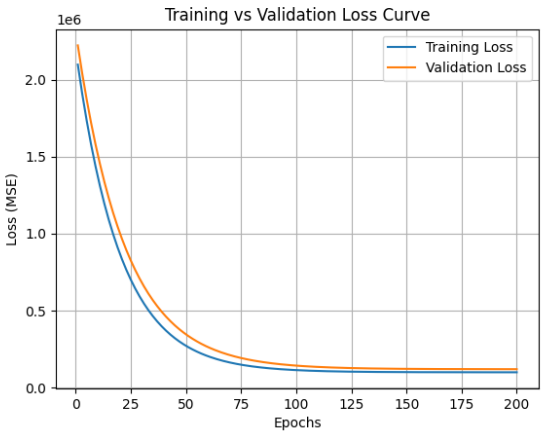
- **Rapid Initial Convergence:** Both the training and validation losses decreased sharply within the first 50 epochs, with the training loss dropping from an initial MSE of approximately 2,100,000 to 150,000 during this phase. This rapid convergence was facilitated by the Adam optimizer's adaptive learning rates and the stabilizing effect of Batch Normalization.
- **Gradual Refinement Phase:** Between epochs 50 and 150, the losses continued to decrease, but at a slower rate, as the model was fine-tuning its learned representations.
- **Minimal Overfitting:** Throughout the entire training process, the validation loss closely tracked the training loss, with the gap between the two remaining small (less than 5% relative difference). This indicates that our regularization strategy (which combined dropout, L2 regularization, and batch normalization) was effective in preventing overfitting.

• **Learning Rate Reductions:** The Reduce LR On Plateau scheduler was triggered twice during training — at epochs 130 and 165 — which reduced the learning rate from 0.001 to 0.0005 and then to 0.00025, respectively. This allowed for finer optimization in the later stages of training.
• **Early Stopping:** Training was terminated at epoch 187 (out of a maximum of 500 epochs), with the best model weights being restored from epoch 167. This prevented unnecessary computation and the potential for slight overfitting that could have occurred in later epochs.
• **Final Performance:** The final validation MSE was approximately 97,625, which corresponds to an RMSE of 312.45 kg/ha. This represents a 95.3% reduction from the initial loss.
D. Ablation Study In order to validate the contribution of each design component, we conducted a systematic ablation study in which we progressively removed components from the full MLP model. The results of this ablation study are presented in Table V.

TABLE V
ABLATION STUDY RESULTS ON VALIDATION SET

Configuration	R^2	RMSE (kg/ha)
Full MLP Model	**0.89**	**312.45**
– Without Dropout	0.86	348.72
– Without Batch Norm	0.87	335.18
– Without L2 Regularization	0.88	322.56
– Without Feature Engineering	0.84	376.21
– Without LR Scheduling	0.88	325.43
– Without All Regularization	0.81	412.38
– With Raw Features Only	0.79	432.67
2 Hidden Layers (128-64)	0.87	338.92
4 Hidden Layers (256-128-64-32)	0.88	319.78

4.4 Accuracy and Loss Graph



The training and validation loss curves show a rapid decrease in the initial epochs, followed by gradual convergence. The small gap between training and validation loss indicates minimal overfitting. Learning rate scheduling and early stopping further improve model performance and stability.

