# Neural Network Based Smart Irrigation System with Edge Computing Control for Optimizing Water Use

Freddy Artadima Silaban, Ahmad Firdausi
Faculty of Engineering, Electrical Engineering Study Program, Universitas Mercu Buana, Jakarta, Indonesia

## ARTICLE INFORMATION

**Article History:**

Received 07 October 2024
Revised 27 November 2024
Published 10 December 2024

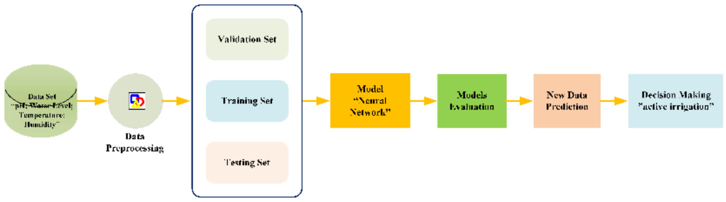
**Keywords:**

Neural Networks;
Irrigation;
Accuracy;
Edge Computing;
Crop

**Corresponding Author:**

Freddy Artadima Silaban
Faculty of Engineering,
Electrical Engineering Study Program, Mercu Buana University, Jakarta Indonesia
Email: freddy.artadima@mercubuana.ac.id

## ABSTRACT



Efficient irrigation is critical in agriculture, particularly in regions with erratic rainfall. As global water scarcity intensifies, optimizing irrigation processes is essential to ensure sustainable food production. This study proposes a novel smart irrigation system leveraging neural networks and edge computing to enhance water use efficiency and crop yields. The dataset comprises environmental variables, including pH, water level, temperature, and humidity, sourced from reputable open repositories. Preprocessing steps included handling anomalies, encoding categorical variables, and feature standardization. A neural network with optimized architecture was trained using 70% of the data, validated with 15%, and tested on the remaining 15%. The system achieved a testing accuracy of 91.33%, with precision, recall, F1-score, and AUC metrics exceeding industry benchmarks (AUC: Base = 0.99, Ideal = 0.97, Dry = 0.98). The model was deployed on an NVIDIA Jetson Nano using Docker, demonstrating real-time prediction capabilities with minimal latency. The smart irrigation system automates water pump operations based on soil conditions, providing practical benefits such as reduced water waste and improved crop health. With its adaptable design and scalability, this system represents a step forward in sustainable agriculture, contributing to global efforts to address food security challenges.

**Document Citation:**

F. A. Silaban and A. Firdausi, "Neural Network Based Smart Irrigation System with Edge Computing Control for Optimizing Water Use," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 6, no. 4, pp. 234-333, 2024, DOI: 10.12928/biste.v6i4.11965.

## 1. INTRODUCTION

Irrigation plays an important role in agriculture, especially in areas that experience uncertain or limited rainfall [1]. With the increase in the global population, the need for efficient and sustainable food production is becoming increasingly crucial, highlighting irrigation as a vital factor in maintaining global food security [2]. Amid the increasing challenges of water resource scarcity, it has become essential to develop more efficient and water-saving irrigation systems [3].

Traditional irrigation methods, such as manual pumps, require significant time and financial investment, especially on large agricultural lands. These methods often lead to water wastage and inefficiencies, which are no longer sustainable in the face of rising water scarcity. Plant health monitoring, as a component of efficient irrigation, is essential to achieving optimal crop potential. While several researchers have explored automatic or semi-automatic irrigation methods, these systems still face limitations, including high dependency on cloud-based processing, latency issues, and the need for reliable internet connectivity [4]–[6].

Technological advances, especially in artificial intelligence (AI) [7][8], have provided significant opportunities to increase efficiency, productivity, and sustainability in agriculture through the concept of smart farming [9]. AI-driven systems, supported by high-quality sensors and big data analysis [10][11], enable dynamic management of irrigation by analyzing environmental conditions such as temperature, humidity, soil moisture content, and weather conditions [12][13]. By collecting real-time data from sensors, these systems can identify patterns and trends, allowing for more informed agricultural decision-making. This capability is especially critical in addressing the dynamic and often unpredictable nature of agricultural environment

The application of AI, particularly neural networks, enables dynamic irrigation systems that automatically regulate water distribution based on real-time data analysis [14]. Neural networks are particularly well-suited for agricultural applications due to their ability to model complex, non-linear relationships between environmental variables and crop requirements. This research focuses on developing a smart irrigation system using neural networks [15][16], implemented on edge computing devices such as the NVIDIA Jetson Nano, to ensure efficient and reliable operation in dynamic field conditions. By leveraging Docker for deployment, the system is designed to perform real-time data processing, enabling immediate responses to changes in soil conditions and reducing the dependency on external connectivity [17].

The advantages of an AI-based automatic irrigation system include optimization of water use, minimization of environmental impact, and increased crop yields [18][19]. By delivering timely and appropriate irrigation, water use can be saved, reducing waste and supporting sustainable agriculture. Efficient water management also mitigates negative environmental impacts, such as reducing soil erosion and water pollution [20]. This research aims to develop and investigate irrigation systems that are more adaptive, intelligent, and responsive to dynamic agricultural conditions [21][22]. By adopting a neural network model and utilizing edge computing devices [23]–[25], this system offers a robust and scalable solution to the challenges faced in modern agriculture.

