GrowSync: An IoT-Based Agriculture System for Small and Medium Farm

Geoffrey Kan Zi Yoong
School of Computer Science
Taylor's University
Subang Jaya, Malaysia
0368387@sd.taylors.edu.my

Eric Lau Lik Shien
School of Computer Science
Taylor's University
Subang Jaya, Malaysia
ericlikshien.lau@sd.taylors.edu.my

Ong Zheng He
School of Computer Science
Taylor's University
Subang Jaya, Malaysia
zhenghe.ong@sd.taylors.edu.my

Chloe Tee Rouyi
School of Computer Science
Taylor's University
Subang Jaya, Malaysia
chloerouyi.tee@sd.taylors.edu.my

Hng Qi Yean
School of Computer Science
Taylor's University
Subang Jaya, Malaysia
0364483@sd.taylors.edu.my

***Abstract*—This paper presents GrowSync, an IoT smart agriculture system that is designed to help small and medium sized farmers monitor and optimise their farming. It also integrates sensors to measure environmental conditions and using cloud platforms like AWS and Agroview for data analytics, storage and forecasting. Lastly, a mobile application developed with Flutter enables real-time monitoring and control of their farms. As a conclusion, GrowSync supports sustainable and data-driven farm management by combining sensor data, cloud intelligence and automated actuation.**

***Keywords*— Smart agriculture, IoT, precision farming, AWS, Agroview, Flutter, environmental monitoring, automation, cloud computing.**

I. Introduction

In the modern era, the agriculture sector is evolving and adapting as agriculture is facing challenges such as climate change, inefficient resources utilization and limited access to technology. This is seen in small and medium-sized farms as the farmers are often struggling to maintain the optimal environmental conditions such as soil pH level, temperature, humidity and air quality which are crucial for maximizing crop yield and ensuring sustainability. Although several unique smart farming solutions exist, they are fragmented, expensive and hard to scale making the adaption in agriculture sector difficult. Moreover, high labour demands, resource wastage and inconsistent productivity also happen due to the lack of real time monitoring and automation. As a result, a cost-effective and intelligent Internet of Things (IoT)-based system is required that can help farmers make data driven decisions and have a remote management.

II. Objectives

The main objective of this paper is to design an IoT smart agriculture system named GrowSync. The main goal is to develop a multi layered IoT architecture that do real-time monitoring, data transmission, cloud analytics and automated actuation. This system will integrate sensors to monitor key environmental factors and leveraging cloud services such as AWS and Agroview to store, visualise data and forecast important data using machine learning models. Additionally, a mobile application will be developed using Flutter that allow users to monitor and control their farm operations. All and all, the system is designed to be scalable, affordable and secure to support development in the agriculture sector.

III. Literature Review

Although there is an increasing number of IoT-based smart agriculture solutions, most of them often fall short for small and mid-sized farms. For example, although systems like FarmBot and FieldAgent offer features like automated planting or drone-based monitoring, they tend to be expensive, fragmented and hard to scale. Due to their lack of interoperability, proprietary platforms restrict system flexibility and make device integration challenging.

High upfront costs remain a significant barrier to IoT adoption in agriculture, especially for small and mid-sized farms in developing regions [1]. Farmers are often unable to afford sensors, controllers, and cloud services without external support. In addition, sensors exposed to harsh outdoor environments can degrade over time, leading to inaccurate data and poor decisions in irrigation or fertilization [2]. Many existing systems also lack built-in error checking, while minimal encryption leaves devices vulnerable to intrusion or tampering [2].

To solve the issues above, we use technologies such as LoRaWAN, MQTT and also cloud platforms like AWS and Agroview. LoRaWAN enables long-range, low-power communication which is ideal for dispersed farm sensors [3] while MQTT supports lightweight, reliable data transmission across constrained networks [4]. Moreover, the integration of cloud-based analytics, edge computing, and mobile applications built on Flutter allows users to have real-time monitoring, data visualization, and user-friendly farm control from any location [5].

Nevertheless, the lack of unified, low-cost, and user-accessible platforms continues to challenge the goal of democratizing smart agriculture. Therefore, there is a clear need for a system that integrates affordable hardware, reliable data pipelines, robust security, and an intuitive user experience, especially one that is modular and scalable for both small and mid-sized farms.

