Journal of Computing Innovations and Emerging Technology

Volume 1 | Issue 2 | December (2025)

Published by **Novamind Press**

JCIET

Journal of Computing Innovations and Emerging Technology

Implementation of IoT-Based Automatic Irrigation System Using Decision Tree Algorithm on Hydroponic Garden at Institut Shanti Bhuana Bengkayang

Kristian Novando[1*], Noviyanti P[2]

[1,2] *Department of Information Technology, Institut Shanti Bhuana, Bengkayang, Kalimantan Barat, Indonesia*

**Corresponding author: kristian2108@shantibhuana.ac.id*

Abstract* — This study presents the development and implementation of an automatic irrigation system based on the Internet of Things (IoT) utilizing the Decision Tree algorithm. The system was applied in a hydroponic garden at Institut Shanti Bhuana Bengkayang. It employs a water level sensor to detect the volume of water, which is then processed using the Decision Tree classification to determine whether the irrigation valve should be opened or closed. Data collected from the sensor were analyzed both manually and programmatically to find the optimal threshold for decision-making. The system was integrated with the Blynk platform, allowing real-time monitoring and control. Testing was conducted over 7 days with 210 data points, and the classification model achieved an accuracy of 100%. The results indicate that the proposed system effectively automates irrigation, minimizes manual intervention, and provides a reliable solution for small-scale smart farming applications.

***Keywords*—IoT, Decision Tree, Smart Irrigation, Hydroponic Garden, Water Level Sensor**

Manuscript received 22 July. 2025; revised 09 August. 2025; accepted 12 August. 2025. Date of publication 15 Des. 2025.

I. Introduction

The global agricultural [1] landscape is facing unprecedented challenges due to rapid population growth [2], climate change, and decreasing availability of arable land. Traditional farming practices are increasingly inadequate in addressing the growing food demand, which is projected to rise by 70% by 2050 to support an estimated 9.3 billion people worldwide [3][4] In response to these challenges In recent years, there have been significant advancements in the integration of smart technologies for precision farming within the agricultural domain [5], innovative technologies such as the Internet of Things (IoT) have emerged as transformative tools in modern agriculture, offering real-time data collection, automation, and precision farming capabilities[6][7].

IoT has great potential and is one of the key areas for future development of internet services[8]. The evolution of tecchnology has made it possible to create such integrated devices that, although limited in computational resources, can execute relatively complex tasks especially in the open field, precision agriculture can improve its management and, thus, its productivity [9][10][11]. IoT-based systems have proven particularly beneficial in controlled environments like hydroponic gardens, where parameters such as nutrient levels, temperature, humidity, and water flow require constant monitoring to ensure optimal plant growth. The major benefit of this innovation was the capability to producer and consumer services in real time[12]. These systems can significantly reduce labor requirements and improve water-use efficiency, which is especially important in regions with limited agricultural infrastructure[7]. At Institut Shanti Bhuana in Bengkayang, the adoption of hydroponic gardening presents an opportunity to integrate such technology into educational and sustainable food production initiatives.

The incorporation of artificial intelligence [13], particularly decision tree algorithms [14], further enhances the effectiveness of IoT-based irrigation systems by enabling automated decision-making [15] based on real-time environmental data. Decision trees provide a transparent and interpretable method [16] [17] of evaluating sensor inputs to determine precise irrigation schedules, helping to minimize resource waste and ensure consistent crop health[6][18] .

This research focuses on the implementation of an IoT-based automatic irrigation system using a decision tree algorithm within a hydroponic setup at Institut Shanti Bhuana Bengkayang. The aim is to develop a scalable, low-cost, and efficient solution that contributes to smart agriculture [19]practices in educational and rural settings.

II. METHODOLOGY

This study used a quantitative research method involving several stages: identifying the problem, analyzing needs, designing the system, implementing the prototype, and evaluating the results.