The main contribution of this research is the development and implementation of an AI-based smart irrigation system that adopts neural network algorithms for better decision-making in irrigation management. By using neural network algorithms, this system is expected to provide a more accurate and reliable solution for managing water distribution, ultimately saving water resources and increasing agricultural efficiency. Furthermore, this research contributes to the development of methods that can be adapted and applied to various types of crops and agricultural conditions, thereby expanding the scope and benefits of smart irrigation technology.

## 2. METHOD

The method of this research is depicted in Figure 1, showing the stages of the process that will be completed. Starting from collecting datasets obtained from open source datasets, pre-processing is done, which is a way to find whether the data has anomalies and outliers, followed by dividing the data into 3 parts, namely training set, testing set, and validation set, the application of the neural network model is tested and parameter tuning is carried out, then the model is evaluated by getting the best accuracy value, next the model is saved in .pth format for testing using edge computing devices, finally automatic decision making for irrigation.

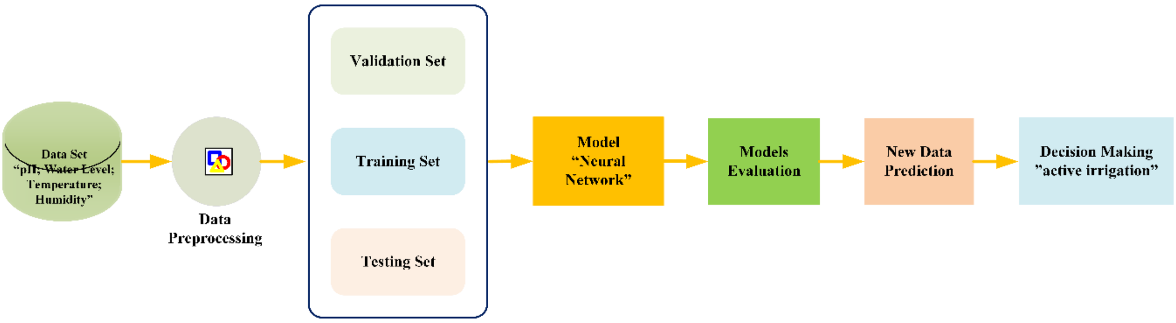


**Figure 1.** Research method block

### 2.1. Data and Pre Process

The research data used in this study was obtained from several reputable open-source datasets, containing essential environmental features such as pH, water level, temperature, and humidity. These features play a critical role in predicting soil conditions and making informed irrigation decisions. Since the data originated from open repositories, it can be further extended and refined for future applications. To ensure the dataset was clean and reliable for model training, several preprocessing steps were conducted. Initially, the data underwent cleaning to handle missing values, outliers, and anomalies. Missing values were imputed where appropriate, while outliers were addressed using the interquartile range (IQR) method. The categorical variable soil_status was encoded into numeric labels using the LabelEncoder, and the waterpumponoff column was converted into binary values, with "ON" mapped to 1 and "OFF" mapped to 0. Finally, all features were standardized using the StandardScaler, transforming them to have a mean of 0 and a variance of 1. This standardization ensured consistency across features, reducing the risk of bias from features with larger scales. Together, these preprocessing steps created a robust dataset suitable for effective model training and evaluation.

### 2.2. Data Splitting

After preprocessing, the data was split into three distinct subsets to enable comprehensive model training and evaluation. Seventy percent of the data was allocated to the training set, which was used to optimize the neural network's parameters. Fifteen percent was set aside as the validation set, which played a crucial role in tuning hyperparameters and performing cross-validation to prevent overfitting. The remaining fifteen percent was reserved as the testing set, used to evaluate the model's final performance on unseen data. This 70-15-15 split ensures that the model is trained on sufficient data while maintaining independent datasets for validation and testing to assess its generalizability and robustness.

### 2.3. Application of Neural Network Models

Following the data splitting process, a neural network model was defined and trained to predict soil conditions. The model architecture consisted of an input layer, two hidden layers, and an output layer. The input layer received four features—pH, water level, temperature, and humidity—while each hidden layer contained 64 neurons and employed the ReLU activation function to capture complex, non-linear relationships within the data. The output layer, designed for multi-class classification, included three neurons corresponding to soil condition categories (Base, Ideal, and Dry) and used a softmax activation function to produce class probabilities. The model's performance was optimized using the CrossEntropyLoss function to quantify prediction errors and the Adam optimizer, which dynamically adjusted learning rates for efficient convergence. The training process involved forward passes to generate predictions, loss calculations to measure errors, backward passes to compute gradients, and weight updates to iteratively improve the model's accuracy. Through this iterative learning process, the neural network adapted its weights and biases to recognize patterns in the data, enabling reliable predictions of soil conditions.