IV. Methodology

This section outlines the proposed methods for designing the GrowSync IoT smart agricultural system, which provides a detailed explanation on hardware components, software platform, communication protocols, data management, and mobile app functionalities.

A. Hardware Components

1) LSP01 LoRaWAN Soil pH Sensor: The sensor monitors soil pH and moisture levels with high accuracy by using the durable IP68-rated waterproof probe and stable AgCl and metal electrodes. It also uses low-power LoRaWAN communication which is perfect for remote farms and it has 5 years of operations on an 8500mAh battery [6].

2) DHT22 Temperature and Humidity Sensor: This sensor is selected due to its low-cost, reliability and offering a wide temperature sensing range from -40°C to 80°C with ±0.5°C accuracy. It has a 0-100% relative humidity (RH) with 2-5% of difference of accuracy and it is selected for simple integration with the microcontroller and low power consumption making it ideal for long-term deployment [7].

3) MQ135 Gas Sensor: This sensor is chosen for its affordability and its ability to detect a wide variety of harmful gases making it ideal for greenhouses and open field environments. It can output analogue voltage based on the gas concentration value and works well with the ESP32 microcontroller for monitoring purposes [7]. Besides, the sensor operates at 5V and does not consume too much electricity and can operate immediately as it only requires a preheating time of 24–48 hours for it to stabilize its readings.

4) HC-SR501 Passive Infrared Motion Sensor: This PIR sensor is used for motion detection which was chosen for its wide detection range of 110 degrees and a range of up to 7 meters. This makes it ideal for ensuring safety of greenhouses, open-field and agriculture storage. Moreover, the sensor is energy efficient as it operates on 5V–20V and consume little to no energy in standby mode [8].

5) U-blox NEO-6M GPS Module: This GPS module is chosen because it is compact and reliable which can be used for location tracking and field mapping. The sensor is also perfect for the system as it has an accuracy of up to 2.5m and uses low power hence making it a very capable module to use for a long time [9]. Furthermore, it has a feature called the Assisted-GPS (A-GPS) and a 5 Hz data refresh rate which can ensure reliable performance in obstructed rural conditions.

6) ESP32 Microcontroller: The ESP32 microcontroller is used for its built-in Wi-Fi and Bluetooth features which are crucial for wireless communication and real-time monitoring functions. The microcontroller has a dual-core 32-bit LX6 processor and 48 GPIO pins allowing it to handle multiple sensor inputs such as the DHT22 and MQ135 sensors. Furthermore, it also has an energy saving sleep mode where it will use less energy when not active making it ideal and suitable for a long duration of deployment [10].

7) Arduino Mega 2560 Microcontroller: This microcontroller will be used as a supporting microcontroller due to its extensive input and output capabilities, in addition to strong compatibility with different sensors and actuators used in the GrowSync system. Another reason that this Arduino module is used is that it has 54 digital pins and 16 analogue pins inputs making it ideal for a variety of tasks. It also supports PWM outputs making precise control of devices like irrigation valves easy [11].

8) GDSTIME 12V 80mm Fan: A ventilation fan is used to normalize the air temperature and get rid of excessive pollutant gases when the gas levels exceed safe thresholds.

9) JT-180A 5V DC Submersible Water Pump: A water pump controls the humidity and misting system to increase air moisture when levels fall below the desired range.

10) Adafruit 1150 12V DC Peristaltic Pump: A peristaltic pump is used for soil pH level control and injecting acidic or alkaline solutions based on sensor feedback gathered on the LSP01 pH sensor. This ensures that the precise level of soil pH level is achieved by using this actuator.

11) JFC-1 Solenoid Valve: A solenoid valve is used to control irrigation lines to enable efficient water distribution based on gathered soil moisture data.

12) 16x2 Liquid Crystal Display (LCD): A display offers real time feedback of temperature, humidity, soil pH, gas concentration and irrigation status, making it ideal for on-site monitoring in areas with limited connectivity [12].

13) Piezoelectric Buzzer: A buzzer provides immediate audio alerts up to 85dB for critical conditions such as when hazardous gas levels is high, or pest is in the vicinity [13].