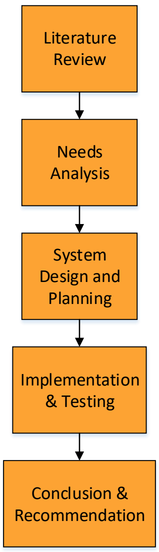

Fig 1. Research Flowchart

Fig 1 illustrates the research design flowchart outlines the step-by-step process followed in the development of the IoT-based automatic irrigation system using the Decision Tree algorithm. The process begins with literature review, which involves gathering and studying references related to IoT-based irrigation systems and Decision Tree methodologies. This step also includes field observations and interviews with members of the Hydroponic UKM at Institut Shanti Bhuana to better understand existing challenges and requirements.

The needs analysis follows, which defines both functional and non-functional requirements. This includes determining the necessary hardware (such as water level sensors, ESP8266, and micro servos) and software tools (Arduino IDE, Blynk, and Decision Tree classification logic) needed to develop the system.

In this study, the Decision Tree classification model is implemented directly on the ESP8266 NodeMCU microcontroller. The algorithm is hard-coded in the Arduino IDE environment, using threshold-based rules derived from entropy and information gain calculations. The ESP8266 processes the raw water level sensor readings in real time without relying on any external server or cloud-based computation for decision-making. This local processing approach was chosen to reduce latency, ensure the system can function even during internet disruptions, and minimize dependency on third-party platforms. The Blynk application is used solely for real-time monitoring and manual override control, while all classification logic and valve actuation decisions are handled internally by the microcontroller.

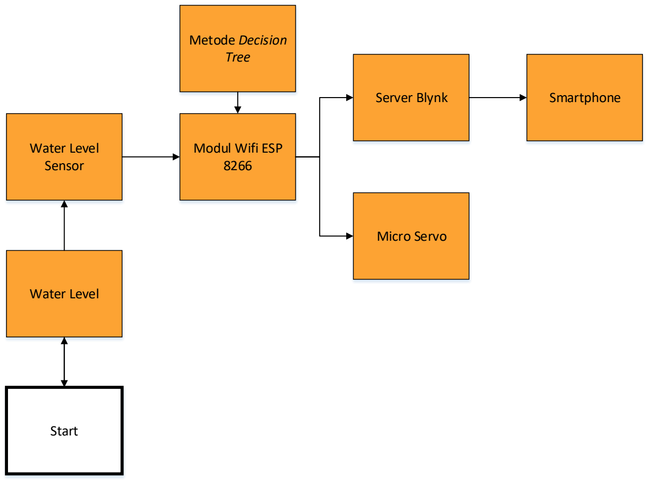

Fig 2. Communication Architecture

Fig 2 illustrates the IoT communication architecture in this system is designed for direct data exchange between the ESP8266 microcontroller and the Blynk cloud server via Wi-Fi. The water level sensor sends readings to the ESP8266 every 5 seconds, which then transmits the data to the Blynk application for real-time monitoring. Control commands from the user are also sent through Blynk to the microcontroller, allowing bidirectional communication. In the event of temporary network instability, the classification and valve control processes continue to operate locally on the ESP8266 to ensure uninterrupted irrigation. Once the connection is restored, buffered sensor data is transmitted to synchronize the Blynk dashboard.

Once the requirements are defined, the system design and planning stage begins. A prototype is developed by integrating all components into a working model that represents how the system should function. The design phase also includes the creation of flowcharts and diagrams to visualize data flow and system interactions.

This is followed by the implementation and testing phase, where the prototype is tested in a real environment (the hydroponic garden). The testing is done over a period of 7 days to gather real-time data and assess the system's accuracy, reliability, and responsiveness.

This is followed by the conclusion and recommendation phase, where the results from system implementation are analyzed to evaluate whether the objectives of the research have been achieved. The analysis confirms that the IoT-based automatic irrigation system successfully responded to real-time water level data and was able to control irrigation actions through a micro servo mechanism. By applying the Decision Tree algorithm, the system was able to classify sensor input and make appropriate decisions, such as opening or closing the water valve, with a reported accuracy of 100% during testing.

Furthermore, the integration with the Blynk application provided a practical and accessible interface that allowed users to monitor system activity remotely through smartphones. This feature significantly reduced the need for manual observation and enabled users to manage irrigation more efficiently.