### 2.4. Model Evaluation

Once the model is trained, its performance is tested using the test set. This evaluation involves using the model to predict values on the test set and calculating evaluation metrics such as accuracy, classification reports (precision, recall, F1-score), confusion matrix, ROC curve, and AUC. Accuracy is the proportion of correct predictions from the total number of predictions. Precision is the proportion of correct positive predictions out of all positive predictions. Recall is sensitivity or true positives, which measures the proportion of true positives from all actual positive cases. The F1 Score is the harmonic average of precision and recall. F1 Score describes a more balanced model performance when there is a trade-off between precision and recall. Macro Average measures the average metric (eg precision, recall, F1-score) for each class without taking into account class proportions [26].

$$\text{accuracy} = \left(\frac{\sum \text{TP} + \sum \text{TN}}{\sum \text{TP} + \sum \text{TN} + \sum \text{FP} + \sum \text{FN}}\right) \times 100 \tag{1}$$

$$\text{Precision} = \left(\frac{\sum \text{TP}}{\sum \text{TP} + \sum \text{FP}}\right) \times 100 \tag{2}$$

$$\text{Recall} = \left(\frac{\sum \text{TP}}{\sum \text{TP} + \sum \text{FN}}\right) \tag{3}$$

$$F1\ Score = 2^{*}\left(\frac{Precision^{*}Recall}{Precision + Recall}\right)^{*}100 \tag{4}$$

$$\text{Macro Avg Measure} = \frac{1}{N}\sum_{i=1}^{N} \text{Measure in class } i \tag{5}$$

### 2.5. New Data Process

Once the model has been trained, it is used to predict new data reflecting real-time ground conditions. Before making predictions, the new data is standardized using the same scaler applied during the training phase to ensure consistency and compatibility with the trained model. The standardized data is then fed into the neural network, which predicts the soil state, such as “Critical,” “Ideal,” or “Dry.” These predictions form the basis for irrigation decisions, providing actionable insights to regulate water distribution efficiently. This stage demonstrates how the trained model transitions from theoretical evaluation to practical application, enabling real-time decision-making in agricultural environments.

### 2.6. Edge Computing Process

The edge computing process leverages an NVIDIA Jetson Nano as a local server to execute real-time predictions on new data. The trained neural network model is saved in PyTorch’s .pth format and deployed on the Jetson Nano using Docker containers to ensure portability and ease of implementation. The Jetson Nano, equipped with an NVIDIA Maxwell architecture GPU featuring 128 CUDA cores, a Quad-core ARM Cortex-A57 processor, and 4GB LPDDR4 memory, delivers powerful on-site computational capabilities. Running on a Debian-based operating system, the device consumes a maximum of 10 watts, making it an energy-efficient solution for resource-constrained environments. By processing data locally, the edge computing setup eliminates latency issues associated with cloud-based systems and ensures reliable operations in remote locations with limited or no internet connectivity.

### 2.7. Irrigation Decision Making Process

The final stage of the system involves making automated irrigation decisions based on the model’s predictions. When the soil status is predicted as “Critical,” the system activates the water pump to provide the necessary irrigation. Conversely, when the status is classified as “Ideal” or “Dry,” the pump remains off, conserving water. This decision-making logic is implemented seamlessly, relying entirely on the real-time predictions made by the edge computing device. The automated nature of this process ensures timely and accurate irrigation management, significantly reducing water waste and optimizing resource utilization. By integrating prediction with action, this system provides a robust solution for sustainable agriculture, enhancing productivity and minimizing environmental impact.

### 2.8. Research Procedure Algorithm

The logic procedure stages are shown in Algorithm 1 below, where the metrics evaluation uses Confusion Matrix, Classification Report, and ROC Curve.

```
1. Load the dataset
      ➢ Load data from file_path.
2. Preprocess the data
```

```
        ➢ Encode soil_status with LabelEncoder.
        ➢ Convert water pump off to numerical values.
        ➢ Select features and targets.
        ➢ Standardize features using StandardScaler.
3.  Split the data
        ➢ Split data into training, validation, and testing sets.
4.  Convert data to PyTorch tensors
5.  Define the Neural Network model
6.  Initialize the model, criterion, and optimizer
        ➢ Initialize model, criterion, and optimizer.
7.  Train the model
        ➢ Train for num_epochs epochs.
        ➢ Print loss every 10 epochs.
8.  Validate the model
        ➢ Compute validation loss and accuracy.
9.  Evaluate the final model
        ➢ Compute test predictions.
10. Convert numerical predictions back to original labels
11. Evaluate the model
        ➢ Compute test accuracy and classification report.
12. Compute and plot the confusion matrix
13. Compute ROC curve and AUC for each class
14. Define function to predict soil status
15. Define function to decide water pump status
16. Example new data prediction
17. Define and print model weights
```

## 3. RESULTS AND DISCUSSION

### 3.1. Descriptive Analysis of Data

Table 1 is a descriptive statistic of the data showing that the features pH, water level, temperature, and humidity have a mean of 3.28, 2.51, 1.43, and 1.79, respectively, with a standard deviation of 1.00 for each feature, indicating normalization of the data. The minimum values for pH, water level, temperature, and humidity were 3.61, 3.80, 3.54, and 4.12, respectively, while the maximum values were 3.94, 4.29, 4.13, and 3.80, indicating For each feature, the first quartile (25%), median (50%), and third quartile (75%) are around 0.68, 0.00-0.01, and 0.67-0.68, respectively, indicating a fairly symmetric data distribution. Before undertaking further analysis or data-driven decision making, it is important to understand the basic features of the data.

