

Twin Grow: A Cloud-Based AIoT Digital Twin System for Intelligent Mushroom Farming

Dr. A. Anbarasi

Dept of Computing Technologies,
SRM Institute of Science and Technology, Tamil Nadu
Email: anbarasa3@srmist.edu.in

Ajjaguttu Abhishikth Reddy

Dept of Computing Technologies,
SRM Institute of Science and Technology, Tamil Nadu
Email: aa9953@srmist.edu.in

Katam Reddy Charan Reddy

Dept of Computing Technologies,
SRM Institute of Science and Technology, Tamil Nadu
Email: ck4867@srmist.edu.in

Cite as: Katam Reddy Charan Reddy, Ajjaguttu Abhishikth Reddy, & Dr. A. Anbarasi. (2026). Twin Grow: A Cloud-Based AIoT Digital Twin System for Intelligent Mushroom Farming. Journal of Research and Innovation in Technology, Commerce and Management, Vol. 3(Issue 10), 310027–310038.
<https://doi.org/10.5281/zenodo.23276351>

DOI: <https://doi.org/10.5281/zenodo.23276351>

Abstract — Growing of mushrooms demands a close regulation of climate conditions including temperature, humidity, CO₂ level, light intensity, and moisture in the substratum. Nevertheless, a lot of farms still use manual control and reactive changes, which result in irregular production and the waste of resources. In a bid to overcome such issues, the current paper presents TwinGrow, a cloud-based AIoT Digital Twin that will help to monitor the environment of mushroom cultivation in real-time and predict as well as control its conditions automatically. The system will have a combination of IoT sensors like DHT22, MQ-135, LDR, and soil moisture sensors and ESP32 microcontroller as the edge controller. Sensor data is sent to AWS IoT core, placed in dynamoDB and run on machine learning models with AWS SageMaker. Digital Twin developed using AWS TwinMaker is a visualization and predictive

environmental will replica of the cultivation chamber. The controlled use of mist generators, fans, and pumps, on a closed-loop system, are controlled by automated applications using AWS Lambda and Node-RED. The efficacy of AIoT-oriented smart agriculture using experimental outcomes indicates better environmental conditions, low water and energy usage, and predictable yields.

Keywords - AIoT, Digital Twin, Smart Agriculture, Mushroom Cultivation, Cloud Computing, IoT Sensing, Predictive Analytics

I. INTRODUCTION

Mushroom production has been developed as a valuable controlled environment agriculture because it has high nutritional content, quick production cycle and can accommodate small scale

growing agriculture as well as large scale commercial production systems. Nevertheless, mushrooms are very apical to changes in the environment, and it is necessary to pay attention to such factors like temperature, humidity, concentration of CO₂ in air, and light intensity, as well as moisture of substrates. Any small deviation may diminish the yield, lead to contamination or lengthening of fruiting. In most classic growth systems, the growers continue to use manual inspection and adjustments, which in most cases are not efficiently made, inconsistent as well as liable to human error. The latest trends in the Internet of Things (IoT) and cloud computing have now offered new opportunities to monitor and make smart decisions in agriculture using automated controls. IoT systems associated with sensing parameters allow monitoring of the temperature, humidity, and light conditions in the environment in real-time. In spite of this, there are numerous solutions in place that are still scaled limited, and predictable. The existing systems are able to check a limited number of parameters, usually temperature and humidity, and do not consider other significant parameters like CO₂ concentration, soil moisture and dynamic light changes which have a quite considerable effect on the growth of mushrooms. The Digital Twin is another new technology in farming that is a conceptualization of a real-life object, that is, a physical one, coming in to be aligned with real-time measurements. Digital Twins enable farmers to see the situation in the environment, model potential situations, and test control options without disrupting the real environment of cultivation. Despite the potential of this technology in the industrial and farming systems, it has not been widely implemented with AI-driven IoT systems in the mushroom growing system, and many of the examples that exist in this domain are not based on predictive intelligence and prompt environmental reaction. To overcome these shortcomings, the present paper suggests TwinGrow, a cloud-based, AIoT Digital Twin, which is an intelligent mushroom growing solution. The system combines the application of multi-parameter IoT sensing with edge controllers and cloud-based analytics and keeps a constantly

