INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

ORIGINAL ARTICLE

Sensor Malfunction Simulation and Data Imputation using Deep Learning in Precision Agriculture

S. Aasha[1]*, **R. Sugumar**[2]

1 Full Time Research Scholar, PG & Research Department of Computer Science, Christhu Raj College, Affiliated to Bharathidasan University, Tiruchirappalli-620012, Tamil Nadu, India
2 Professor & Director, PG & Research Department of Computer Science, Christhu Raj College, Affiliated to Bharathidasan University, Tiruchirappalli-620012, Tamil Nadu, India

OPEN ACCESS

Received: 24/04/2025
Accepted: 15/06/2025
Published: 29/06/2025

Citation: Aasha S, Sugumar R (2025) Sensor Malfunction Simulation and Data Imputation using Deep Learning in Precision Agriculture. Indian Journal of Science and Technology 18(25): 1985-1997. https://doi.org/10.17485/IJST/v18i25.779

* **Corresponding author**.

s.aashaphd@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: This research initiative develops a new GAN-based technique for handling sensor malfunction-induced data gaps in precision agriculture data through robust precision improvement of machine learning applications. **Methods:** A GAN-based imputation method operates to recover missing data points from agricultural datasets. The analysis of multiple sensor parameters created missing data by using conditional probability rules. The GAN received training through a combination of authentic data and simulated information to make its value predictions and imputation capabilities. **Findings:** The GAN-based imputation process proved better than standard methods since it achieved superior results in measurement accuracy and precision together with higher recall and F1-score. The proposed technique reduced errors related to missing data more significantly than the standard methods imputation. Traditional imputation techniques proved ineffective in datasets with sensor malfunctions, but the model achieved better outcomes in these situations. The experimental findings established that GAN-based imputation presents potential worth for real-time agricultural data processing because it helps produce reliable predictions which benefit precision farming operations. **Novelty:** GAN-based imputation method for agricultural IoT systems becomes a proposed solution which processes non-random missing data along with sensor malfunctions in an efficient light-weight system.

Keywords: IoT; Precision Agriculture; Deep Learning; Machine Learning; Data Imputation; Sensors

1 Introduction

IoT devices enable real-time crop monitoring by collecting sensor data using precision agriculture methods[(1)],[(2)]. The systems need sensors which must measure primary environmental parameters that consist of soil moisture and temperature measurements together with humidity factors[(3)],[(4)],[(5)]. As a result, sensor malfunctions occur too often which causes both data loss and contamination which negatively affects the decision-making process.

The decision-making capacity of such systems gets compromised because sensor malfunction creates data accuracy issues[6]. Zou et al. (2023)[7] investigates Ag-IoT (Agriculture – IoT) sensor fault diagnosis techniques because dependable sensor systems remain essential for smart agriculture platforms. This research described machine learning-based diagnostics for large agricultural system sensor failures as well as economical solutions to improve sensor dependability.

Sami (2022)[8] developed Long Short-Term Memory (LSTM) for developing methods to boost sensor reliability in smart irrigation systems that predicted sensor data in real-time during external interference events. Bawankule et al. (2024)[9] monitored agricultural fields through real-time data acquisition for decision support. The simulation validity emerged from the research findings, yet the article acknowledged the technological limitations of simulating complex real-world environments.

Dabrowski and Rahman (2020)[10] developed a sequence-to-sequence forecasting technique that combines forward and backward RNNs for IoT sensor data imputation which yielded a 12% better result than traditional approaches. Khan et al. (2022)[11] demonstrated GANs' capability to create synthetic data which enhances imputation accuracy in mixed datasets through their study. The implementation of new techniques has solved several issues, but researchers still confront both complicated data structures and big training data requirements.

A Domain-specific Rule implementation of MissForest (DRMF) presents a solution for imputing missing IoT agricultural data[12]. The algorithm displayed better results than mean imputation, kNN and MissForest in dealing with datasets whose information was 10% incomplete. The system retains limitations due to its computational complexity which increases moderately to high when processing extensive systems. Veerasamy et al. (2023)[13] presented a farming solution which integrates uncertainty expert systems with paraconsistent logic and butterfly optimization to analyze inconsistent and vague data for crop recommendations. High resource requirements impeded the practical use of a system that enhanced forecast accuracy.

The LSSDEL framework focused on enhancing IoT-based crop recommendation systems by using selective model stacking and regularization methods[14]. The framework offered high predictive accuracy with reduced computational complexity, providing a lightweight yet effective solution for real-time applications. Sajindra et al.[15] introduced a deep learning model for predicting soil nutrients, which aids in optimizing crop cultivation. Expanding on this, Abekoon et al.[16] used explainable AI techniques like SHAP and LIME to enhance the interpretability of predictions in agriculture. In another study,[17] applied machine learning to improve soil nutrient prediction accuracy. Moreover,[18] focused on leveraging machine learning models for optimizing precision agriculture, contributing to better management of agricultural systems. Together, these works highlight the importance of machine learning and explainable AI in advancing agricultural practices.

The development of machine learning together with deep learning techniques demonstrates potential for solving this issue. Data imputation approaches from traditional times cannot manage sensor failure complexity effectively so researchers need to develop better robust solutions.

A Generative Adversarial Network (GAN) enables this work to introduce deep learning simulations that both model sensor faults and impute absent data points. The model works to boost both accuracy and dependability of predictions for crop yields within IoT-based precision agricultural systems. The proposed research work focuses on three fundamental objectives which include generating authentic data gaps and applying graph theory to detect patterns and developing a deep learning system for data completion.

2 Methodology

The data imputation methodology for IoT-based systems includes multiple operations starting with sensor-derived data collection. An advanced data imputation technique such as GAN-based imputation fills in missing data in collected sensor data to improve its quality for analysis purposes. IoT gateway receives data containing humidity and temperature and atmospheric pressure measurement data as illustrated in Figure 1. The dataset integrity remains unaffected by missing values in Sensor 1 because the imputation mechanism preserves data completeness. The application server receives the imputed data following the process so that the system functions optimally without missing resources.