**Table 1.** Summary of research statistical data

| | Ph | Water level | temperature | humidity |
|---|---|---|---|---|
| **mean** | 3.28 | 2.51 | 1.43 | 1.79 |
| **std** | 1.00 | 1.00 | 1.00 | 1.00 |
| **min** | 3.61 | 3.80 | 3.54 | 4.12 |
| **25%** | 0.68 | 0.67 | 0.67 | 0.68 |
| **50%** | 0.01 | 0.00 | 0.01 | 0.01 |
| **75%** | 0.67 | 0.68 | 0.68 | 0.68 |
| **max** | 3.94 | 4.29 | 4.13 | 3.80 |

Dataset variables such as pH, water level, temperature, and humidity do not have a significant linear relationship as depicted from the feature correlation matrix; for each pair of features, the correlation value is almost zero. pH has a correlation of -0.00 with temperature, -0.01 with temperature, and -0.01 with humidity. Temperature has a correlation of -0.01 with pH, -0.01 with water level, and -0.01 with humidity. Humidity has a correlation of -0.00 with pH, -0.01 with water level, and -0.01 with temperature. Perfect correlation is indicated by a value of 1 on the diagonal. Overall, this shows that each feature of the dataset tends to be different from each other, contributing unique and unlimited information to the analysis or prediction model can be seen in Figure 2.

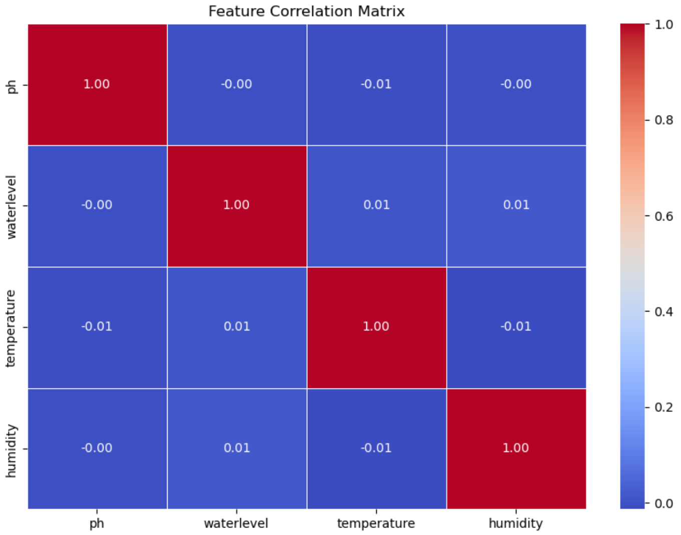


**Figure 2.** Data Feature Correlation

### 3.2. Model Classification Report

The classification report in Table 2 provides a comprehensive overview of how the model successfully classifies alkaline, ideal, and dry soils. The precision value for the alkaline class is 0.97, which shows that the model is very accurate in identifying alkaline soil, and the precision value for the ideal class is 0.90, which shows that the model is less accurate in identifying dry soil. How well the model can identify all actual events from each class is indicated by the recall value. The alkaline recall was 0.80, indicating that only 80% of alkaline soil events were correctly identified. The ideal recall is 0.99, indicating that almost all ideal soil events are identified correctly. The lowest dry recall was 0.74, indicating that 26% of Dryland events were not correctly identified by the model.

**Table 2.** Classification Performance Report

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| **Basa** | 0.97 | 0.8 | 0.87 | 249 |
| **Ideal** | 0.9 | 0.99 | 0.94 | 1005 |
| **Dry** | 0.96 | 0.74 | 0.83 | 246 |
| **Accuracy** | | | 0.91 | 1500 |
| **Macro Avg** | 0.94 | 0.84 | 0.88 | 1500 |
| **Weighted Avg** | 0.92 | 0.91 | 0.91 | 1500 |

The F1-Score value provides an in-depth picture of how the two are balanced because it is a harmonic value of precision and recall. The overall base F1-Score was 0.87, Ideal 0.94, and Dry 0.83, indicating that the base class had the best model performance, with the dry and base classes slightly worse. The overall model accuracy was 0.91, indicating that 91% of all predictions were correct. Macro average (avg) precision, recall, and F1-Score give simple average values of 0.94, 0.84, and 0.88. With a precision of 0.92, recall of 0.91, and F1-Score of 0.91, the weighted average value considers the proportion of each class in the dataset. Overall, the report shows that the model classifies ideal soils very well, even for alkaline and dry soils. However, there is still room for improvement in detecting dry soil events more accurately. Figure 3 is a confusion matrix where with 991 correct predictions from 1005 events and several small errors, the classification model shows very good performance in detecting ideal soil. Overall, the model shows high accuracy but requires improvement in differentiating between Ideal and Dry soils and reducing prediction errors for dry soils. For alkaline soil, the model successfully identified 198 of 249 events correctly but frequently misclassified 51 events as Ideal. For dry land, predictions were more difficult, with 181 of 246 events correctly predicted, and 65 events incorrectly classified as Ideal.