B. Software Platform

1) Cloud Platform – Agroview: Agroview is a cloud platform based on AWS used for smart agriculture. It uses the high scalability and efficiency of AWS to carry out tasks that require a lot of CPU and GPU power like mapping crop health, estimating crop height, and detecting gaps between trees. It reduces manual scouting and supports high-precision farming decisions. Agroview can serve various farm sizes while ensuring its reliable performance and cost-effectiveness using AWS's flexible infrastructure. In addition, its ability to integrate with other AWS services, namely Amazon S3, Amazon IoT Core, Amazon RDS and AWS TimeStream allows room for scalability. It serves as an ideal cloud platform for large-scale, data-driven agricultural operations [14].

2) Mobile App Development Framework – Flutter: Flutter which is developed by Google is widely recognized for its performance and flexibility across platforms. It enables programmers to create a single Dart codebase and publish apps for web, iOS, Android and embedded platforms. Flutter's hot reload feature is crucial for smart agriculture as it allows for real-time user interface modifications when incorporating dynamic sensor data. Furthermore, its adaptable widget system enables developers to design user-friendly monitoring and control dashboards. Its embedded device compatibility also makes it appropriate for IoT gateways or on-field tablets. These features are suitable for building interactive agriculture applications that require real-time responsiveness.

C. Communication Protocols

1) ZigBee: ZigBee is used for short and medium range data transmissions. Short range transmissions involve sensors which are placed close to the central microcontroller. Therefore, Zigbee is used due to its low latency of around 20 to 30ms and 2MHz bandwidth, which is suitable for transmitting small sensor data. Medium range transmissions apply to sensors which are distributed across the field. Thus, a Zigbee mesh network is used, where each sensor acts as a node which can relay data to the central microcontroller [15]. Although the latency is increased, the trade-off is justified due to being able to cover a larger area. Additionally, Zigbee consumes a low amount of power, allowing it to be widely deployed across the area.

2) LoRaWAN: LoRaWAN is used for long range transmissions, which involves agricultural zones located far away from the central building. As the amount of data sent is very small, LoRaWAN is ideal due to its narrow bandwidth range of 125 – 500KHz. The collected data also doesn't require constant monitoring with a high refresh rate, making LoRaWAN's latency of approximately 10 seconds suitable for the task. Its very low power consumption allows it to be used in a large-scale across long distances, without the need for frequent battery replacements [16].

3) MQTT: Message Queuing Telemetry Transport (MQTT) is used for data transmissions to the cloud due to its reliability and scalability. It uses a streamlined data transmission method, minimizing the loss of power and preserving network performance during the transmission process [4]. Wi-Fi is used as the communication medium due to its widespread usage for internet connectivity and having a bandwidth of 22MHz at 2.4GHz frequency band, allowing large amounts of data to be transferred at once. The minor latency of approximately 50ms is negligible since it doesn't affect the normal operations of the IoT system.

4) HTTPS: Hypertext Transfer Protocol Secure (HTTPS) is used for communications between the cloud and mobile application, with Wi-Fi as the communication medium. HTTPS is selected as it is a widely accepted protocol based on the request/reply model, suitable for the client/server architecture in the GrowSync system. In this process, the mobile application sends a request message to the cloud, which processes it and returns a response message back to the mobile application. HTTPS uses TLS to encrypt data and establish a secure communication channel between the cloud and the mobile application [17].

D. Data Management

1) Storage Technologies

a) Amazon TimeStream: Time-series sensor data are stored as structured data and transferred continuously to the cloud for storage and real-time processing, which requires stream processing. This approach ensures that time-sensitive data can be processed and responded to immediately with the appropriate actuators. Therefore, Amazon TimeStream will be used as the database technology to store sensor data for time-series analytics [18]. It can effectively and efficiently store, query, and retrieve large volumes of time-series data for analysis and visualization purposes [19].

b) Amazon S3: System environment logs are stored in the form of semi-structured or unstructured data and are not updated frequently. Therefore, they will be transferred to the cloud in batches for storage in Amazon S3, which provides durable and scalable data storage *[23]*. S3's log storage capability allows for the collection and storage of environmental logs securely in tamper resistant storage, while also providing evaluating, monitoring, alerting, and auditing of access and actions in the GrowSync system [20].

c) Amazon RDS: System settings are stored as structured data and processed upon event triggers, such as when a user reconfigures a specific sensor. The settings will be stored in Amazon RDS, a relational database service that provides SQL databases for storing system settings and configurations [21].