2.1 Data Missingness Simulation

The occurrence of data missingness remains inevitable in precision agriculture IoT systems because of multiple causes including sensor malfunction, environmental conditions and transmission system breakdowns. The simulation of such missing data conditions requires perfect representation of real-world scenarios for developing precise imputation models. The proposed methodology details how to simulate data missingness through environmental factors by applying conditional probability rules based on structured missingness patterns.

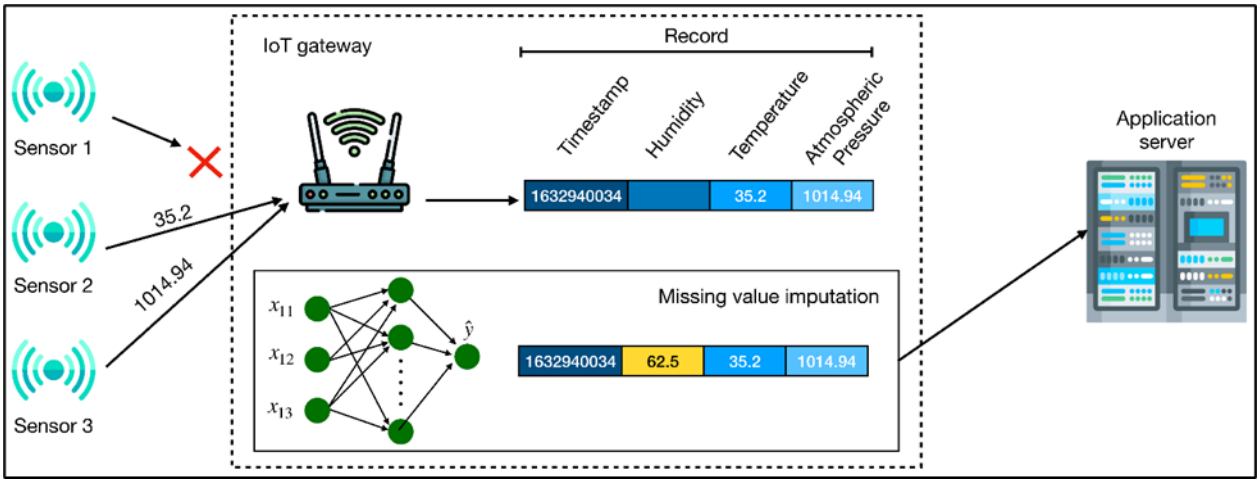

Fig 1. Workflow of the Proposed Work

2.1.1. *Modeling Missingness Based on Environmental Conditions*

The probability that data becomes missing depends on environmental factors which affect sensor performance. Data from sensors becomes either missing or inaccurate because of environmental factors including heavy rain and high winds together with sensor degradation throughout time. Standard statistical methodology enables the evaluation of environmental influence on missing data by using conditional probability to measure missing data risks under different environmental conditions.

The environmental vector consists of elements E_1, E_2, and E_n which represent distinct environmental factors influencing the sensor data including temperature, humidity, and rainfall. The sensor reading data $X = \{X_1, X_2, \ldots, X_m\}$ represents environmental measurements through the sequence of values X_i at specific times.

We calculate the conditional probability $P(M|E_i)$ that defines data missing as event M while using E_i as the environmental factor affecting sensors. The probability model of data absence takes the following form of **equation 1**:

$$P(M|E_i) = \frac{P(M \cap E_i)}{P(E_i)} \tag{1}$$

The assessment combines joint probabilities of missing data together with specific environmental factor E_i while accounting for environmental factor occurrence probabilities $P(M \cap E_i)$ and $P(E_i)$. A model can be developed to monitor missing data occurrences by adjusting probabilities according to environmental variations using this framework.

2.1.2. *Introducing Missing Data*

Moving forward to introducing missing data requires application of the established conditional probabilities of missing data. After applying evaluation probabilities to the authentic dataset, it generates artificial gaps which follow actual sensor malfunction patterns. The observed sensor reading value as X_i while $\widetilde{X_i}$ represents the same value after missingness is introduced. Here the missing data structure by setting $\widetilde{X_i}$ to NaN under the condition $P(M|E_i)$. The value NaN shows that the data point has been omitted from the dataset. The process of simulating the dataset with introduced missing values depending on environmental conditions is implemented by repeating this method across all X_i sensor readings.

$$\widetilde{X_i} = \begin{cases} NaN \ \ with\ probability\ P(M|E_i) \\ X_i \ \ with\ probability\ 1 - P(M|E_i) \end{cases} \tag{2}$$

Here, NaN indicates that the data point is missing. By iterating this process for each sensor reading X_i, it simulates a dataset where missing values are introduced based on the underlying environmental conditions.

2.1.3. *Patterns of Missing Data*

Actual missing data records in agricultural IoT systems show non-random characteristics. The patterns show that sensor reading missingness depends on other sensor reading values instead of occurring randomly. A graph-theoretical representation models such dependencies between sensors by treating each sensor as a node while edges show sensor relationship.

The graph is represented by $G = (V, E)$ where V consists of sensor nodes and E shows sensor connections using interdependent readings. The pattern of missing data follows the graph structure which enables information loss in one sensor to change values in related sensors according to the dependencies mapped in the graph. The graph node $v_i \in V$ allows calculation of data missing probability at v_i under the condition of its neighboring nodes $\mathcal{N}(v_i)$ as in the **equation 3**:

$$P\left(M_{v_i} \middle| \mathcal{N}(v_i)\right) = f\left(P\left(M_{v_j} \middle| E_j\right), \forall v_j \in \mathcal{N}(v_i)\right) \tag{3}$$

The function $f(\cdot)$ depicts the sensor-neighboring connection dependency through modeling of conditional probabilities $P\left(M_{v_j} \middle| E_j\right)$ from neighbor sensors. The system implements functions ranging from simple linear models to non-linear systems which extend to elaborate dependencies according to sensor relationships.