In light of these achievements, several recommendations are proposed for future system development. First, additional sensors such as temperature and humidity sensors can be integrated to support more dynamic decision-making based on multiple environmental conditions. Second, the system can be enhanced with a dedicated web-based dashboard to display historical data, generate visual performance analytics, and reduce dependency on third-party platforms. Lastly, notification features through commonly used messaging applications like WhatsApp or Telegram can be implemented to improve system accessibility and ensure timely user responses.

This final stage emphasizes how the system not only meets its primary objectives but also offers a scalable foundation for future improvements and broader adoption in smart farming environments.

The implementation was conducted at the hydroponic garden of Institut Shanti Bhuana Bengkayang using a smart irrigation system integrated with an IoT architecture and the Decision Tree algorithm.

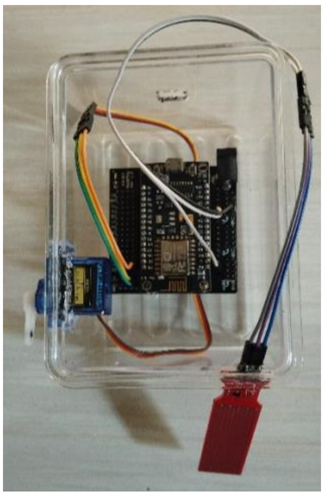

Fig 3. Hardware Prototype

Fig 2 shows the hardware prototype of the IoT-based automatic irrigation system developed in this study. The system consists of an ESP8266 NodeMCU microcontroller as the processing unit, a water level sensor to detect water volume, and a micro servo motor that functions as an actuator to control water flow. These components are connected using jumper wires and mounted inside a transparent plastic enclosure for visibility and protection. The design reflects a compact, low-cost smart irrigation solution that supports real-time data processing and automated decision-making based on water level readings.

The software implementation was done using the Arduino IDE to program the ESP8266 microcontroller, which received input from the sensor and triggered actuator responses. The Blynk application was used for real-time monitoring and remote control of the irrigation system through smartphones.

A similar implementation using ESP-based microcontrollers and the Blynk platform for real-time hydroponic monitoring was demonstrated by highlighting the feasibility and practicality of low-cost smart farming solutions. The system was capable of displaying water levels and the status of the valve (open or closed) in real time via the Blynk dashboard.

For classification, this study used the Decision Tree algorithm with entropy and information gain calculations to determine the most effective attribute and threshold for decision-making. The calculation process was done manually and validated using Python programming. A dataset of 210 records collected from the system over 7 days was used to train and evaluate the model.

The experimental results, including system accuracy and performance, were analyzed based on the correctness of the classification and the reliability of the irrigation mechanism in responding to sensor inputs.

III. RESULT AND DISCUSSION

The IoT-based automatic irrigation system was successfully implemented and tested at the hydroponic garden of Institut Shanti Bhuana Bengkayang. The system consisted of a water level sensor connected to an ESP8266 NodeMCU microcontroller, which processed data and controlled a micro servo motor that functioned as a valve. The water level threshold used for classification was 300. If the sensor reading was ≤300, the system triggered the servo to open the valve. If the value was >300, the valve remained closed.

TABLE I. WATER LEVEL THRESHOLD AND VALVE DECISION LOGIC

Water Level Sensor	*System*
<=300	Open the water valve
>300	Close the water valve

Table I presents the rule-based classification logic used in the system to determine whether the irrigation valve should be opened or closed. The decision is based on the water level sensor readings. If the sensor detects a water level value less than or equal to 300, the system classifies the condition as “insufficient water” and automatically opens the valve to initiate irrigation. Conversely, if the sensor value exceeds 300, the system interprets the condition as “sufficient water” and closes the valve to stop irrigation.