2) Security Protocols

a) At Rest: When data is pending for transmission in the gateway microcontroller or stored in the cloud, it is encrypted using the AES-256 algorithm, which offers stronger security than the AES-128 algorithm. Although AES-256 requires more power and computational time, this trade-off is justified due to the sensitivity and importance of the collected data [22].

b) In Transit: Data transmissions between the gateway, cloud and mobile application will be encrypted using TLS, which employs the AES-256 algorithm to ensure data confidentiality and integrity during transmission. However, data transmission between sensors, actuators, and microcontrollers will only use the AES-128 algorithm since they have limited power and processing capabilities [23].

E. Mobile App Functionalities

1) Real-Time Monitoring: The mobile app provides a live dashboard that displays real-time environmental conditions and actuator status. Real-time data is streamed from AWS Timestream, ensuring minimal latency between sensor input and user interface [5]. It visualizes key parameters such as Air Quality, pH Level and motion alerts. For example, if the air quality drops below the threshold, users will see a red warning icon on the dashboard next to the "CO_2 Levels" indicator. Similarly, actuators such as misting systems, ventilation fans, and pH regulation pumps show their current status ("Active", "Standby", "Offline") in real time. This feature helps farmers to notice and respond quickly to environmental changes without being physically present.

2) Remote Control and Scheduling: The app enables users to override automatic controls and manually manage actuators from anywhere [24]. For example, users can manually control irrigation, adjust water flow, or activate specific zones based on observed moisture levels. Users can increase ventilation fan speed if the temperature of greenhouses is high or activate misting during dry weather. If CO_2 levels are low, users can activate the valve directly from the app to improve photosynthesis conditions. Deterrents such as lights or buzzers can be turned on if pest movement is detected. Lastly, users have the option to schedule automated tasks, such as creating an irrigation schedule to water Zone A every morning at 6 AM.

3) Notifications and Alerts: The GrowSync system provides real-time alerts based on sensor thresholds or system anomalies [25]. Types of alerts include environmental warnings (e.g., "Temperature exceeds 35°C" or "Soil moisture below critical threshold"), system malfunctions (e.g., "Fan not responding," or "Sensor disconnected"), and pest activity (e.g., "Motion detected near tomato greenhouse at 7:10 PM"). Lastly, the app also generates summary reports daily or weekly, highlighting key environmental conditions, actuator activities, and resource usage, enabling farmers to reflect on system performance and plan more effectively.

4) Personalization Features: The mobile app includes personalization features that allow users to modify the system to specific crop types, farm zones, and individual preferences [25]. For example, users can create and manage profiles for different crops, such as defining the temperature, humidity, and pH targets for crops like lettuce, chili, or strawberries. In addition, users can rearrange widgets, for example, placing pH levels and nutrient tank status at the top if those are high priority. Lastly, farm owners can grant access to field workers or agronomists with different roles such as view-only, control, or admin.

V. Architecture Diagram

Fig. 1 shows a multi-layered IoT architecture employed by the GrowSync system to enhance precision farming in small to medium-sized plots. In the perception layer, there are DHT22 temperature and humidity sensor, MQ135 gas sensor, LSPH01 soil pH and moisture sensor, HC-SR501 motion sensor and NEO-6M GPS sensor which continuously gather environmental data. In the gateway layer, the ESP32 microcontroller handles wireless communication via Wi-Fi, ZigBee or LoRaWAN, transmits filtered data to the cloud using the MQTT protocol and manages actuator control while the Arduino Mega 2560 acts as a backup microcontroller.

In the cloud layer, services such as AWS IoT Core, Amazon RDS, AWS Timestream, and Amazon S3 store, process and analyse real-time and historical sensor data. Agroview also provides AI-powered visualisation and decision-making support. Then, the application layer uses the Flutter framework to deliver a mobile/web dashboard for real-time monitoring, alerts and remote control. Lastly, the actuation layer responds automatically using devices like JFC-1 solenoid valves, JT-180A pumps, Adafruit 1150 peristaltic pumps, GDSTIME fans, LCDs and buzzers to regulate irrigation, pH levels, air quality and provide local alerts based on data insights or manual input.

VI. Results and Findings

This section presents the proofs of concept of the proposed smart agricultural system. The platforms used for the demonstration were Wokwi, Google Colab, and Figma.