2.1.4. Rate of Missingness and Simulation Parameters

The controlled rate of missingness in simulated data represents realistic scenarios of data loss. The environmental factors together with sensor operating parameters establish the degree of missing data occurrence. A controlled rate of missingness prevents overly biased datasets that subsequently would affect model training. The ratio of missing observations which introduces data absence in the dataset is r_{miss}:

$$r_{\text{miss}} = \frac{\sum_{i=1}^{n} \mathbb{1}\left(\widetilde{X_i} = \text{NaN}\right)}{n} \tag{4}$$

The indicator function $\mathbb{1}\left(\widetilde{X_i} = \text{NaN}\right)$ returns a value of 1 when sensor readings are found missing while the total observations in the dataset are denoted by n.

Different missing data situations can be simulated through adjustments made to the rate r_{miss} given environmental conditions alongside sensor characteristics that produce realistic and useful data for future imputation and prediction functions.

2.2 Missing Data Pattern Analysis

To build efficient imputation models one needs to establish comprehension about how missing data structures appear. The missing data patterns in IoT-based precision agriculture systems result from multiple factors involving sensor dependencies together with environmental conditions rather than showing random behavior. The identification and analysis of such patterns demonstrates fundamental importance for developing better imputation models and protecting future prediction accuracy.

2.2.1. Graph-Theoretical Framework for Missingness Patterns

Precision agriculture devices operate jointly as interdependent systems because the non-availability of data in one sensor causes an impact on others. The sensor network receives graphical representation where nodes represent sensors while edges show the sensor relationships. Environmental factors simultaneously affecting two sensors together with intrinsic agricultural system behavior led sensors to display comparable behavior patterns.

The graph $G = (V, E)$ contains a sensor set $V = v_1, v_2, \ldots v_m$ and edge set $E = e_{ij}$ connecting v_i to v_j. Object $e_{ij} \in E$ represents the dependency between two sensors v_i and v_j while its weight w_{ij} indicates their relationship strength.

A mathematical description of sensor data missingness depends on relationships which exist between the sensors. The missingness event M_i of sensor v_i can be defined according to the **equation 5**:

$$M_i = \begin{cases} 1, & \text{if data from } v_i \text{ is missing} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The missingness pattern that occurs in each sensor depends directly on the missingness patterns appearing in all neighboring sensors of the graph.

$$P\left(M_i | \mathcal{N}(v_i)\right) = f\left(M_j, \forall j \in \mathcal{N}(v_i), w_{ij}\right) \tag{6}$$

In this framework $\mathcal{N}(v_i)$ becomes the neighbor set for sensor v_i and w_{ij} shows the strength between v_i and v_j when they are connected. The function f enables the representation of dependencies linking the missingness patterns across neighboring sensor units because this mechanism supports missing data propagation throughout the network. The graph-based formulation develops a mathematical system for assessing how missing data in one sensor element affects missing data in connected sensors to represent the structured aspect of missing data.

2.2.2. Dependency Matrix and Missingness Correlation

A dependency matrix D serves to visualize the correlation level between each two sensors in the system. The dependency matrix D receives the following definition:

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1m} \\ d_{21} & d_{22} & \cdots & d_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mm} \end{bmatrix}$$

The dependency matrix D contains the correlation values d_{ij} which are calculated from the statistical measure Pearson's correlation coefficient for sensor pairs v_i and v_j.

$$d_{ij} = \frac{\mathrm{Cov}\left(X_i, X_j\right)}{\sigma_{X_i}\sigma_{X_j}} \tag{7}$$

The universal mathematical expression consists of $\mathrm{Cov}\left(X_i, X_j\right)$ that calculates covariance from sensor pair v_i and v_j and the standard deviations for X_i and X_j which are expressed as σ_{X_i} and σ_{X_j}.

Determining patterns of missingness in datasets depends heavily on the dependency matrix D. The strength of connection between sensors indicates that notations in a linked sensor could represent absent data in other nearby sensors. When such dependencies exist the imputation model makes use of these relationships to improve its accuracy when restoring missing values.

2.2.3. Conditional Missingness and Dependency Propagation

The previous studies show that missing measurements in one sensor depend not only on linked sensor values but also on complex relationships that sensors experience under certain conditions and system requirements[12]. In the proposed work, the updated missingness model measures how missing data in one location depends on missing levels in nearby sensors. The missingness of sensor v_i becomes M_i while direct neighbors of v_i go by $\mathcal{N}(v_i)$.

$$P\left(M_i|\mathcal{N}(v_i)\right) = \prod_{j\in\mathcal{N}(v_i)} P\left(M_i|M_j\right) \tag{8}$$

The above equation describes that v_i missingness depends not only on direct neighbor sensors but receives influence from missingness spreading across the entire network. The expression $P\left(M_i|M_j\right)$ shows how likely observations at sensor position i become missing when sensor j stops working. This dependence reflects the amount of correlation between the two sensors. To review complex patterns of dependencies we need to add new graph links that demonstrate more extensive connections. An algorithm can determine $k-th$ order neighbors of sensor v_i through $\mathcal{N}^k(v_i)$.

$$P\left(M_i|\mathcal{N}^k(v_i)\right) = \prod_{j\in\mathcal{N}^k(v_i)} P\left(M_i|M_j\right) \tag{9}$$

2.2.4. Statistical Testing of Missingness Patterns

To validate the model, it performs tests that check if missing data from multiple sensors appears independently or aligns with our described dependency graph. To check if sensor values v_i and v_j show a separated missingness pattern the Chi-square test statistic χ^2 performs evaluation.

$$\chi^2 = \sum_{i,j} \frac{\left(O_{ij} - E_{ij}\right)^2}{E_{ij}} \tag{10}$$

The statistics show the difference between observed and expected counts of missingness patterns between sensors v_i and v_j. Significant dependence between sensors within the graph-based model proves its correct representation of missingness behavior.

2.3 Imputation Model Development

The task of data imputation is to restore missing values in a dataset by estimating them based on observed data, while preserving the integrity of the dataset. This section develops a GAN for the imputation of missing sensor data, where the goal is to generate plausible sensor values for missing data points while maintaining consistency with the observed data. Figure 2 presents the workflow of the proposed imputation model.