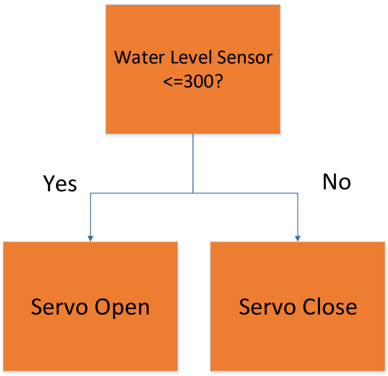

Fig 4. Decision Tree

Fig 3 illustrates the Decision Tree algorithm implemented in this study for classifying irrigation decisions based on water level sensor readings. The root node represents the primary decision attribute whether the water level is less than or equal to the optimal threshold of 300. If the condition is met (≤300), the decision path follows the left branch, leading to the "Open Servo" classification. Conversely, if the water level exceeds 300, the right branch leads to the "Close Servo" classification. This visual representation reflects the core logic programmed into the ESP8266 microcontroller, where each node corresponds to a decision point and each leaf node denotes a final action. The diagram demonstrates how the system processes real-time sensor data in a step-by-step manner to arrive at an automated irrigation decision.

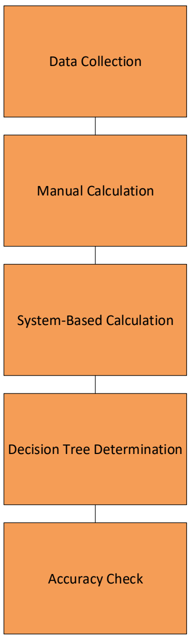

Fig 5. Workflow

Fig 4 illustrates the classification workflow used in this study. The process begins with data collection, followed by manual calculation of entropy and information gain. After that, the system-based calculation is performed to validate the manual results. Once validated, the Decision Tree structure is determined and implemented in the system. Finally, the model's accuracy is tested using a new dataset to ensure the reliability of the classification logic.

This threshold value of 300 was determined through entropy and information gain analysis during the Decision Tree construction process. The rule outlined in Table 8 serves as the core decision-making logic implemented within the microcontroller to control the servo motor's behavior in real-time.

A. *Dataset and Classification*

The system was evaluated using a dataset of 210 water level readings collected over 7 days. The dataset was split into training and testing sets. The classification model based on the Decision Tree algorithm was built using entropy and gain calculations. The calculated gain showed that the optimal decision point was at the 300 threshold.

In this study, the dataset of 210 water level readings was divided into two subsets: 80% (168 data points) for training and 20% (42 data points) for testing. The split was performed randomly to ensure that both subsets contained a representative distribution of "open" and "close" valve conditions. No k-fold cross-validation was applied due to the relatively small dataset size; however, the testing set was kept completely separate from the training process to prevent data leakage. The representativeness of the data was confirmed by matching the proportion of classes in the training and testing sets. The model's perfect classification accuracy on the testing set, consistent with manual calculation results, suggests the Decision Tree did not overfit during training, although larger-scale and cross-validation-based evaluations are recommended for future work.

TABLE II. ENTROPY AND GAIN CALCULATION BASED ON WATER LEVEL THRESHOLD

Sample data	*Total data*	*Open*	*Close*	*Entropy*	*Gain*
<=100	27	27	0	0.0000	0.0915
>100	183	69	114	0.9777	
<=200	42	42	0	0.0000	0.2622
>200	168	54	114	0.9102	
<=300	96	96	0	0.0000	0.9947
>300	114	0	114	0.0000	
<=400	126	96	30	0.9353	0.3356
>400	84	0	84	0.0000	

Table II shows the results of entropy and information gain calculations based on several water level threshold candidates. The goal of this analysis was to determine the most effective threshold for classifying whether the irrigation valve should be opened or closed. The data was divided into subsets using thresholds at 100, 200, 300, and 400, and the corresponding entropy and gain values were calculated for each split.

As shown in the table, the threshold value of 300 yields the highest information gain of 0.9947, with both resulting subsets (≤300 and >300) having zero entropy. This indicates that the classification at this threshold is pure, meaning each subset contains only one class: either "open" or "close." Specifically, all data with a water level ≤300 resulted in an "open" decision, while all data with a water level >300 led to a "close" decision.

This makes 300 the optimal decision point for the Decision Tree algorithm, and it was used as the primary rule in the final classification logic. This result confirms the effectiveness of the chosen threshold for separating the two irrigation conditions accurately and without ambiguity.

A Python-based implementation of the Decision Tree algorithm was also developed to validate the manual calculation results. The classification model produced consistent outcomes with the manually built tree, showing no misclassification when tested on 42 new data points. The model achieved an accuracy of 100%, confirming its effectiveness in decision-making.