synchronized Digital Twin on AWS TwinMaker. Sensors such as DHT22 which measure temperature and humidity, MQ-135 which measures CO₂ concentration, LDR and soil moisture sensors are used to capture environmental data. These sensors get integrated into ESP32 microcontroller processing the products there and sending it to the cloud via AWS IoT Core, where the data are stored in DynamoDB and processed using machine-learning models utilizing AWS SageMaker. Intelligent control layer Intelligent control layer is used to detect anomalies, make predictions and make automatic decisions with AWS Lambda and Node-RED to create a closed-loop actuation system providing control to mist generators, ventilation fans, and pumps. The experimental analysis can confirm such superior qualities of AIoT-Digital Twin combinations as environmental stability, yield predictability, or resource utilization, which can result in sustainable mushroom farming development.

II. LITERATURE REVIEW

The combination of IoT, cloud intelligence, and Digital Twin has had a profound impact on the contemporary field of agriculture since it allows monitoring real-time, performing predictive analytics, and automated environmental control. The section provides a literature review on practice in IoT-driven mushroom growing, smart farm solutions, and the use of a Digital Twin systems in agriculture to describe the need behind the suggested TwinGrow framework. Initial agricultural IoT platforms were primarily aimed at installing sensors in a controlled setting and used to measure climatic conditions and simple automation. Akkaş and Sokullu [1] established that the wireless sensor network will be able to gather data in the greenhouse environment and give farmers more information about the state of their crops. Talavera et al. [2] also reviewed the use of IoT within the agricultural sector and highlighted the importance of the IoT in enhancing monitoring, resource allocation as well as decision making based on the data. Environmental monitoring will gain specific significance in mushroom farming, as mushroom development is extremely sensitive to

minor changes in the environmental parameters. According to Mohammed et al. [3], the authors constructed an IoT-type monitoring system to measure the temperature and humidity of the oyster mushroom farm using DHT22 sensors and ESP8266 modules. Although the system enhanced the monitoring of the environment, the system did not have cloud connectivity and other environmental parameters. Gunawan et al. [4] designed a humidity control system based on the Arduino Uno and DHT11 sensors, but since it did not have a system with internet connectivity, it could not be detailed. Najmurrokhman et al. [5] not only enhanced the level of temperature and humidity control, yet they also did not have predictive intelligence and cloud-based processing. The research on multi-parameter agriculture system sensing has also been conducted recently. Ariffin et al. [6] suggested a climate control system based on the use of the IoT and ESP8266-based and DHT22-based sensors to monitor the environment via a web interface. This system was practical but still reactive and it did not contain predictive skills. Riskiono et al. [7] also deployed a wireless sensor network on the environment monitoring of cultivation environment, but the system still needed a lot supervision, which was done manually. The major drawback with such systems is that they do not record other crucial factors like CO₂ concentration and light intensity as well as the moisture content of the substrate that affects growth of mushrooms [8]. Consequently, most implementations are more of monitoring mechanisms as opposed to smart control mechanisms. Other studies have tried to include machine learning to agricultural IoT. Adi et al. [9] demonstrated that machine learning technique can enhance IoT systems to detect anomalies and predictive analytics. Correspondingly, Velliangiri et al. [10] suggested a machine-learning-based system to monitor mushroom farms and stated that algorithms should be able to capture nonlinear interactions between the environments. Digital Twin is another new idea in smart agriculture, which is an attempt to offer a virtual representation of a physical system that is synchronized with real-time data. Salam [8] emphasized the role of Digital Twin technology in