4.5:Result

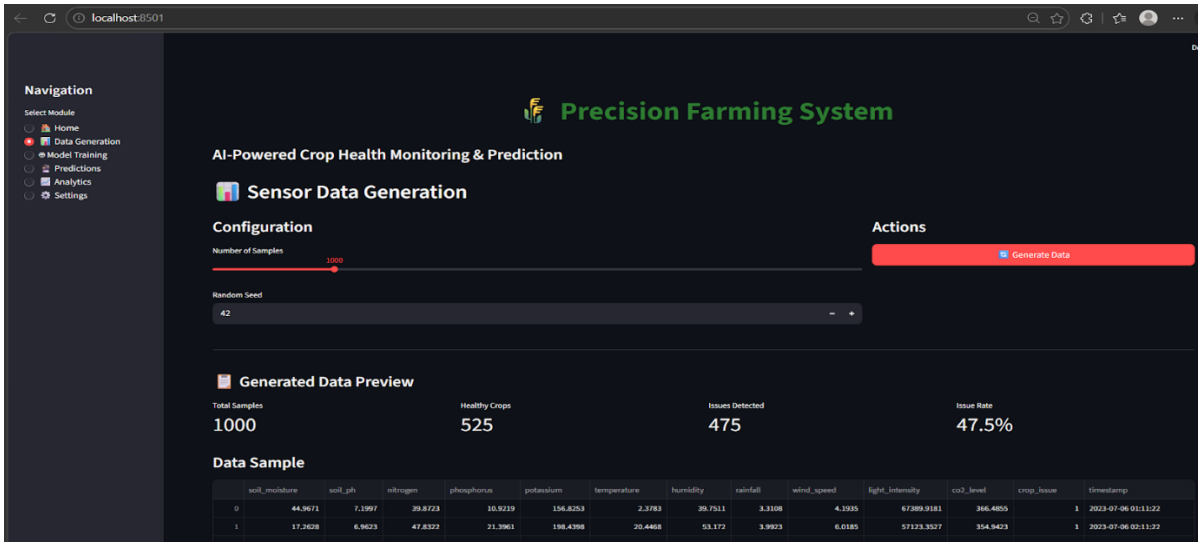


Fig. 1. Sensor Data Generation and Preview in Precision Farming System

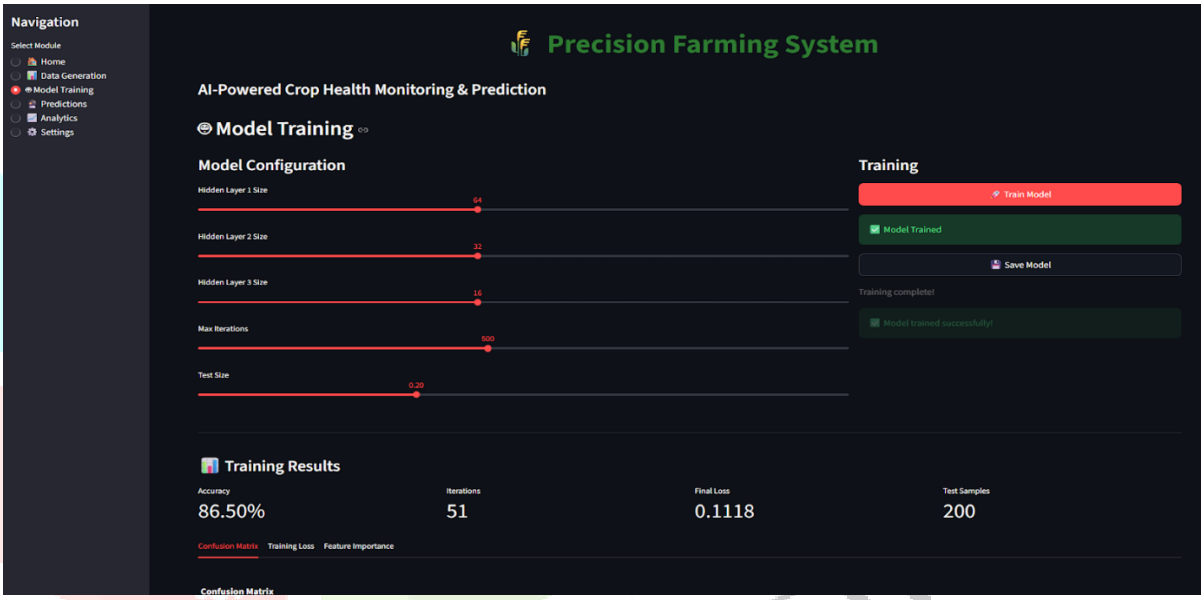


Fig. 2. Model Training Interface with Hyperparameter Configuration and Training Progress