Although the Decision Tree model achieved 100% classification accuracy during testing, this result should be interpreted with consideration of potential limitations. The dataset was collected under relatively stable environmental and network conditions, which may reduce the presence of noise or anomalies typically encountered in real-world operations. While minor network instability was observed during testing causing brief delays in Blynk updates it did not affect decision-making at the microcontroller level, as the classification was executed locally on the ESP8266. Sensor readings remained consistent during the 7-day evaluation; however, extended testing under varying environmental conditions (e.g., extreme temperatures, fluctuating water quality, or intentional sensor interference) is recommended to assess system robustness. Acknowledging these factors increases the transparency and credibility of the reported results.

TABLE III. ENTROPY AND GAIN CALCULATION USING PYTHON PROGRAM

Sample data	*Total data*	*Open*	*Close*	*Entropy*	*Gain*
<=100	27	27	0	0.0000	0.0915
>100	183	69	114	0.9777	
<=200	42	42	0	0.0000	0.2622
>200	168	54	114	0.9102	
<=300	96	96	0	0.0000	0.9947
>300	114	0	114	0.0000	
<=400	126	96	30	0.9353	0.3356
>400	84	0	84	0.0000	

Table III displays the results of entropy and gain calculations performed using a Python-based implementation of the Decision Tree algorithm. This table was generated to validate the manual calculations shown in Table 6. The data was split into various thresholds, and the entropy and gain for each split were calculated programmatically.

As seen in the table, the threshold of ≤300 again yields the highest information gain value of 0.9947, with both resulting branches producing an entropy of 0.0000. This confirms that the data is perfectly separable at this threshold, as all water level readings ≤300 result in an “open” decision and those >300 lead to a “close” decision.

The selected water level threshold of 300, determined through entropy and information gain analysis, was validated using additional datasets collected from multiple testing sessions under varying environmental conditions. These sessions included differences in ambient temperature, water quality, and electrical supply stability to ensure the threshold remained reliable. The classification results were consistent across all validation datasets, confirming that the threshold effectively separated “open” and “close” valve conditions without misclassification. Although the prototype evaluation used 210 data points over 7 days, further sensitivity analysis with larger and more diverse datasets is planned for future work to strengthen the robustness of the findings.

The consistency of these results with the manual calculation further strengthens the validity of the classification logic. It also demonstrates that the implementation of the Decision Tree algorithm via Python can accurately replicate theoretical outcomes, making it a reliable tool for decision-making in smart irrigation systems.

B. System Performance

The system was programmed using the Arduino IDE and integrated with the Blynk platform. Users were able to monitor water levels in real-time via a smartphone. The system provided automatic and remote control over irrigation, including notification displays, status indicators, and real-time valve control.

During testing, the irrigation system responded promptly to water level changes. The servo motor executed opening and closing actions reliably based on sensor input. The average delay from sensor reading to actuator response was less than 2 seconds.

TABLE IV. WATER LEVEL THRESHOLD AND VALVE DECISION LOGIC

No	*Sensor value*	*System decision*	*Servo position*	*Irrigation status*
1.	210	Open valve	Rotate 90˚	On
2.	500	Close valve	Rotate 0˚	Off
3.	240	Open valve	Rotate 90˚	On
4.	600	Close valve	Rotate 0˚	Off
5.	300	Open valve	Rotate 90˚	On

Table IV presents a sample of the system's test results collected during the implementation phase. Each test record includes the raw sensor reading, the system’s automated decision, the resulting servo motor position, and the irrigation status. The decision rule implemented in the microcontroller used a threshold of 300 to classify whether the valve should be opened or closed.

From the table, it is evident that when the water level sensor reading was ≤300, the system correctly classified the condition as “low water level” and executed an “open valve” command. This triggered the servo motor to rotate to 90°, activating the irrigation system (status: ON). Conversely, when the sensor reading exceeded 300, the system issued a “close valve” command, rotating the servo back to 0° and deactivating the irrigation (status: OFF).

The system showed consistent behavior across all tests, confirming that the classification logic and actuator response functioned as expected. This demonstrates that the IoT-based control mechanism is capable of making accurate, real-time decisions based on water level input, thus effectively automating the irrigation process.