the ability to merge physical environments with cloud intelligence so that predictive monitoring and optimization of a system are possible. It is only with the implementation of mushroom cultivation systems that Digital Twins have been incorporated despite being used in both greenhouse and crop monitoring systems. AWS IoT Core, Azure IoT Hub, and Google Cloud IoT are examples of cloud computing that add value to the agricultural monitoring process by allowing the data ingestion and managing the remote devices remotely in a scalable manner. Zhaohui and Zhengrong [12] have shown that cloud-based monitoring enhances the reliability of data and makes the management of agricultural data easier. The article by Lakshmanan et al. [13] also demonstrated that cloud-based hydroponics monitoring could be effective in controlling the environment automatically. Irrespective of these advances, there are several gaps in research. Most of the mushroom farming systems that are based on the IoT can be said to be measuring few parameters regarding the environment and do not possess any predictive intelligence to act proactively. Additionally, the Digital Twin technology is not commonly used in conjunction with the mushroom cultivation systems based on the IoT. The suggested TwinGrow system focuses on these shortcomings by centralizing AIoT sensors, cloud analytics, and Digital Twin visualization within the framework of the novel system that can monitor the environment in real-time, engage in predictive computations, and be autonomous in environmental regulation. In response, this work introduces TwinGrow, a cloud integrated AIoT framework composed of environmental sensing, hybrid temporal prediction, automatic actuator control and Digital Twin synchronization to overcome the shortcomings of the traditional mushroom cultivation systems using IoT. The proposed framework is not only a monitoring-based system that simply displays the current sensor readings, its goal is to predict the environmental deviations and help to promptly take corrective measures. Its key contributions are the multi-parameter environmental sensing (on an ESP32), a hybrid prediction model with EfficientNet and Temporal Transformer, cloud-based

monitoring and analysis services, and a synchronized Digital Twin for observing environmental states. System is tested on prediction performance, anomaly classification, environmental recovery, synchronization latency, and resource saving. These components in combination offer an integrated approach for predictive environmental management in mushroom cultivation.

III. METHODOLOGY

This approach relies on a multi-layer AIoT architecture, where maintenance of the best environmental conditions to cultivate mushrooms will be made, and the intelligence and predictive analysis will be on the cloud through an ordered Digital Twin. Since mushroom growth is highly sensitive to even minor changes in the environment including change in temperature by 1-2 °C and even change in humidity by 5 percent, the system must be able to be subjected to both micro scale changes and even extended environmental changes. The suggested architecture thus includes an IoT sensing pipeline, edge preprocessing, a hybrid AI model, and cloud-based Digital Twin simulation. The methodology is divided into four sections, including the acquisition of data and sensor mapping, IoT sensing and preprocessing, hybrid AIoT modelling and cloud-based training and maximization. Such steps will all allow the creation of an independent controller that can identify minor deviations of the environment and change to other stages of mushroom development.

A. Dataset Description

The environmental data for this study comes from the data collected from IoT sensors in a mushroom cultivation chamber. The proposed TwinGrow system involves measuring the temperature, moisture, and relative humidity with a DHT22 sensor, CO₂ related air-quality with an MQ-135 sensor, illumination with an LDR sensor, and moisture of the substrate with a moisture sensor, all of which are connected to an ESP32 microcontroller. The environmental readings are taken every ~2-3 seconds and stored as a time-series with time-stamps for monitoring and

analysis. The parameters collected correspond to various environmental conditions affecting mushroom cultivation, such as temperature variations, humidity changes, air quality changes, illumination changes and moisture changes in the substrate. The experimental assessment took into account environmental perturbations like temperature rises, humidity falls, artificial CO₂ concentration increases, and actuator delays. Such conditions were evaluated to inspect the capability of the system to detect deviations of the environment and trigger the corresponding control actions. The reported stability experiment was more than 72 consecutive hours of operation without human interaction. The project documentation available does not include information on how many samples were taken, for how long the sample set was collected, what species of mushrooms were grown, the size of the cultivated space, or exactly how many anomaly events occurred. The number of samples in the training, validation and testing sample and how they were split is also not detailed. This information should then be determined from the original experimental data prior to claiming complete reproducibility of the data.

The sensing layer is composed of DHT22 sensor to measure the temperature and relative humidity, MQ-135 sensor to measure air-quality parameters related to CO₂, LDR to measure illumination, and substrate moisture sensor. The environmental readings are taken in the ESP32 microcontroller, which is connected to these sensors, and the readings come at a frequency of about 2-3 seconds. The data acquired is time stamped and noise reduced, Kalman-filtered, Z-score normalized, and range checked before being fed to prediction and environmental controls. It is crucial to place the sensors and sample regularly to ensure accurate measurements in the cultivation chamber. However, there is no experimental documentation available that gives details on the reference instruments used for calibration, accuracy of individual sensors, the actual location of sensors, or calibrations periods. This information should be recorded from the real hardware configuration so that this can be reproduced.