A. Wokwi

Fig. 2 displays the sensor circuit diagram which is simulated using Wokwi. Two libraries are used, namely "DHTesp.h" for reading data from the DHT22 temperature and humidity sensor, and "LiquidCrystal_I2C.h" for controlling the 20x4 I2C LCD display. The hardware pin connections for all sensors such as the DHT22, MQ2, PIR sensors are defined, and the LCD I2C address is set to 0 x 27 with dimensions of 20 columns x 4 rows. Objects for the DHT sensor (dhtSensor) and LCD (lcd) are created, and two variables are initialized to track the current and previous state of the motion sensor for change detection.

Serial communication for debugging via "Serial.begin(115200)" is initialized using "void setup()", which also sets input/output modes for sensors and actuators. Lastly, it starts the DHT22 and LCD operations. In the main program loop, the first part checks for motion using the PIR sensor. If a motion event is detected, the state transitions from LOW to HIGH, which triggers the buzzer with a short beep of 500 Hz for 700 ms to indicate movement. Next, the ESP32 reads the temperature & humidity from the DHT22 sensor, the gas level from the MQ2 sensor and soil moisture & pH from a potentiometer simulating the soil moisture and pH sensor. It then converts potentiometer value into moisture percentage from 0 to 100% and pH level from 3.5 to 9.0.

Each sensor reading is compared to predefined ideal thresholds. If any value is outside its ideal range, a corresponding flag is set to indicate the issue. For example, temperature should be between 20°C and 30°C, humidity between 40% and 80%, and gas level below 3800. If any reading is outside these limits, we consider it a warning condition, and the LED turns on to represent automatic actuator activation such as fan, pump and valve. Besides, if the gas level is dangerously high as of more than 3800, a high-pitched buzzer is activated with 1000 Hz for 700 ms to warn local users. The system outputs all sensor data to both the serial monitor and the LCD screen, which will show four readings, namely temperature and humidity, gas level and status, soil moisture percentage, and soil pH value. A 2-second delay is added at the end of the main program loop to control refresh rates and avoid excessive updates.

B. Google Colab

Two machine learning functionalities of the mobile application are demonstrated using Google Colab, which are sensor data visualization and crop yield prediction.

1) Sensor Data Preprocessing and Visualization: The sensor data dataset used for this section is obtained from Mendeley Data, which is funded by La Trobe University [26]. Each sensor data entry is accompanied with a timestamp indicating the time of collection to allow time-

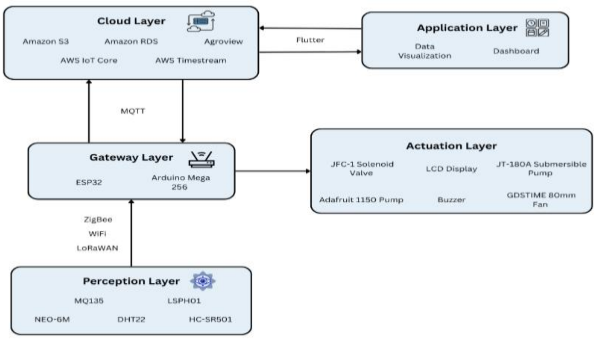

Fig. 1. GrowSync Architectural Diagram

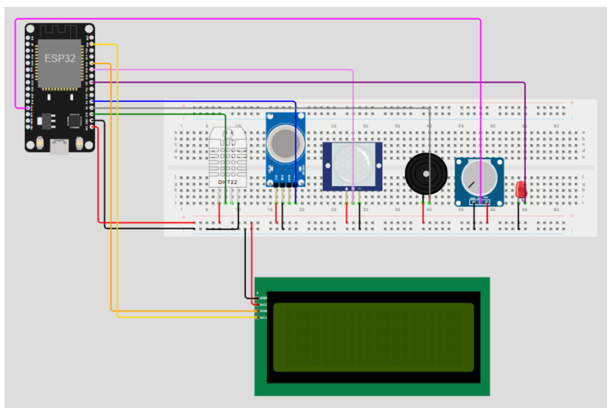

Fig. 2. Wokwi Circuit Diagram

series data analysis.

Fig. 3 presents an example of time-series soil moisture sensor data. The anomalies in the sensor data which deviate from the normal trend will be removed and imputed using linear interpolation. The dotted yellow line indicates the original anomalies in the sensor data, whereas the blue line shows the cleaned sensor data without any anomalies or inconsistencies.