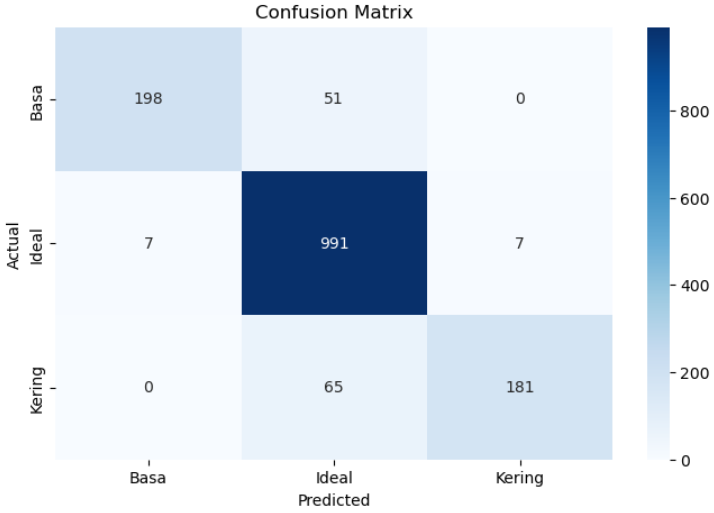


**Figure 3.** Confusion matrix testing

### 3.3. ROC (Receiver Operating Characteristic) Curve

The performance of the classification model is shown in Figure 4 where the receiver operating characteristic curve (ROC) and area under the curve (AUC). The base class had an AUC of 0.99, indicating almost perfect model performance in distinguishing the classes. The ideal class has an AUC of 0.97, which indicates excellent performance, although slightly lower than the base class. The Dry class had an AUC of 0.98, indicating that the model was also very good at classifying this class. The ROC curve that approaches the top left corner of the plot indicates that the model has a high True Positive rate and a low False Positive rate, indicating a strong ability to distinguish classes.

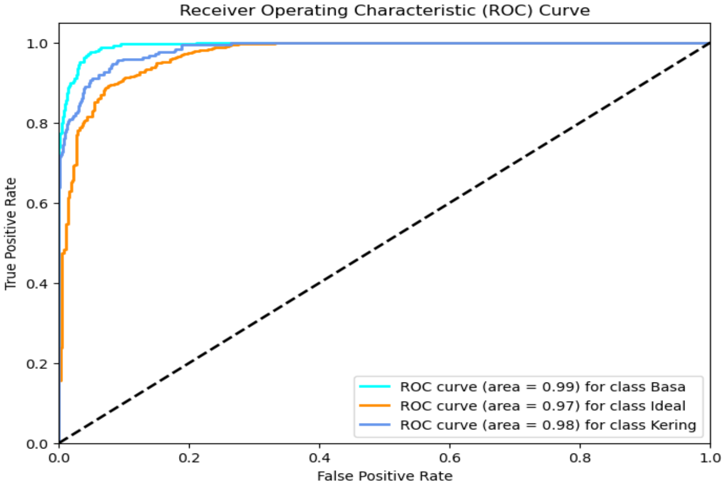


**Figure 4.** Classification Model Performance

Overall, the high AUC values for all three classes indicate that the model has an excellent ability to differentiate between different classes in the dataset. This indicates that the model has strong and reliable performance in this classification task. Although the model performance is very good, there is room for improvement, especially in reducing prediction errors between the Ideal and Dry classes. These results are consistent with the analysis of the confusion matrix which shows that although there are some misclassifications, the model as a whole is very effective.

### 3.4. Weights Model Test Results

The structure and parameters of the two main layers of the neural network model, fc1 and fc2, are depicted in Table 3. The fc1 layer is weighted with size [64, 4], which shows that there are 64 neurons in the layer and each neuron is connected to four input features. The sample weight values indicate some positive and negative relationships between neurons in a layer and input features. For the fc1 layer, the bias, or bias, has the measure [64], which means that each neuron has one bias value. The bias value example shows how the model adjusts each neuron after considering the weighted input.