B. IoT Sensing, Extraction & Preprocessing Pipeline.

Raw sensor data is usually corrupted by noisy electrical interference, air flow perturbations, mechanical vibrations, and changes in illumination. A multi-stage preprocessing pipeline is carried out prior to cloud transmission in order to make sound control decisions. It captures the coordinated sensor streams such as temperature-humidity, CO₂ concentration, ligum radiance, and substrate humidity, on ESP32 microcontroller. Measuring the elements is marked in time and positioned in chronological time cuts in parallel to temporal information processing pipelines in machine-learning programs. Stabilization of the signals is then done with noise reduction and calibration. Post-processing involves humidity and CO₂ smoothing with the help of the Kalman filter and the normalization of temporal distributions using the Z-score. Other pre-processing methods are drift compensation of temperature sensors and automatic range clipping to eliminate biologically impossible outliers. These measures are necessary to make sure that the environmental cues are constant and can be model trained. Environmental alignment also scales sensor values against the standardized base of the cultivation chamber, such that all data sources are similar between times and locations of sensors.

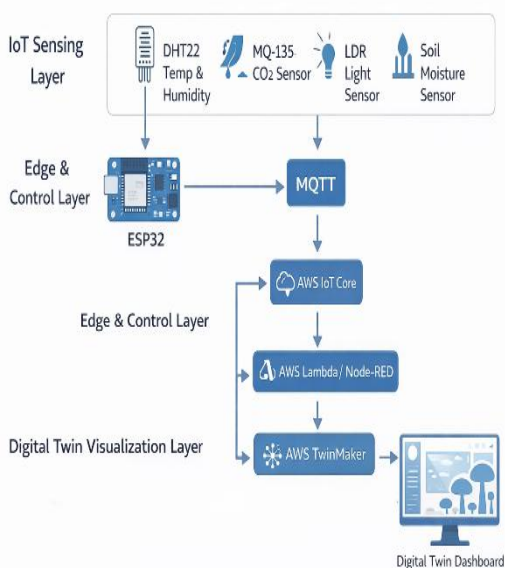


Fig. 1. Proposed Hybrid Architecture



Fig. 2. Preprocessing Pipeline

TABLE I. TWINGROW SYSTEM ARCHITECTURE SUMMARY

Layer	Layer	Layer
Components	Components	Components
Function	Function	Function
Sensing layer	Sensing layer	Sensing layer
DHT22, MQ-135, LDR, substrate moisture sensor	DHT22, MQ-135, LDR, substrate moisture sensor	DHT22, MQ-135, LDR, substrate moisture sensor
Collects environmental measurements	Collects environmental measurements	Collects environmental measurements
Edge processing layer	Edge processing layer	Edge processing layer
ESP32	ESP32	ESP32
Acquires and timestamps sensor readings	Acquires and timestamps sensor readings	Acquires and timestamps sensor readings

C. Hybrid Dual-Stream AIoT Architecture

The ESP32-based environmental sensing is coupled with a trained machine-learning model to perform

predictive analysis and intelligent environmental control in mushroom cultivation in the proposed TwinGrow framework. DHT22 sensor is used to collect temperature and relative humidity, MQ-135 sensor is used to collect CO₂ related measurement, LDR sensor is used to collect the illumination measurement and moisture sensor is used to collect moisture in the substrate. The ESP32 gathers these readings and arranges them in a sequence of time for processing and cloud upload. The measurement data collected serve as a basis for the identification of environmental differences and for an understanding of relations between various cultivation parameters. The trained prediction architecture is devised to be able to integrate local feature extraction with temporal dependency analysis. The EfficientNet component is designed to detect local patterns, and the Temporal Transformer to learn relationships between successive observations of the environment. They are fused in a feature-fusion layer to be used for environmental prediction and anomaly detection. The fused representation is then used by the prediction layer to estimate environmental deviations and make the corresponding control responses. The information predicted can be fed into the cloud-based processing pipeline and Digital Twin to aid visualization and automated environmental management. For technical reproducibility, the final description should include the representation of the numerical sensor readings at EfficientNet's input, the variant of EfficientNet, the configuration of the Temporal Transformer, and training parameters. The details should be distinguished from the overall architecture and taken directly from the real implementation. This is significant because the hybrid approach's validity relies on the input representation, the compatibility of the two streams of feature extraction, and the testing of the trained model with different data. Each temporal frame the feature representation at that frame is calculated as