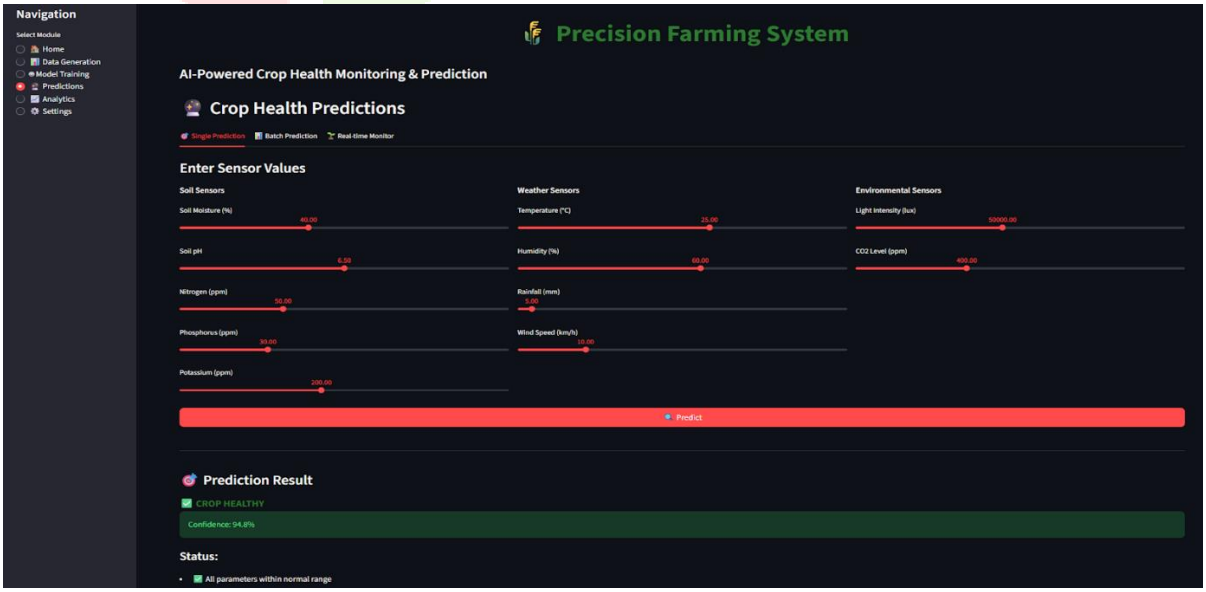


Fig. 3. Statistical Summary of Sensor Data in Data Analytics Module

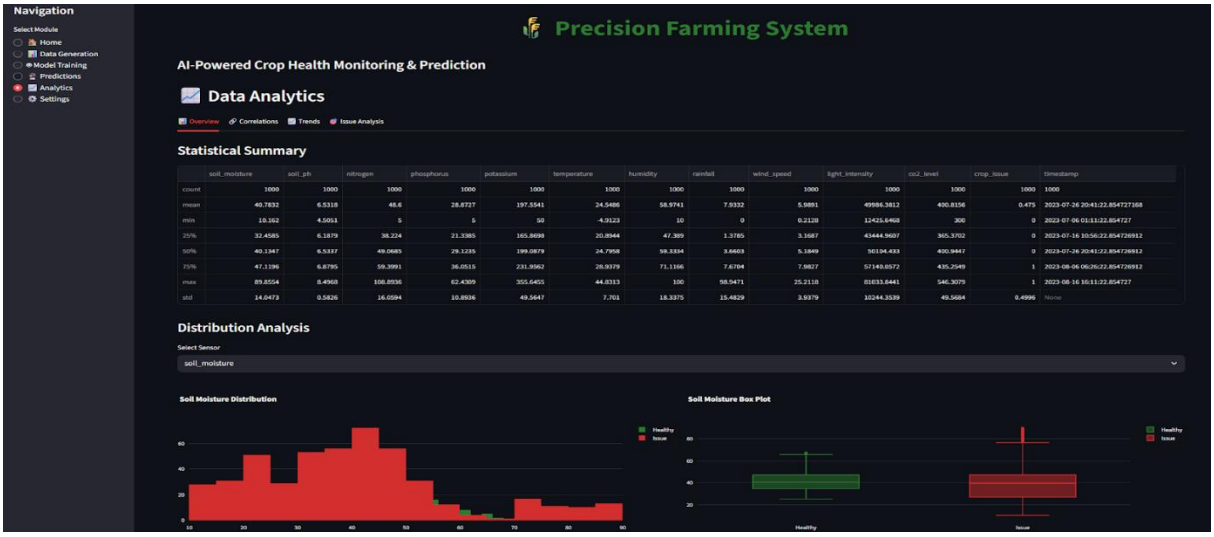


Fig. 4. Distribution Analysis using Histogram and Box Plot for Sensor Features

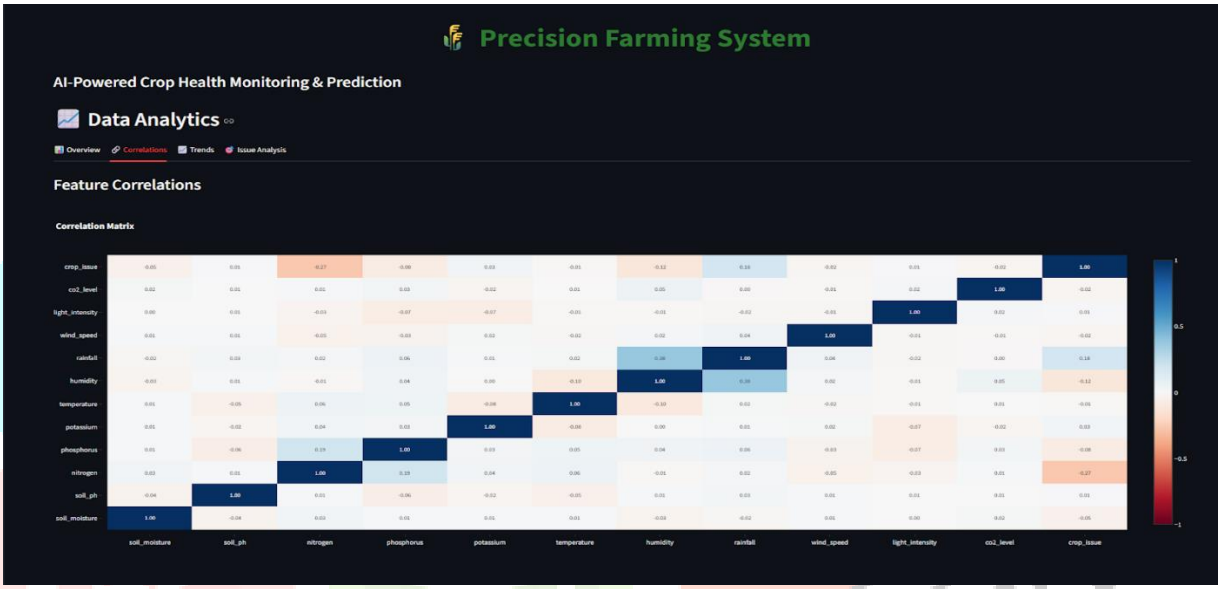


Fig. 5. Feature Correlation Heatmap for Crop Health Parameters

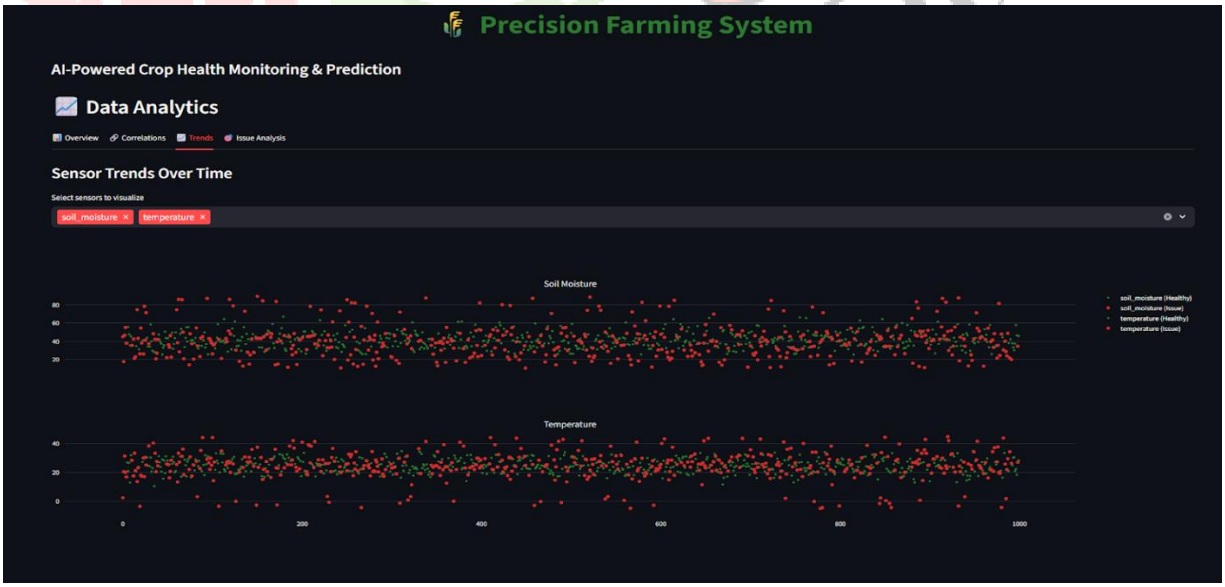


Fig. 6. Sensor Trends Over Time Visualization

Fig. 7. Issue Pattern Analysis using Pie Chart and Bar Graph

Limitations

There are several limitations associated with this study that should be acknowledged:

1) Dataset Specificity: The results of this study are based on a single dataset and may not generalize to all crops, geographic regions, or data sources. Different crops have distinct physiological responses to environmental factors, and the relative importance of features may shift accordingly. Therefore, validation on crop-specific and region-specific datasets is needed in order to confirm the generalizability of our findings.