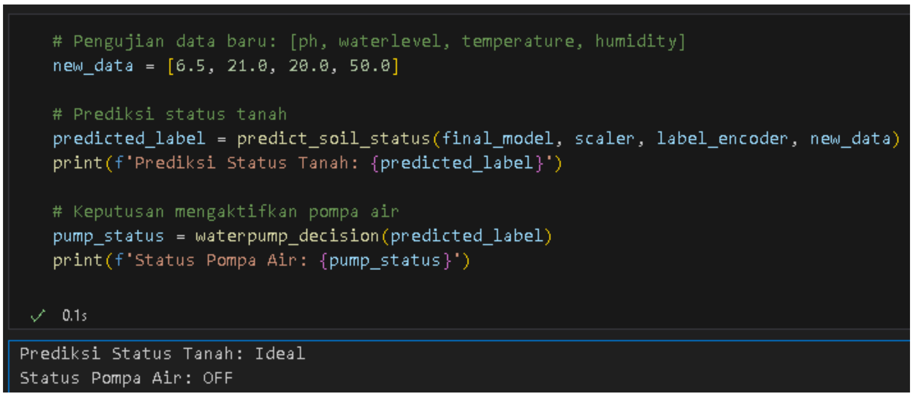
**Table 3.** Weights Model Values

| Layers | Size | Values |
|---|---|---|
| fc1. weight | torch.Size([64, 4]) | tensor([[ 0.1125, -0.2145, -0.3016, -0.1764], [ 0.1489, -0.0966, -0.0348, -0.2827]], grad_fn=<SliceBackward0>) |
| fc1. bias | torch.Size([64]) | tensor([-0.3368, 0.4285], grad_fn=<SliceBackward0>) |
| fc2. weight | torch.Size([3, 64]) | tensor([[-0.1059, -0.1650, 0.2084, 0.0405, 0.0287, 0.1378, 0.1303, 0.0735, -0.2481, -0.0006, 0.0022, 0.1434, 0.0513, -0.0839, -0.135 4, -0.1485, -0.0037, -0.0493, 0.1425 , 0.2289, 0.0413, 0.1221, -0.0848, 0.0101, -0.0842, 0.0428, 0.1489, 0.1425, -0.2589, 0.2054, -0.1963, -0.2122, 0.2797, -0.2490, 0.0780, -0.1820, -0.0015, -0.0413, 0.1753, -0.0117, -0.0748, 0.0683, 0.3205, 0.1169, -0.2224, -0.0749, 0.2081, -0.0464, 0.1690, 0.0202, -0.0726, -0.1370, -0.1954, -0.2234, -0. 0782, -0.1150, -0.2471, -0.0409 , 0.2885, 0.0308, -0.1052, -0.1417, 0.0849, -0.2577], [-0.0948, 0.0716, 0.0922, 0.0767, 0.1360, -0.1619, 0.1262, -0.0110, -0.0113, 0.0574, 0.0144, -0.0980, 0.0093, - 0.0988, -0.1029, 0.2305, 0.0315, 0.1749, -0.1632, -0.0439, 0.0601, -0.2105, 0.0394, 0.2112, -0.0241, 0.2074, 0.1453, -0.0750, 0. 2033, 0.0031, -0.0172, 0.0664, -0.1008, 0.0558, 0.1101, 0.1285, 0.2209, 0.1839, -0.0280, -0.0367, 0.1320, -0.0138, -0.1695, 0.0551, 0.0588, -0.0358, -0.0561, 0.0235, -0.0091, -0. 0546, -0.0280, 0.1343, 0.1196, 0.0998, -0.0357, -0.0984, 0.1326, -0.0154, 0.0039, 0.0767, -0.0653, 0.0270, 0.0699, 0.2085]], grad_fn=<SliceBackward0>) |
| fc2. bias | torch.Size([3]) | tensor([-0.1074, 0.1355], grad_fn=<SliceBackward0>) |

The fc2 layer has three large output neurons [3, 64], indicating that there are 64 output neurons connected to the fc1 layer. This corresponds to the desired three-class classification: basic, ideal, and dry. The weight value of this layer shows the complexity of the relationship between the hidden layer and the output layer. The bias value of the fc2 layer has a size of [3], with a bias value for each output neuron. This example layer bias value shows how the model adjusts the final output based on the input considered from the previous layer.

### 3.5. Testing New Data

To ensure that the model that has been trained and tested is tested, Figure 5 is a test of new data carried out randomly, where the results obtained are that the model that has been built can predict the status of soil conditions and recommendations for irrigation whether it is necessary to activate the water pump.



```
# Pengujian data baru: [ph, waterlevel, temperature, humidity]
new_data = [6.5, 21.0, 20.0, 50.0]

# Prediksi status tanah
predicted_label = predict_soil_status(final_model, scaler, label_encoder, new_data)
print(f'Prediksi Status Tanah: {predicted_label}')

# Keputusan mengaktifkan pompa air
pump_status = waterpump_decision(predicted_label)
print(f'Status Pompa Air: {pump_status}')
```

```
Prediksi Status Tanah: Ideal
Status Pompa Air: OFF
```

**Figure 5.** Test new data

Once the ground status is predicted, the water pump decision function is used to activate the water pump. If the ground status is “Critical”, the water pump is activated. Since the predicted ground status is “Ideal”, the water pump remains in the “OFF” condition. This shows that the model can not only predict soil status well but can also help make automatic decisions based on predicted conditions. This output is critical for automated irrigation management, as real-time options can optimize water use and increase agricultural efficiency.