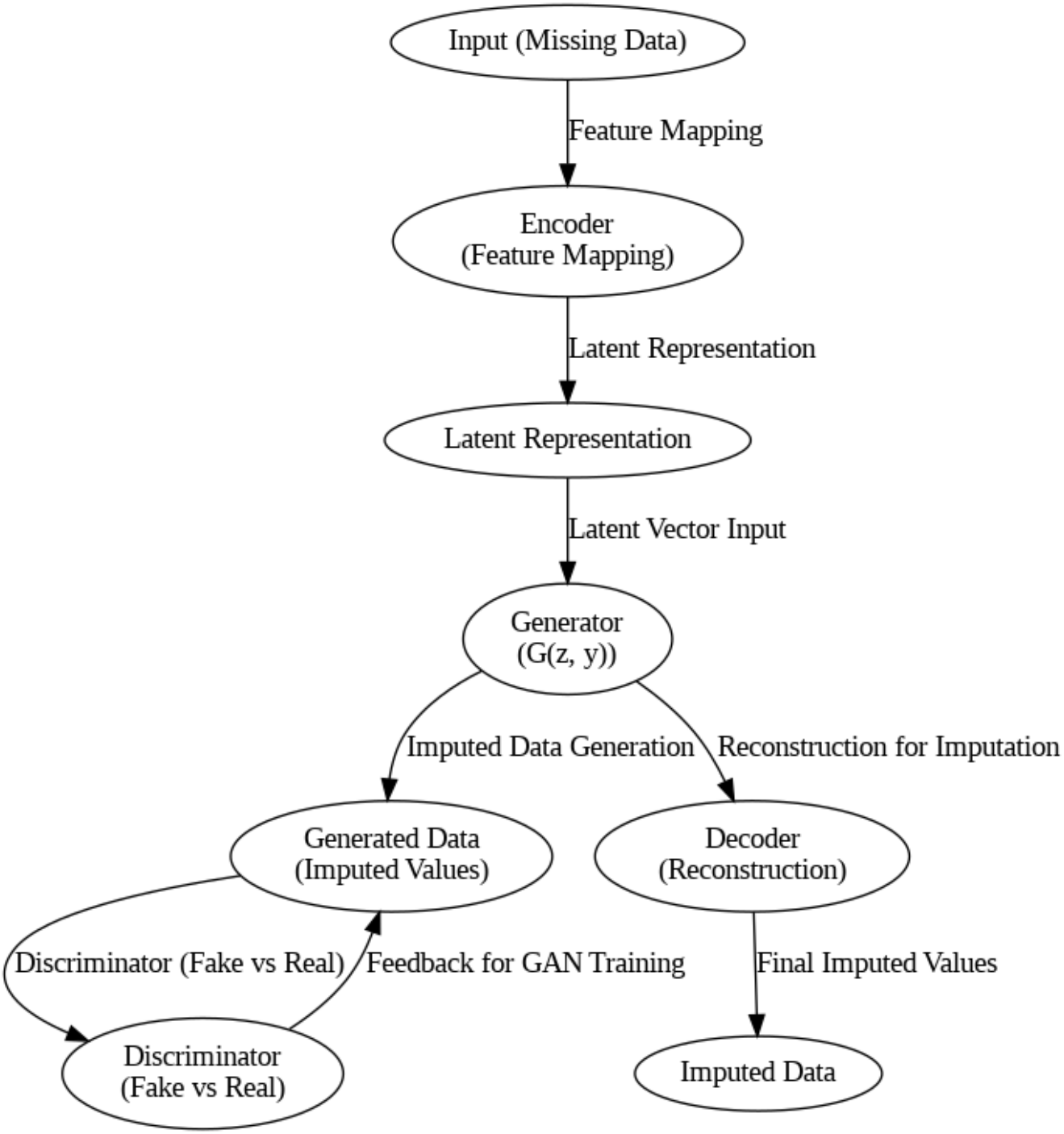

Fig 2. Workflow of proposed imputation model

2.3.1. GAN Architecture for Imputation

A GAN consists of two neural networks: a generator and a discriminator, which are trained simultaneously in an adversarial manner. The generator aims to produce data (in this case, imputed sensor readings) that closely resembles real data, while the discriminator attempts to distinguish between the imputed data and real data. The training process involves the generator trying to fool the discriminator into classifying imputed data as real, and the discriminator trying to correctly identify whether the data is real or imputed.

Let X_{obs} represent the observed sensor data, and X_{miss} represent the missing data. The imputation task is to estimate the missing values in X_{miss} by generating plausible data points X_{gen} such as:

$$X_{imputed} = X_{obs} \cup X_{gen} \tag{11}$$

where X_{gen} is the data generated that replaces the missing values in X_{miss}.

Generator G takes the input data, which consists of the observed values and the missingness pattern and generates imputed values. The generator's objective is to minimize the following loss function:

$$\mathcal{L}_{\mathcal{G}} = E_{X \sim p_{data}} [\log (1 - D(G(X)))] \tag{12}$$

where p_{data} is the true distribution of the data, $G(X)$ is the generated data (i.e., imputed values), and $D(\cdot)$ is the discriminator's output, which represents the probability that the data is real (i.e., observed data). The generator's loss function encourages it to produce imputed values that are indistinguishable from the real data.

Discriminator D is a binary classifier that distinguishes between real data and generated data. It is trained to maximize the likelihood of correctly classifying the observed data as real and the imputed data as fake. The discriminator's objective is given by:

$$\mathcal{L}_{\mathcal{D}} = -E_{X \sim p_{data}} [\log D(X)] - E_{X \sim p_{gen}} [\log (1 - D(X))] \tag{13}$$

where p_{gen} is the distribution of the generated data, and $D(X)$ is the discriminator's prediction for whether X is real or fake. The discriminator's loss function encourages it to correctly classify real and imputed data.

2.3.2. Objective Function and Adversarial Training

The overall objective of training a GAN is to optimize both the generator and the discriminator simultaneously. The generator minimizes its loss function $\mathcal{L}_{\mathcal{G}}$, while the discriminator minimizes its loss function $\mathcal{L}_{\mathcal{D}}$. This creates an adversarial process where the generator strives to produce data that the discriminator cannot distinguish from the real data. The combined objective function for the GAN is:

$$\mathcal{L}_{\mathcal{GAN}} = \mathcal{L}_{\mathcal{G}} + \mathcal{L}_{\mathcal{D}} \tag{14}$$

The adversarial training process leads to the generator learning how to produce realistic sensor readings that accurately impute missing values in X_{miss}.