An XGBoost model is trained and used to forecast the next 20 intervals of environment light intensity. The model is trained using lag features, which are created from the previous 10 intervals of each data point to allow the model to learn from past trends in sensor data [27]. According to Fig. 4, the blue line represents the past sensor data used to predict forecasted values, which is shown by the yellow line. This feature allows the system or user to take preemptive measures according to the forecasted results. For example, if the forecasted environment temperature is expected to increase, the irrigation system can be activated first to avoid watering the plants too late.

2) Crop Yield Prediction: The dataset used for this section is obtained from Kaggle, which is published by Samuel Oti Attakorah [28]. The dataset contains 10 features, with Yield_tons_per_hectare as the target feature.

A linear regression model is trained on the input features: Rainfall_mm, Temperature_Celcius, Fertilizer_Used, and Irrigation_Used, which achieved an R-squared value of 0.913, indicating that it is a reliable model for predictions. A program is created using the trained model, as shown in Fig. 5. It asks the user for specific details, and the program will predict the crop yield based on the input data. This feature allows the user to estimate the amount of crop yield based on their inputs, allowing them to satisfy their curiosity and plan in advance to reap higher crop yields.

C. Figma

Fig. 6 displays the GrowSync mobile application prototype, which focuses on the app's security, real-time alerts, and data visualization. The login interface uses Role-Based Access Control (RBAC), which ensures that users are assigned the appropriate permissions based on their roles as farm owner or farm worker. After entering valid credentials (email and password), users must complete Multi-Factor Authentication (MFA) with a one-time password (OTP), which adds an additional layer of security before accessing the system. The alert system includes real-time alerts that appear directly on the user interface, encouraging users to take immediate action. Users can choose whether to control specific actuators based on the alert. To ensure secure operations, any actuator activation requires secondary verification via OTP, reinforcing the system's multi-layered security approach. Lastly, users can navigate the app to access various functionalities such as interactive dashboards, zone maps, data analytics, user profiles, and archived notifications, all accessible through interface buttons.

VII. Discussion

While the proposed GrowSync system can tackle most issues in smart agriculture, there are several limitations that need to be addressed in future development. The first limitation is the lack of contextual awareness in automation. Current automation relies on fixed thresholds and static rules, which do not adapt to changing environmental or economic conditions. To address this, future work should focus on integrating external APIs, such as real-time weather forecasts, energy pricing data, and crop cycle databases. Additionally, implementing context-aware AI models could enable the system to dynamically adjust thresholds and offer adaptive recommendations that reflect real-world variability.

Apart from that, the lack of edge-level collaborative intelligence is also another limitation. Currently, edge devices in different zones or greenhouses operate independently and will cause missed opportunities for shared learning and situational awareness. This isolation reduces the system's ability to detect developing threats, such as the outbreak of pests or disease between zones. To overcome this, future enhancements could include federated learning to allow distributed AI model training across devices without sharing raw data. Furthermore, introducing multi-agent communication protocols between zones could enable proactive alerts, supporting early interventions and improved

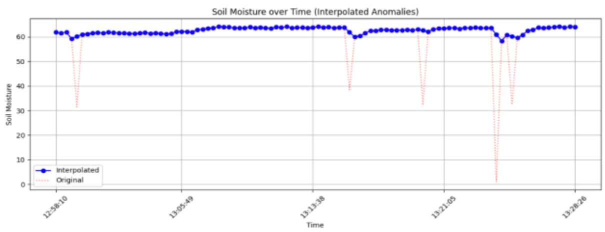

Fig. 2. Visualization of Cleaned Sensor Readings

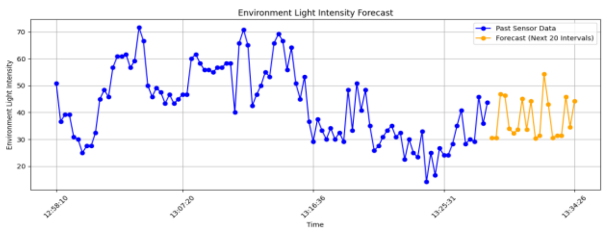

Fig. 4. Visualization of Forecasted Sensor Readings

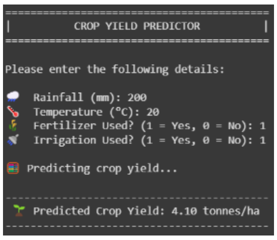

Fig. 5. Crop Yield Prediction Program

Fig. 6. Application Prototype

resource coordination across larger farms.