## 4. CONCLUSION

The results of this research demonstrate that artificial intelligence (AI) has significant potential to increase agricultural crop yields and conserve water resources. By integrating neural network models with edge computing devices, this system can predict soil status in real time and regulate water distribution automatically. The model was rigorously trained and evaluated using preprocessed datasets, achieving high accuracy in classifying alkaline, ideal, and dry soils. The strong Area Under the Curve (AUC) values across all classes confirm the model's reliability in differentiating between soil conditions. This AI-based smart irrigation system offers multiple benefits, including increased crop yields, optimized water use, and reduced environmental impact. Its real-time decision-making capability, supported by edge computing, ensures efficient and reliable operation in dynamic agricultural settings. This makes it particularly valuable in remote areas with limited internet connectivity, where traditional cloud-based systems might not perform effectively. Despite its strengths, this study has certain limitations. The dataset used, while sufficient for proof-of-concept validation, could be expanded to include more diverse soil types, crops, and environmental conditions to improve generalizability. Additionally, the system's long-term reliability and performance in extreme field conditions require further investigation. Future research could explore integrating additional features, such as soil salinity or weather forecasts, to enhance predictive accuracy and robustness. Another promising direction would be implementing renewable energy sources, such as solar power, to make the system more sustainable. The practical implications of this research are far-reaching. The system's adaptability to a wide range of crops and farming conditions makes it a scalable solution for farmers aiming to improve irrigation efficiency. By reducing water waste and minimizing environmental degradation, this technology aligns with Sustainable Development Goals (SDGs), particularly SDG 12 (Responsible Consumption and Production) and SDG 13 (Climate Action).

## ACKNOWLEDGEMENTS

This work was partially supported by Mercu Buana University

## REFERENCES

[1] Z. Gu, T. Zhu, X. Jiao, J. Xu, and Z. Qi, "Neural network soil moisture model for irrigation scheduling," *Comput. Electrons. Agric.*, vol. 180, no. 1, p. 105801, 2021, https://doi.org/10.1016/j.compag.2020.105801.

[2] M. M. Rahman, *et al.*, "Farm mechanization in Bangladesh: A review of the status, roles, policy, and potentials," *Journal of Agriculture and Food Research*, vol. 6, p. 100225, 2021, https://doi.org/10.1016/j.jafr.2021.100225.

[3] K. Prema and CM Belinda, "Smart farming: IoT based plant leaf disease detection and prediction using deep neural nework with image processing," *Int. J. Innov. Technol. Explore. Eng.*, vol. 8, no. 9, p. 3081–3083, 2019, https://doi.org/10.35940/ijitee.I7707.078919.

[4] R. K Jain, "Experimental performance of smart IoT-enabled drip irrigation system using and controlled through web-based applications," *Smart Agriculture. Technol.*, vol. 4, p. 100215, 2023, https://doi.org/10.1016/j.atech.2023.100215.

[5] S. Solanki, A. Bagchi, P. Johri, and A. Tyagi, "A study on smart irrigation system using IOT," In *Challenges and Opportunities in Industrial and Mechanical Engineering: A Progressive Research Outlook*, pp. 900-908, 2024, https://doi.org/10.1201/9781032713229-100.

[6] M. Pramaniket *al.*, "Smart Agricultural Technology Automation of soil moisture sensor-based basin irrigation system," *Smart Agric. Technol.*, vol. 2, p. 100032, 2022, https://doi.org/10.1016/j.atech.2021.100032.

[7] A. Goap, D. Sharma, AK Shukla, and C. Rama Krishna, "An IoT based smart irrigation management system using Machine learning and open source technologies," *Comput. Electrons. Agric.*, vol. 155, p. 41–49, 2018, https://doi.org/10.1016/j.compag.2018.09.040.

[8] A. T. Rathod and A. U. Awate. Design and development of automated irrigation system. 2018. https://doi.org/10.31142/ijtsrd19178.

[9] JM Dealmeida, L. Dasilva, CB Bonato Both, CG Ralha, and MA Marotta, "Artificial Intelligence-Driven Fog Radio Access Networks: Integrating Decision Making Considering Different," *IEEE Veh. Technol. Mag.*, vol. 16, no. 3, p. 137–148, 2021, https://doi.org/10.1109/MVT.2021.3078417.

[10] S. Raja Gopal and VSV Prabhakar, "Intelligent edge based smart farming with LoRa and IoT," *Int. J. Syst. Assur. Eng. Manag.*, 2022, https://doi.org/10.1007/s13198-021-01576-z.

[11] A. D. Boursianiset *al.*, "Smart Irrigation System for Precision Agriculture - The AREThOU5A IoT Platform," *IEEE Sens. J.*, vol. 21, no. 16, p. 17539–17547, 2021, https://doi.org/10.1109/JSEN.2020.3033526.

[12] C. C. Baseca, S. Sendra, J. Lloret, and J. Tomas, "A smart decision system for digital farming," *Agronomy*, vol. 9, no. 5, 2019, https://doi.org/10.3390/agronomy9050216.

[13] S. L. Chukkapalli*et al.*, “Ontologies and artificial intelligence systems for the cooperative smart farming ecosystem,” *IEEE Access*, vol. 8, p. 164045–164064, 2020, https://doi.org/10.1109/ACCESS.2020.3022763.