2.3.3. Conditional GAN for Missing Data Imputation

For missing data imputation, a Conditional GAN (cGAN) is used, with the missingness pattern provided as additional input to both the generator and discriminator. This allows the model to condition the imputation on the known structure of missingness, which is essential in precision agriculture, where the pattern of missing data is often non-random and dependent on environmental conditions.

Let M represent the missingness mask, where each element $M_{ij} \in \{0, 1\}$ indicates whether the data at position (i, j) is missing $(M_{ij} = 1)$ or observed $(M_{ij} = 0)$. Generator G takes both the observed data X_{obs} and the missingness pattern M as inputs and generates the imputed data X_{gen}. The generator's objective in the cGAN setting is given in **equation 15**:

$$\mathcal{L}_{\mathcal{G}} = E_{X \sim p_{data}} [\log (1 - D(G(X, M)))] \tag{15}$$

where the generator is conditioned on both the observed data and the missingness mask M. The discriminator also receives the missingness mask as input, and its loss function becomes:

$$\mathcal{L}_{\mathcal{D}} = -E_{X \sim p_{data}} [\log D(X, M)] - E_{X \sim p_{gen}} [\log (1 - D(G(X, M)))] \tag{16}$$

The discriminator distinguishes between real and generated data, while also considering the missingness pattern. The generator learns to produce imputed values that are not only realistic but also consistent with the missing data structure.

2.3.4. Regularization and Convergence

To ensure that the imputation model converges efficiently and avoids overfitting, regularization techniques such as Batch Normalization and dropout are applied. Batch normalization helps stabilize the training process by normalizing the activations in each layer of the network, thereby accelerating convergence. Dropout is used to prevent overfitting by randomly setting some of the weights to zero during training, forcing the model to generalize better. The overall regularization term can be expressed as in **equation 17**:

$$\mathcal{L}_{reg} = \lambda_1 \cdot \mathcal{L}_{batchnorm} + \lambda_2 \cdot \mathcal{L}_{dropout} \tag{17}$$

where λ_1 and λ_2 are hyperparameters that control the strength of the regularization terms. The total loss function for training the generator is then:

$$\mathcal{L}_{total} = \mathcal{L}_{\mathcal{G}} + \mathcal{L}_{reg} \tag{18}$$

2.4 Dataset Description

This research uses Smart Farming 2024 (SF24)[19] dataset that spans numerous IoT-based system features found in agricultural applications. The integrated set of parameters includes environmental components and soil characteristics together with essential crop data points which aid both crop management efforts and yield prediction needs. The dataset holds 2,200 records that include twenty features that concentrate on environmental monitoring and agricultural management.

Several environmental factors including temperature and humidity, and soil moisture get registered through IoT sensors that monitor the farming area. Real-time data provided by these sensors helps administrators make efficient decisions throughout crop management operations. The dataset contains supplementary factors like recorded wind speed entries in association with tracked CO_2 levels and quantified solar exposure measurements that boost the dynamic approach to farm environment monitoring. Farmers use features including pest pressure together with fertilizer usage and irrigation frequency to both monitor resource usage and implement sustainable agricultural practices.

The detailed summary from Table 1 demonstrates the IoT features within the dataset by showing their essential parameters that support precision farming activities. The data points function as essential tools for studying crop environmental conditions because they help farmers make precise decisions leading to better agricultural practices.

Table 1. Sensors Dataset

Sensors	N, P (Phosphorus), K (Potassium), Temperature, Humidity, pH, Rainfall, Soil Moisture, Sunlight Exposure, Wind Speed, CO2 Concentration, Irrigation Frequency, Pest Pressure, Frost Risk
Records	2200
Labels	22
Features	23

3 Results and Discussion

3.1 Experimental Setup

Table 2 shows choices based on GAN training practices and the imputation model's requirements to ensure accuracy, stability, and generalization to real-world agricultural data. The hyper parameters used for the proposed GAN-based imputation model are appropriately chosen to achieve the best performance, but at the same time avoid overfitting, and good convergence. The learning rate of 0.001 is selected as some form of balance between the speed of training and stability. A lower learning rate helps to avoid the model from overshoot and leads to more stable convergence, especially with deep learning models that need a lot of tuning for iterations. The batch size of 32 is a compromise between the requirement to memory efficiency and stability of gradient estimation. This value takes care that the model is trained without wasting a lot of computational resources. Smaller batch sizes allow us to have more frequent updates, but it may carry more noise in the gradients estimates and larger batch sizes might slow the update and cause memory inefficiency. A dropout rate of 0.5 is one of the regularization techniques used to prevent overfitting. Dropout serves to decrease reliance on some specific nodes to the model while training as it disables neurons at random. This leads the network to learn stronger features and generalize better on unseen data.

For weight decay (L2 regularization), 1e-5 is the chosen parameter to penalize excessive weights to stop the overfitting. Regularization makes sure that the model doesn't become complex and memorizes the data used for training, which may unfavorably influence its generalization capability. The Adam optimizer parameters are typical selections that support the optimizer, sustain the momentum and accommodate the gradients' second moments. These values help efficient training because it provides an adaptive learning rate that updates throughout the process of optimization and thus enhances the convergence of the model. The model is trained for 100 epochs, with the aim of proper exposure to the data without overfitting it. A moderate number of epochs both avoid under fitting and at the same time, prevent the model from overfitting the data. This is complemented by an early stopping patience of 10 that stops the training if the model is not improving the validation as compared to the 10 epochs. This is also helpful in terms of saving computation time and eliminating overfitting by reducing pointless relaxation of training further than the ideal of maximum generalization. Finally, the discriminator and generator loss weights are set to 1.0, which means that both components of the GAN are optimized in a balanced way. The two losses do play an equal part in this process of adversarial learning, whereby the generator utilizes the feedback from the discriminator efficiently.