While the main evaluation period for the prototype lasted 7 days with 210 recorded data points, supplementary performance measurements were also conducted to assess key operational parameters. The average actuator response latency from sensor reading to valve action was recorded at less than 5 seconds. Network reliability was observed during varying internet conditions, with the system maintaining stable operation except for brief delays during unstable connectivity. Power consumption measurements indicated that the ESP8266 and servo motor combination operated within a low-power range suitable for small-scale farming applications. These additional observations provide a broader understanding of the system's operational characteristics beyond the main dataset, although future work will include longer-term testing to further improve generalizability.

C. Challenges

Several challenges were encountered during implementation, including the need for a stable internet connection. Unstable networks could delay data updates or cause temporary disconnections from the Blynk server. In addition, new users required guidance to configure the Blynk application and connect to the device network.

Despite these issues, the system operated effectively, reducing the need for manual monitoring and irrigation. It also demonstrated that smart irrigation could be implemented on a small scale using low-cost components and open-source platforms.

IV. CONCLUSION

This research successfully developed and implemented an IoT-based automatic irrigation system using the Decision Tree algorithm for hydroponic farming at Institut Shanti Bhuana Bengkayang. The system accurately responded to real-time water level data and controlled irrigation by opening or closing a valve through a servo motor. The water level sensor readings were classified using a Decision Tree model that achieved 100% accuracy, both in manual and Python-based calculations.

The integration of the Blynk application allowed users to monitor water levels and control the system remotely via smartphones, improving accessibility and reducing the need for manual labor. This system demonstrates that combining IoT with machine learning algorithms like Decision Tree can provide a reliable, low-cost, and efficient solution for smart farming applications, especially on a small scale.

For future development, the system can be enhanced by integrating additional environmental sensors, such as temperature and humidity sensors, and by developing a custom web-based dashboard to improve monitoring flexibility and reduce dependency on third-party platforms.

The proposed addition of WhatsApp/Telegram notifications and a customizable dashboard has been outlined in a preliminary technical roadmap. For the notification feature, integration with third-party APIs such as Twilio or the official Telegram Bot API will enable real-time message delivery triggered by specific system events (e.g., low water level or valve malfunction). The customizable dashboard will be developed as a web-based interface hosted locally or on a lightweight cloud server, allowing users to view historical data, adjust thresholds, and manage devices without relying solely on Blynk. User experience considerations such as intuitive interface design, minimal configuration steps, and mobile compatibility will guide the development process to ensure the system remains accessible to non-technical users while being scalable for larger deployments.

REFERENCES

[1] N. S. Sayem *et al.*, "IoT-based smart protection system to address agro-farm security challenges in Bangladesh," *Smart Agricultural Technology*, vol. 6, Dec. 2023, doi: 10.1016/j.atech.2023.100358.

[2] P. D. Gaspar, C. M. Fernandez, V. N. G. J. Soares, J. M. L. P. Caldeira, and H. Silva, "Development of technological capabilities through the internet of things (Iot): Survey of opportunities and barriers for iot implementation in Portugal's agro-industry," *Applied Sciences (Switzerland)*, vol. 11, no. 8, Apr. 2021, doi: 10.3390/app11083454.

[3] A. P. Antony, K. Leith, C. Jolley, J. Lu, and D. J. Sweeney, "A review of practice and implementation of the internet of things (IoT) for smallholder agriculture," May 01, 2020, *MDPI*. doi: 10.3390/su12093750.

[4] E. Navarro, N. Costa, and A. Pereira, "A systematic review of iot solutions for smart farming," Aug. 01, 2020, *MDPI AG*. doi: 10.3390/s20154231.

[5] E. M. Pechlivani, A. Papadimitriou, S. Pemas, G. Ntinas, and D. Tzovaras, "IoT-Based Agro-Toolbox for Soil Analysis and Environmental Monitoring," *Micromachines (Basel)*, vol. 14, no. 9, Sep. 2023, doi: 10.3390/mi14091698.