Lastly, the system's centralized dependency on cloud platforms introduces vulnerabilities such as increased latency and service disruptions during internet outages. For greater resilience, the system should adopt a hybrid cloud-edge architecture. Critical decisions such as activating irrigation or ventilation can be handled locally using lightweight TinyML models deployed on microcontrollers. Additionally, using local storage buffers like SQLite can support data caching and synchronization once connectivity is restored, enabling seamless offline operation during outages.

Addressing these limitations will enhance GrowSync into a more intelligent, adaptable, and collaborative smart agriculture platform, empowering farmers with smarter tools to meet the growing challenges of modern agriculture. By enabling dynamic automation that responds to real-time environmental and economic factors, the system can support more efficient and sustainable decisions. Moreover, introducing edge-level collaboration will allow zones to share insights and respond proactively to risks like pests or disease spread. Lastly, reducing cloud dependency through hybrid architectures will also improve reliability and offline resilience, especially in rural areas.

VIII. CONCLUSION

This manuscript introduced GrowSync, an IoT-based smart agriculture solution that would assist small and medium farms through real-time monitoring, automation, and data-driven decision-making. It combines environmental sensors, cloud platforms such as AWS and Agroview, and a mobile app built using Flutter to make farm management easier and improve crop yields. It supports scalable deployments through communication protocols such as ZigBee, LoRaWAN, and MQTT. While it has its strengths, GrowSync is constrained in contextual understanding, edge co-working, and cloud reliance. To address this, future research will focus on combining contextual AI, federated learning, and hybrid cloud-edge computing to enhance system flexibility and reliability. Overall, GrowSync is a viable and scalable solution for emerging agricultural issues.

REFERENCES

[1] W. Tao, L. Zhao, W. Guangwen and L. Ruobing, “Review of the internet of things communication technologies in smart agriculture and challenges. Computers and Electronics in Agriculture, 189, 106352–106352,” 2021. [Online]. Available: https://doi.org/10.1016/j.compag.2021.106352.

[2] A. Rettore, E. Silva and C. Pessoa, “Security challenges to smart agriculture: Current state, issues, and future directions. Array, 8, 100048–100048,” 2020. [Online]. Available: https://doi.org/10.1016/j.array.2020.100048.

[3] J. Arshad, M. Aziz, A. A. Al-Huqail, M. H. Muhammad, A. U. Rehman and M. Shafiq, “Implementation of a LoRaWAN based smart agriculture decision support system for optimum crop yield. Sustainability, 14(2), 827,” 2022. [Online]. Available: https://doi.org/10.3390/su14020827.

[4] T.-C. Hsu, “Designing a secure and scalable service agent for IoT transmission through blockchain and MQTT. Applied Sciences, 14(7), 2975,” 2024. [Online]. Available: https://doi.org/10.3390/app14072975.

[5] f. Services., “Mobile app development for IoT: Best practices. StudioLabs,” 17 February 2025. [Online]. Available: https://www.studiolabs.com/mobile-app-development-for-iot-best-practices/.

[6] Dragino.com, “LSPH01 -- LoRaWAN Soil pH Sensor,” 2024. [Online]. Available: https://www.dragino.com/products/agriculture-weather-station/item/184-lsph01.html.

[7] W. A. Devanannd, R. D. Raghunath, A. S. Baliram and K. S. Kazi, “Smart Agriculture System Using IoT. Academia.edu,” 12 Janauary 2020. [Online]. Available: https://www.academia.edu/download/61762899/IJIRT147761_PAPER2020 0112-36455-1dz4prw.pdf.

[8] H.-S. Datasheet., “Passive Infrared Motion Detector Sensor Module. [PDF],” 2013. [Online]. Available: https://www.mpja.com/download/31227sc.pdf.

[9] u-Blox, “NEO-6 series,” 25 June 2015. [Online]. Available: https://www.u-blox.com/en/product/neo-6-series.

[10] E. Systems, “ESP32 Series Datasheet Version 4.9. [PDF],” 2020. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet _en.pdf.

[11] Arduino, “Arduino Mega 2560 Rev3 Datasheet (A000067). [PDF],” 2015. [Online]. Available: https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf.

[12] T. Rajesh, Y. Thrinayana and D. Srinivasulu, “IOT BASED SMART AGRICULTURE MONITORING SYSTEM. In International Research Journal of Engineering and Technology,” 2020. [Online]. Available: https://www.irjet.net/archives/V7/i3/IRJET-V7I3346.pdf.