[14] P. K. Kashyap, S. Kumar, A. Jaiswal, M. Prasad, and AH Gandomi, “Towards Precision Agriculture: IoT-Enabled Intelligent Irrigation Systems Using Deep Learning Neural Network,” vol. 21, no. 16, p. 17479–17491, 2021, https://doi.org/10.1109/JSEN.2021.3069266.

[15] E. M. B. M. Karunathilake, A. T. Le, S. Heo, Y. S. Chung, and S. Mansoor, “The path to smart farming: Innovations and opportunities in precision agriculture,” *Agriculture*, vol. 13, no. 8, p. 1593, 2023, https://doi.org/10.3390/agriculture13081593.

[16] N. Ahmed, D. De and I. Hussain, "Internet of Things (IoT) for Smart Precision Agriculture and Farming in Rural Areas," in *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4890-4899, Dec. 2018, https://doi.org/10.1109/JIOT.2018.2879579.

[17] S. Katiyar and A. Farhana, “Smart Agriculture: The Future of Agriculture using AI and IoT,” *J. Comput. Sci.*, vol. 17, no. 10, p. 984–999, 2021, https://doi.org/10.3844/jcssp.2021.984.999.

[18] Z. Arifin, M. Saeri, and Balai, "Water Management and Mulch on Shallot Plants in Dry Land," no. Sinaga 2008, p. 159–168, 2019, https://doi.org/10.21082/jhort.v29n2.2019.p159-168.

[19] R. S. Basuki, "Identification of Problems and Analysis of Shallot Farming in the HighlandsDuring the Rainy Season in Majalengka Regency," *J. Hortik.*, vol. 24, no. 3, p. 266, 2016, https://doi.org/10.21082/jhort.v24n3.2014.p266-275.

[20] A. A. Farooque*et al.*, “Forecasting daily evapotranspiration using artificial neural networks for sustainable irrigation scheduling,” *Irrig. Sci.*, no. 0123456789, 2021, https://doi.org/10.1007/s00271-021-00751-1.

[21] S. Balachandran, S. Lakshmi, and N. Rajendran, "Irrigation system using hyperspectral data and machine learning techniques for smart agriculture," *J. Comput. Sci.*, vol. 16, no. 4, p. 576–582, 2020, https://doi.org/10.3844/jcssp.2020.576.582.

[22] C. Cambra, S. Sendra, J. Lloret, and R. Lacuesta, “Smart system for bicarbonate control in irrigation for hydroponic precision farming,” *Sensors (Switzerland)*, vol. 18, no. 5, 2018, https://doi.org/10.3390/s18051333.

[23] R. N. Bashir*et al.*, “Smart reference evapotranspiration using Internet of Things and hybrid ensemble machine learning approach,” *Internet of Things (Netherlands)*, vol. 24, p. 100962, 2023, https://doi.org/10.1016/j.iot.2023.100962.

[24] B. P. Sangeetha, *et al.*, “IOT based smart irrigation management system for environmental sustainability in India,” *Sustainable Energy Technologies and Assessments*, vol. 52, p. 101973, 2022, https://doi.org/10.1016/j.seta.2022.101973.

[25] C. Kishore Kumar and V. Venkatesh, “Design and development of IOT based intelligent agriculture management system in greenhouse environment,” *Int. J.Eng. Adv. Technol.*, vol. 8, no. 5 Special Issue 3, p. 47–52, 2019, https://doi.org/10.35940/ijeat.E1013.0785S319.

[26] D. Wu, H. Xu, Z. Jiang, W. Yu, X. Wei, and J. Lu, “EdgeLSTM: Towards Deep and Sequential Edge Computing for IoT Applications,” *IEEE/ACM Trans. Netw.*, vol. 29, no. 4, p. 1895–1908, 2021, https://doi.org/10.1109/TNET.2021.3075468.

## AUTHOR BIOGRAPHY

**Freddy Artadima Silaban** (Student Member, IEEE) received a bachelor's degree in computer systems and a master's degree in electrical engineering with a specialization in embedded systems. He is currently pursuing a Ph.D. degree in Electrical Engineering and Informatics at the Bandung Institute of Technology (ITB), Bandung, Indonesia. His research focuses on embedded systems, Artificial Intelligence, the Internet of Things, and Edge Computing.

**Ahmad Firdausi** (Student Member, IEEE) received a bachelor's and master's degree in electrical engineering with specialization in antenna design, He is currently pursuing a Ph.D. degree in Electrical Engineering at Institut Teknologi Sepuluh November (ITS) Surabaya, Indonesia. His research interests include radio-frequency (RF) circuit design, radio-frequency identification (RFID), signals and systems, wearable and flexible antennas, metamaterial-based antennas, implantable antennas, multi-input-multi-output (MIMO) antennas, mm-wave antennas, 5G antennas, antennas arrays, metasurfaces antennas, dielectric resonator antennas (DRAs), photonics antennas, antennas for the Internet of Things applications, ultra-wideband (UWB) antennas, wideband antennas, reconfigurable antennas, terahertz antennas, rectennas for energy harvesting applications, active sensors, and the IoT-based communications devices. He is a member of the IEEE Antennas and Propagation Society.