Table 2. Hyperparameter Configuration

Parameter	Value/Range
Learning Rate	0.0001
Batch Size	32
Dropout Rate	0.5
Weight Decay (L2 Regularization)	1e-5
Adam Optimizer Parameters (Beta1, Beta2)	Beta1: 0.9, Beta2: 0.999
Epochs	100
Early Stopping Patience	10
Discriminator Loss Weight	1.0
Generator Loss Weight	1.0

3.2 Results

The outcome of the proposed imputation method needs evaluation to determine its success in solving missing data problems within precision agriculture systems. Figure 3 demonstrates how temperature alongside humidity alongside soil moisture stands out as major environmental and soil-based factors that influence crop yield predictions. The analysis features stem from simulating dataset missing patterns, as outlined in Section 2.1.

After implementing the proposed imputation method researchers used Table 3 to demonstrate performance evaluations of different machine learning algorithms. The Random Forest model achieved 92% accuracy alongside Gradient Boosting at 96% accuracy while XGBoost reached 97% accuracy according to the provided table. The proposed GAN-based imputation model enhances data integrity, so machine learning models achieve better prediction accuracy than basic imputation techniques.

Table 3. Results of Proposed Imputation Method with various ML models

Model	Precision	Recall	F1-Score	Accuracy
Naive Bayes	0.91	0.99	0.95	0.99
SVM	0.92	0.47	0.57	0.48
Logistic Regression	0.84	0.68	0.74	0.67
Random Forest	1	0.93	0.96	0.92
Gradient Boosting	0.99	0.96	0.97	0.96
MLP	0.92	0.78	0.84	0.77
XGBoost	0.99	0.97	0.98	0.97

Figure 4 compares traditional imputation methods (mean, median), genetic algorithm, previous work DMRF[12], and the proposed GAN-based method. GAN-based imputation performs better in precision, recall, F1-score, and accuracy. The GAN shows effectiveness in generating simulated missing data which helps XGBoost and Gradient Boosting models restore complex sensor malfunction patterns. Like SVM the GAN-based method performed better than other algorithms because it uses its ability to learn complex feature relations which enhances data imputation consistency to observed data patterns.