[13] M. Electronics, “Piezoelectric Sounder PS1240P02BT Datasheet. Mouser Electronics. [PDF],” 2018. [Online]. Available: https://www.mouser.com/datasheet/2/400/ef532_ps-13444.pdfhttps://www.mouser.com/datasheet/2/400/ef532_ps-13444.pdf?srsltid=AfmBOoqLO3iSlCHqyetoiHgd2LHYDq8n.

[14] A. Soussi, E. Zero, R. Sacile, D. Trinchero and M. Fossa, “Smart Sensors and Smart Data for Precision Agriculture: A Review. Sensors, 24(8), 2647,” 2024. [Online]. Available: https://doi.org/10.3390/s24082647.

[15] E. Navarro, N. Costa and A. Pereira , “A Systematic Review of IoT Solutions for Smart Farming. Sensors, 20(15), 4231,” 2020. [Online]. Available: https://doi.org/10.3390/s20154231.

[16] M. O. Ojo, I. Viola, M. Baratta and S. Giordano, “Practical Experiences of a Smart Livestock Location Monitoring System Leveraging GNSS, LoRaWAN and Cloud Services. Sensors, 22(1), 273,” 2021. [Online]. Available: https://doi.org/10.3390/s22010273.

[17] J. Dizdarević, F. Carpio, A. Jukan and Masip-Bru, “A Survey of Communication Protocols for Internet of Things and Related Challenges of Fog and Cloud Computing Integration. ACM Computing Surveys, 51(6), 1–29,” 2019. [Online]. Available: https://doi.org/10.1145/3292674.

[18] G. Kumar Sinha, “Sensor Data Analytics for Optimized Methane Leak Detection and Mitigation. International Journal of Science and Research (IJSR), 13(4), 223–231,” 2024. [Online]. Available: https://doi.org/10.21275/sr24330010938.

[19] Rashmi R, H. Prahalad, H. S. Kumar and A. Holsok, “Exploring Digital Twins for Plant Growth Monitoring,” 2023. [Online]. Available: https://doi.org/10.1109/csitss60515.2023.10334087.

[20] AWS, “Guidance for Log Storage on AWS. Amazon Web Services, Inc,” 2025. [Online]. Available: https://aws.amazon.com/solutions/guidance/log-storage-on-aws/.

[21] I.-D. Filip, C. Iliescu and F. Pop, “Assertive, Selective, Scalable IoT-Based Warning System. Sensors, 22(3), 1015–1015,” 2022. [Online]. Available: https://doi.org/10.3390/s22031015.

[22] N. Raheja and A. Kumar Manocha, “IoT based ECG monitoring system with encryption and authentication in secure data transmission for clinical health care approach. Biomedical Signal Processing and Control, 74, 103481,” 2022. [Online]. Available: https://doi.org/10.1016/j.bspc..

[23] P. Thaenkaew, B. Quoitin and A. Meddahi, “Leveraging Larger AES Keys in LoRaWAN: A Practical Evaluation of Energy and Time Costs. Sensors, 23(22), 9172–9172,” [Online]. Available: https://doi.org/10.3390/s23229172.

[24] H. X. Huynh, L. N. Tran and N. Duong-Trung, “Smart Greenhouse Construction and irrigation control system for optimal brassica juncea development. PLOS ONE, 18(10),” 2023. [Online]. Available: https://doi.org/10.1371/journal.pone.0292971.

[25] N. Wilberforce and J. Mwebaze, “A framework for IOT-enabled Smart Agriculture. arXiv.org,” 17 January 2025. [Online]. Available: https://arxiv.org/abs/2501.17875.

[26] S. DY, “Smart Agriculture and Plant Health Monitoring using IoT. Mendeley Data, 1,” 2024. [Online]. Available: https://doi.org/10.17632/65jxyrxv7b.1.

[27] Y. Reddy, “Forecasting Using Xgboost (lag and decomposition) - Medium,” 2025. [Online]. Available: https://medium.com/@byashwanth77/forecasting-using-xgboost-lag-and-decomposition-864815bd98c5.

[28] S. O. Attakorah, “Agriculture Crop Yield. Kaggle.com,” 2024. [Online]. Available: https://www.kaggle.com/datasets/samuelotiattakorah/agriculture-crop-yi