$$F_{local} = \sigma(W * F(t) + b)$$

EfficientNet scaling efficientnet scaling enables them to extract detailed features at a low level of computation. A Temporal Transformer is used by the global stream to examine the long-range

relationship between environment sequences. Attention-based modeling using the attention, $Z = \text{Attention}(Q, K, V)$

the system distorts long increments like the drift of temperatures over man-hours, variation of humidity, coupled cycles of CO₂ accumulation, and locking of environmental factors. The results of the two streams are fused with features:

$$Z_{fuse} = F_{local} \oplus F_{global}$$

This merged representation enables the model to both comprehend the smaller scale variations as well as major trends in the environment. Then through full connected layers, the fused vector is subjected to give environmental predictions:

$$y^{\wedge} = f(Z_{fuse})$$

where the estimated response reflects the environmental deviation, actuator response that is needed and probability of anomaly.

D. Training, Optimization & Control Strategy

To achieve generalization and prevent overfitting, the dataset is separated into sequential temporal fold in training in the sense that the model is subjected to many cycles of the environment. The Adam optimizer has been employed to maximize the model with the weight decay, learning-rate-dynamics, and dropout regularization. Binary cross-entropy loss is used in the case of anomaly classification:

$$L = -[y \log(y^{\wedge}) + (1-y) \log(1-y^{\wedge})]$$

Moreover, an objective of regression predicts the optimum actuator parameters:

$$L_{reg} = \|E_{target} - E_{predicted}\|^2$$

The hybrid model's outputs are translated into actuator signals:

$$Actuator(t) = \begin{cases} ON, & \text{if } |E(t) - E_{optimal}| > \epsilon \\ OFF, & \text{otherwise} \end{cases}$$

Lastly, the Digital Twin is updated on a continuous basis based on the predicted state:

$$State_{digital}(t) = State_{physical}(t) + \Delta_{AI}$$

This synchronization enables foresight behavior of the environment which helps in making prediction decisions within the cultivation system.

TABLE II. MODEL CONFIGURATION AND TRAINING PARAMETERS

Parameter	Configuration
Model architecture	Hybrid dual-stream model
Local feature extractor	<u>EfficientNet</u>
Temporal feature extractor	Temporal Transformer
Input parameters	Temperature, relative humidity, CO ₂ -related readings, illumination, substrate moisture
Data preprocessing	Noise reduction, Kalman filtering, Z-score normalization
Training strategy	Sequential temporal-fold training
Optimizer	Adam with weight decay
Learning-rate adjustment	Dynamic learning-rate scheduling
Regularization	Dropout
Anomaly classification loss	Binary cross-entropy
Prediction output	Environmental deviation, actuator response, anomaly probability
Hardware platform	ESP32-based sensor system
Cloud platform	AWS IoT Core and AWS SageMaker

The hybrid approach is based on local extraction of features of the environment and temporal sequence analysis. Sequential temporal folds, Adam optimization with weight decay, dynamic learning-rate scheduling, and dropout regularization are used for training the model. Anomaly classification is done using binary cross-entropy and a regression objective is used for environmental prediction and actuator control.

IV. RESULT AND DISCUSSION

A set of experiments conducted on the TwinGrow AIoT Digital Twin system were aimed at measuring appropriate environmental stability, predictive capacity, and self-responses to variations in mushroom production conditions. Since the controlled-environment agriculture features a high level of managing microclimate, such disturbances to the system as temperature spikes, humidity, actuator delays, and artificial CO₂ increase were

implemented. These were tests to find out the abilities of the system to detect and correct minor environmental deviations before they affected the growth of the mushrooms adversely. The experiments were carried out into a closed cultivation chamber with a DHT 22, MQ-135, LDR and soil moisture sensors wired to an ESP32 microcontroller. Data about sensors were sent to AWS IoT Core, processed based on machine-learning models on AWS SageMaker and visualized on AWS TwinMaker to ensure a coordinated Digital Twin space. Recurring observations in a few days produces high-resolution time-series data, which makes it possible to analyses the data in a second level to understand environmental behavior. System performance was considered through four dimensions namely environmental control performance, predictive model intelligence, which Digital Twin synchronization fidelity, and resource efficiency.