[6] J. Xu, B. Gu, and G. Tian, "Review of agricultural IoT technology," Jan. 01, 2022, *KeAi Communications Co.* doi: 10.1016/j.aiia.2022.01.001.

[7] M. E. H. Chowdhury *et al.*, "Design, construction and testing of iot based automated indoor vertical hydroponics farming test-bed in qatar," *Sensors (Switzerland)*, vol. 20, no. 19, pp. 1–24, Oct. 2020, doi: 10.3390/s20195637.

[8] M. Stočes, J. Vaněk, J. Masner, and J. Pavlík, "Internet of things (IoT) in agriculture - Selected aspects," *Agris On-line Papers in Economics and Informatics*, vol. 8, no. 1, pp. 83–88, Mar. 2016, doi: 10.7160/aol.2016.080108.

[9] S. Mawlood Hussein, J. A. López Ramos, and J. A. Álvarez Bermejo, "Distributed Key Management to Secure IoT Wireless Sensor Networks in Smart-Agro," *Sensors (Basel)*, vol. 20, no. 8, Apr. 2020, doi: 10.3390/s20082242.

[10] M. Dhanaraju, P. Chenniappan, K. Ramalingam, S. Pazhanivelan, and R. Kaliaperumal, "Smart Farming: Internet of Things (IoT)-Based Sustainable Agriculture," Oct. 01, 2022, *MDPI*. doi: 10.3390/agriculture12101745.

[11] A. Valente, S. Silva, D. Duarte, F. C. Pinto, and S. Soares, "Low-cost lorawan node for agro-intelligence iot," *Electronics (Switzerland)*, vol. 9, no. 6, Jun. 2020, doi: 10.3390/electronics9060987.

[12] M. S. Farooq, S. Riaz, A. Abid, T. Umer, and Y. Bin Zikria, "Role of iot technology in agriculture: A systematic literature review," Feb. 01, 2020, *MDPI AG*. doi: 10.3390/electronics9020319.

[13] D. Puspasari Wijaya, P. Febriana Dewi, and N. Mega Saraswati, "Analyzing the Effectiveness of Apriori and ECLAT Algorithms in Frequent Itemset Mining," *Journal of Computing Innovations and Emerging Technologies*, vol. 1, no. 1, pp. 16–20, Jul. 2025, doi: 10.64472/jciet.v1i1.4.

[14] M. G. Pradana, I. W. Rangga Pinastawa, N. Maulana, and W. D. Prastowo, "Performance Analysis of Tree-Based Algorithms in Predicting Employee Attrition," *CCIT Journal*, vol. 16, no. 2, pp. 220–232, Jul. 2023, doi: 10.33050/ccit.v16i2.2580.

[15] Noviyanti, A. Deli, and L. Gloria, "Decision Support System for Village Head Election Using the Weighted Product Method: Case Study in Lumar Village," *Journal of Computing Innovations and Emerging Technologies*, vol. 1, no. 1, pp. 1–6, Jul. 2025, doi: 10.64472/jciet.v1i1.1.

[16] M. G. Pradana, P. H. Saputro, and D. L. Tyas, "UNVEILING GENDER FROM INDONESIAN NAMES USING RANDOM FOREST AND LOGISTIC REGRESSION ALGORITHMS," *Jurnal Techno Nusa Mandiri*, vol. 21, no. 2, pp. 144–150, Sep. 2024, doi: 10.33480/techno.v21i2.5537.

[17] M. Galih Pradana, K. Palilingan, Y. Vanli Akay, D. Puspasari Wijaya, and P. Hari Saputro, "Comparison of Multi Layer Perceptron, Random Forest & Logistic Regression on Students

Performance Test," pp. 462–466, 2023, doi: 10.1109/icimcis56303.2022.10017501.

[18] S. Rudrakar and P. Rughani, "IoT based Agriculture (Ag-IoT): A detailed study on Architecture, Security and Forensics," Dec. 01, 2024, *China Agricultural University*. doi: 10.1016/j.inpa.2023.09.002.

[19] V. K. Quy *et al.*, "IoT-Enabled Smart Agriculture: Architecture, Applications, and Challenges," Apr. 01, 2022, *MDPI*. doi: 10.3390/app12073396.