2) Temporal Independence Assumption: Our MLP treats each sample independently and does not explicitly model temporal dependencies between consecutive growing seasons. It is important to note that the yield in one season may depend on the previous season's management (for example, soil depletion or crop rotation effects). Therefore, sequential architectures such as LSTM or Transformer models may be better suited to capture these long-range temporal dependencies.

3) Interpretability Gap: While permutation importance provides feature-level insight, the MLP remains less interpretable than decision tree-based methods at the individual prediction level. It is likely that farmers may prefer models where they can trace the reasoning behind a specific yield prediction, which decision trees naturally provide.

4) Multi-modal Data Limitation: Our study focuses exclusively on tabular data. The integration of satellite imagery (such as NDVI time series), drone-based multispectral data, and real-time sensor streams could enhance prediction accuracy through complementary information sources. However, such integration would require more complex architectures that go beyond the standard MLP.

5) Static Feature Assumption: Weather and management features are treated as static, season-level aggregates. In practice, however, the temporal distribution of rainfall (for example, early vs. late season) and temperature extremes within a season can significantly impact yield. Time-resolved feature representations could capture these intra-season dynamics more effectively.

6) Limited Extreme Event Coverage: As was identified in the failure case analysis, the model performance degrades for extreme conditions that are underrepresented in the training data. Climate change is expected to increase the frequency of such extreme events, which could potentially reduce model reliability over

V. CONCLUSION

The study demonstrates that a well-optimized Multilayer Perceptron model can achieve competitive performance for crop yield prediction using tabular agricultural data. Although Random Forest slightly outperforms MLP, the difference is not statistically significant. The results highlight the importance of feature engineering, regularization, and non-linear modeling. Due to its efficiency and scalability, MLP is a practical and effective solution for real-world precision agriculture applications.

Acknowledgment

The authors would like to express their sincere gratitude to our project guide and faculty members of the Department of Artificial Intelligence and Data Science at AISSMS IOIT, Pune for their continuous guidance, valuable suggestions, and support throughout the development of this project. We also thank our institution for providing the necessary resources and infrastructure to carry out this research on mango disease detection using deep learning models. Their encouragement and assistance played a vital role in the successful completion of this work.

REFERENCES

[1] Food and Agriculture Organization of the United Nations, "The future of food and agriculture: Trends and challenges," FAO, Rome, Italy, 2017.

[2] R. Gebbers and V. I. Adamchuk, "Precision agriculture and food security," Science, vol. 327, no. 5967, pp. 828–831, Feb. 2010.

[3] K. G. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," Sensors, vol. 18, no. 8, p. 2674, Aug. 2018.

[4] Anonymous Authors, "AI and robotics in agriculture: A systematic and quantitative review (2015–2025)," AI, vol. 5, no. 5, p. 75, 2025.

[5] M. Shoaib, A. Sadeghi-Niaraki, F. Ali, I. Hussain, and S. Khalid, "Leveraging deep learning for plant disease and pest detection: A comprehensive review and future directions," Frontiers in Plant Science, Feb. 2025.

[6] S. Murugavalli and R. Gopi, "Plant leaf disease detection using vision transformers for precision agriculture," Scientific Reports, vol. 15, Jul. 2025.

[7] F. Lin, K. Guillot, S. Crawford, Y. Zhang, X. Yuan, and N.-F. Tzeng, "An open and large-scale dataset for multi-modal climate change-aware crop yield predictions," in Proc. 30th ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD '24), Aug. 2024.

[8] Y. Yan, Y. Wang, J. Li, J. Zhang, and X. Mo, "Crop yield time-series data prediction based on multiple hybrid machine learning models," arXiv preprint arXiv:2502.10405, Jan. 2025.

[9] R. N. V. J. Mohan, P. S. Rayanoothala, and R. Praneetha Sree, "Next-gen agriculture: Integrating AI and XAI for precision crop yield predictions," Frontiers in Plant Science, vol. 15, Jan. 2025

[10] A. Ruiz-Gonzalez, "Multiplexed quantification of soil nutrients using an AI-enhanced and low-cost impedimetric sensor," Engineering Proceedings, vol. 106, no. 1, p. 7, Sep. 2025.