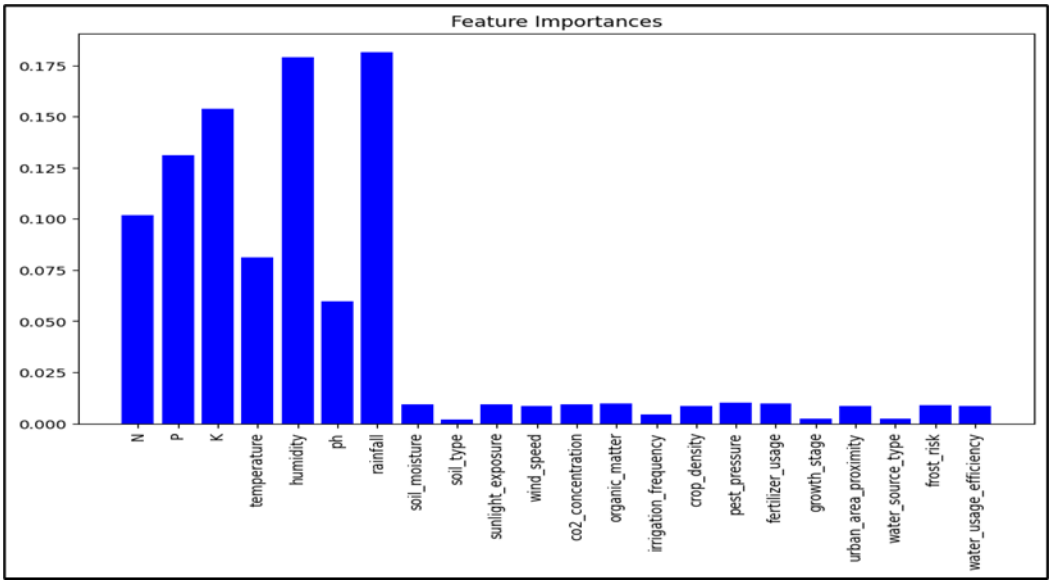

Fig 3. Feature importance of the collected data

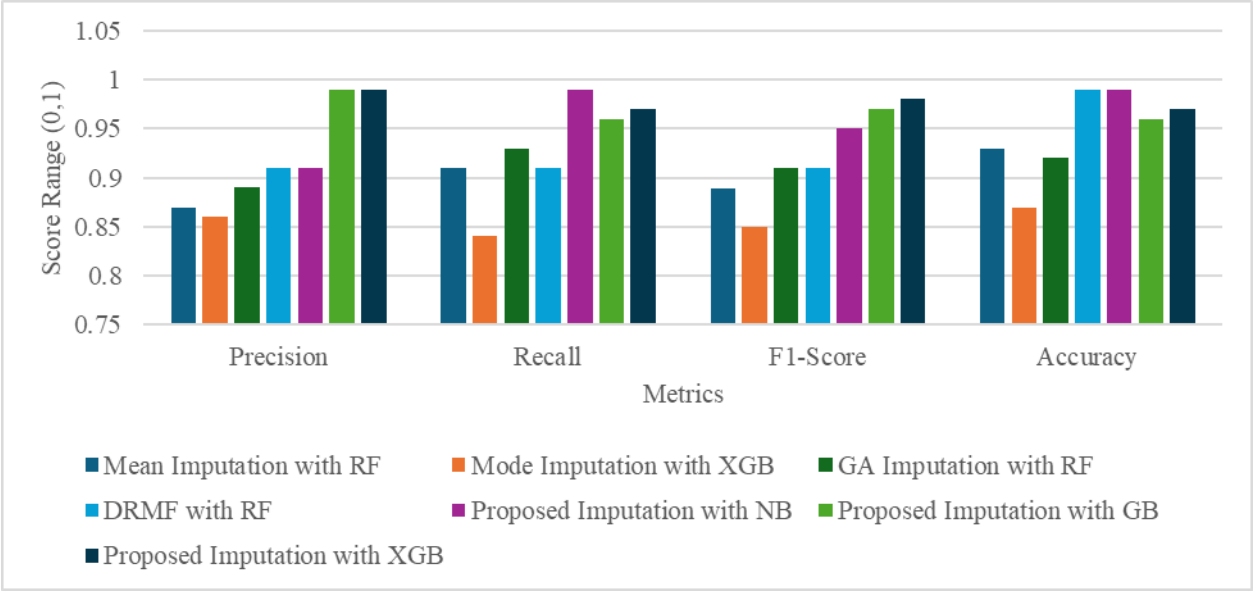

Fig 4. Comparative analysis of proposed work with previous works

Sensitivity analysis evaluated the robustness of the GAN-based imputation against 5%, 10%, and 20% missing data rates on key features in Table 4. Various machine learning classifiers were then trained on the imputed datasets to examine how predictive performance is affected by increasing amounts of missingness. The findings show that GAN-based imputation approach is effective in preserving high classification accuracy as the amount of missing data increases. Gradient Boosting and XGBoost produced the best precision and accuracy on all levels of missingness, with the values of precision over 0.98 and accuracy of approximately 99.7%. Logistic Regression and SVM reported relatively lower recall and F1-scores, particularly when the missingness level was high, indicating that the two methods were sensitive to the quality of imputation. The accuracy and F1-scores of all classifiers experienced minor reductions when the percentage of missing data varied between 5 and 20%, indicating that the imputation model has substantial potential to recover the integrity of data in precision agriculture settings. The GAN-based imputation method reliably maintains downstream model performance during sensor failures.

Table 4. Sensitivity Analysis of Proposed Imputation Method with Various Missingness Rate

Missingness (%)	Model	Precision	Recall (Sensitivity)	F1-Score	Accuracy
	Gradient Boosting	0.986	0.961	0.971	0.997
	Logistic Regression	0.839	0.680	0.737	0.981
	MLP	0.914	0.788	0.839	0.987
5	Naive Bayes	0.910	0.993	0.944	0.993
	Random Forest	0.996	0.926	0.955	0.996
	SVM	0.917	0.476	0.568	0.975
	XGBoost	0.987	0.959	0.971	0.997
	Gradient Boosting	0.988	0.958	0.970	0.997
	Logistic Regression	0.828	0.680	0.734	0.981
	MLP	0.910	0.781	0.836	0.987
10	Naive Bayes	0.914	0.994	0.947	0.994
	Random Forest	0.997	0.920	0.953	0.996
	SVM	0.916	0.473	0.566	0.975
	XGBoost	0.988	0.960	0.972	0.997
	Gradient Boosting	0.988	0.961	0.972	0.997
	Logistic Regression	0.835	0.684	0.740	0.982
	MLP	0.911	0.789	0.841	0.987
20	Naive Bayes	0.914	0.995	0.947	0.994
	Random Forest	0.996	0.927	0.956	0.996
	SVM	0.921	0.472	0.566	0.975
	XGBoost	0.988	0.955	0.969	0.997

3.3 Discussion

Precision agriculture benefits from the advanced sensor malfunction imputation capabilities of GANs because Logistic Regression and MLP together with their basic imputation techniques show inferior precision and recall. The analysis demonstrates how the proposed approach effectively handles raw data issues to enhance prediction power in IoT-based farming systems.

DRMF[12] effectiveness vanishes when the missing data exhibits systematic patterns or responds to environmental triggers in uncontrolled environments. The DRMF method demonstrates a significant weakness in precision agriculture because it fails to recognize that missing values tend to relate specifically to environmental conditions which this method does not address directly. The GAN-based imputation method learns dynamic patterns automatically while delivering more precise predictions which match the database dependencies.

The GAN-based imputation method outperformed additional advanced techniques like DRMF[12] for accurate data generation across various experimental settings particularly in agricultural data situations. The main drawback of GAN-based imputation occurs when it needs a large amount of training data because the model output might deteriorate using limited data or data with excess noise beyond GAN capabilities.

The main accomplishment of this research lies in developing a GAN-based imputation model which unites deep learning capabilities with IoT-based agricultural data processing. The GAN implements adversarial learning techniques to restore missing values while maintaining the original data patterns and distribution characteristics of the dataset instead of using traditional statistical techniques or basic machine learning models. The imputed data predicts more accurately due to this model improvement which works best for agricultural datasets that are complex and high-dimensional.

Both a generator and a discriminator form the model's architecture which produces more authentic imputed data than heuristic or predefined rule-based methods during imputation. The proposed research demonstrates that GAN models succeed as dependable solutions to handle missing data imputation in IoT-based crop recommendation systems.

The proposed work shows promising results but needs further improvement through future studies because of existing challenges. GANs face main obstacles because their training process becomes increasingly complicated when used with large-scale datasets. GAN models demand extensive processing capabilities along with ample memory storage which makes them impractical for quick applications operating in resource-limited systems such as small-scale agricultural establishments. Any

defects or biases within the training data affect GAN performance in a major way because these neural networks respond intensely to information quality input.

3.4 Ethical and Practical Considerations

The application of the proposed cGAN-based imputation algorithm in a real farmer's environment raises several ethical and practical issues that should be considered. Ethically, data privacy should be of the highest order since IoT sensor networks have a tendency of gathering sensitive data pertaining to farm activities, land utilization, and unique crop production mechanisms. It is important to ensure that the imputation model and the data handling procedures will be in line with the appropriate data protection regulations and will be highly confidential, to establish trust relationships with the stakeholder and farmers.

The GAN-based model requires significant computing resources. While accurate in the lab, its use in diverse farming conditions demands efficient computation solutions due to varying sensor densities and data volumes. Training and inference for this model requires high processing power and memory, posing challenges for integration into resource-constrained IoT gateways or edge devices in small- to medium-scale farming. Model compression, incremental training, or hybrid cloud-edge systems are essential for practical deployment.

It is important to be aware of any potential biases in the imputation procedure. The patterns learned by the GAN are derived from limited training data and may not comprehensively represent all environments or sensor dynamics. As a result, the imputed values might reinforce pre-existing biases or miss rare yet important sensor anomalies, thus making inferences incorrect. To minimize risks, it is essential to apply continuous model validation, integrate domain knowledge, and use adaptive updates for ethical imputation in precision agriculture.

4 Conclusion

The study demonstrated an innovative way to handle missing precision agriculture data through its cGAN-based model implementation. The imputation procedures that rely on cGAN model proved better than DRMF[12] according to results provided by the study. The proposed method demonstrated enhanced ability to handle sensor malfunction-generated non-random missing data that is frequently encountered in real-time agricultural implementation. The deployment of cGAN for agricultural IoT system missing data imputation represents the main distinguishing aspect of this research compared to typical random data imputation techniques that dominate current approaches. The cGAN model improves missing value restoration because it integrates information about environmental dependencies. The approach offers better functionality than DRMF because it does not recognize sensor data loss patterns effectively.

The implementation of hyperparameter tuning exists as a limitation because it demands time from experts who need to set the perfect model parameters. Future research should concentrate on GAN-based imputation automation because the model has proven effectiveness, but practitioners need improved access to its application. This research study generates important findings that impact the current progress of precision agriculture. Through its functionality as a GAN-based imputation model the proposed method demonstrates the capability to improve agricultural crop recommendation systems while enhancing farmer decision-making processes. Research in the field should direct efforts toward both simplifying GAN model computational effectiveness as well as implementing these models with other machine learning algorithms for better crop recommendation system scalability and performance.

References

1) Choudhary V, Guha P, Pau G, Mishra S. An overview of smart agriculture using internet of things (IoT) and web services. *Environmental and Sustainability Indicators*. 2025. https://doi.org/10.1016/j.indic.2025.100607.
2) Sharma K, Shivandu SK. Integrating artificial intelligence and internet of things (IoT) for enhanced crop monitoring and management in precision agriculture. *Sensors International*. 2024. https://doi.org/10.1016/j.sintl.2024.100292.
3) Woo-García RM, Pérez-Vista JM, Sánchez-Vidal A, Herrera-May AL, de-la Rosa EO, Caballero-Briones F, et al. Implementation of a wireless sensor network for environmental measurements. *Technologies*. 2024;12(3):41. https://doi.org/10.1016/j.sintl.2024.100292.
4) Irwanto F, Hasan U, Lays ES, Croix NJDL, Mukanyiligira D, Sibomana L, et al. IoT and fuzzy logic integration for improved substrate environment management in mushroom cultivation. *Smart Agricultural Technology*. 2024;7:100427. https://doi.org/10.1016/j.atech.2024.100427.
5) Morchid A, Jebabra R, Khalid HM, Alami RE, Qjidaa H, Jamil MO. IoT-based smart irrigation management system to enhance agricultural water security using embedded systems, telemetry data, and cloud computing. *Results in Engineering*. 2024;23:102829. https://doi.org/10.1016/j.rineng.2024.102829.
6) Wang Y. Deep Learning Methods Used in Precision Agriculture. vol. 142. EDP Sciences. 2024;p. 01004. https://doi.org/10.1051/bioconf/202414201004.
7) Zou X, Liu W, Huo Z, Wang S, Chen Z, Xin C, et al. Current status and prospects of research on sensor fault diagnosis of agricultural internet of things. *Sensors*. 2023;23(5):2528. https://doi.org/10.3390/s23052528.
8) Sami M, Khan SQ, Khurram M, Farooq MU, Anjum R, Aziz S, et al. A deep learning-based sensor modeling for smart irrigation system. *Agronomy*. 2022;12(1):212. https://doi.org/10.3390/agronomy12010212.

9) Bawankule G, Urwate P, Chavan K, Inamdar F, Deshpande S. Smart Precision Agriculture using IoT Simulation. *Int J Adv Res Sci Commun Technol*. 2024;p. 302–312. https://doi.org/10.48175/IJARSCT-18246.
10) Dabrowski JJ, Rahman A. Sequence-to-Sequence Imputation of Missing Sensor Data. *AI 2019: Advances in Artificial Intelligence*. 2019. https://doi.org/10.1007/978-3-030-35288-2_22.
11) Khan W, Zaki N, Ahmad A, Masud MM, Ali L, Ali N, et al. Mixed data imputation using generative adversarial networks. *IEEE Access*. 2022;10:124475–124490. https://doi.org/10.1109/ACCESS.2022.3218067.
12) Sindhu S, Arockiam L. DRMF: Optimizing machine learning accuracy in IoT crop recommendation with domain rules and MissForest imputation. *The Scientific Temper*. 2024;15(3):2570–2578. https://doi.org/10.58414/SCIENTIFICTEMPER.2024.15.3.24.
13) Veerasamy K, Fredrik EJT. Intelligent Farming based on Uncertainty Expert System with Butterfly Optimization Algorithm for Crop Recommendation. Infinite Study. 2023. https://doi.org/10.58346/JISIS.2023.I4.011.
14) Sindhu S, Arockiam L. A lightweight selective stacking framework for IoT crop recommendation. *The Scientific Temper*. 2024;15(4):3173–3181. https://doi.org/10.58414/SCIENTIFICTEMPER.2024.15.4.26.
15) Sajindra H, Abekoon T, Jayakody JADCA, Rathnayake U. A novel deep learning model to predict the soil nutrient levels (N, P, and K) in cabbage cultivation. *Smart Agricultural Technology*. 2024;7:100395. https://doi.org/10.1016/j.atech.2023.100395.
16) Abekoon T, Sajindra H, Buthpitiya BLSK, Rathnayake N, Meddage DPP, Rathnayake U. Justifying the prediction of major soil nutrients levels (N, P, and K) in cabbage cultivation. *MethodsX*. 2024;12:102793. https://doi.org/10.1016/j.mex.2024.102793.
17) Abekoon T, Sajindra H, Rathnayake N, Ekanayake IU, Jayakody A, Rathnayake U. A novel application with explainable machine learning (SHAP and LIME) to predict soil N, P, and K nutrient content in cabbage cultivation. *Smart Agricultural Technology*. 2025;11:100879. https://doi.org/10.1016/j.atech.2025.100879.
18) Kumar V, Sharma KV, Kedam N, Patel A, Kate TR, Rathnayake U. A comprehensive review on smart and sustainable agriculture using IoT technologies. *Smart Agricultural Technology*. 2024;p. 100487. https://doi.org/10.1016/j.atech.2024.100487.
19) Engineer D. Smart Farming Data 2024 (SF24). *Kaggle*. 2024. Available from: https://www.kaggle.com/datasets/datasetengineer/smart-farming-data-2024-sf24.