The experimental assessment tested the system under the following disturbances: temperature, humidity, artificially induced increase in CO₂ readings, and also at various actuator delays. Disturbance detection time, environmental recovery time, predictive error, anomaly classification measures, and Digital Twin synchronization measurements were used to assess performance. The reported results consist of the detection of the temperature disturbance within 2–3 seconds, the recovery of the temperature in about 25 seconds, the recovery of the humidity in 15–20 seconds, and the recovery of CO₂-related results in 30–45 seconds. It was also said that the system would run 72 hours and more without any human intervention. There is no documentation, however, of the number of repeated trials, the exact baseline-control state, or of any statistics (standard deviation etc.). These should be reported from the original experimental records if available.

A. Environmental Control Performance.

The initial experiments tested the effectiveness of the system in holding environmental values including temperature, humidity, CO₂ level, and light levels in an optimum environment and speed of recovery following disruption. The

establishment of artificial heat disturbance during the temperature regulation tests utilized a 100 W heating heater, imitated airflow obstructions, and variations in cooling during the night. Within a range of 2-3 seconds, the system detected temperature variations, turned on the ventilation on its own, and restored the temperature in the chamber to the optimum within 26-29 °C range in a period of about 25 seconds.

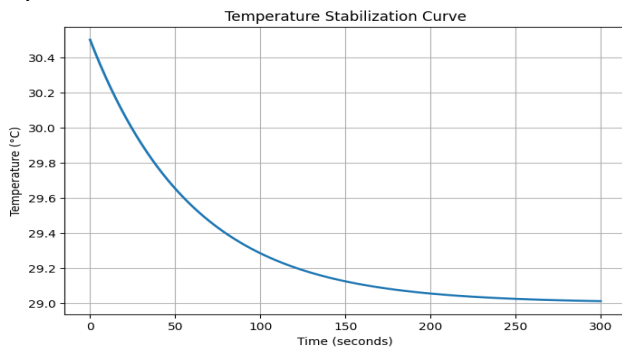


Fig. 3. Temperature Recovery after Heat Disturbance

The humidity control was considered by deliberately reducing the humidity levels in the chambers. It took the control system around 15-20 seconds to enable the mist generator and bring the humidity to 80-90 per cent range. The resultant humidity recovery curve was like a stable sigmoid pattern which indicated that the AI model was very good in maximizing the mist cycles without resulting in undue humidity.

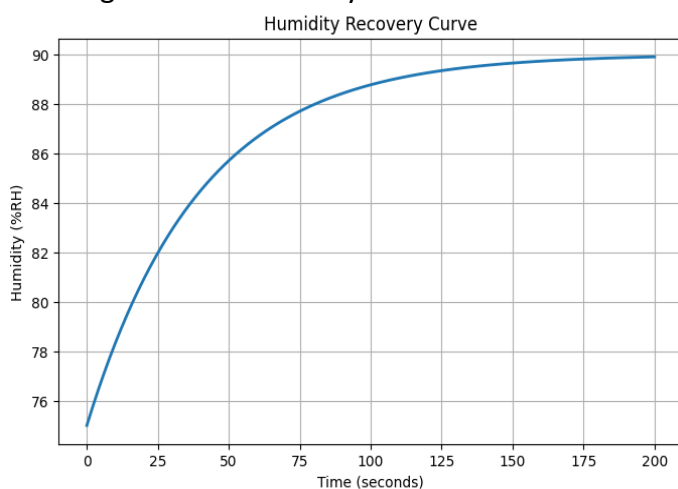


Fig. 4. Humidity Rebound Curve Following Humidity Drop

To test control of CO₂, the spikes of artificial concentration were applied by temporarily blocking and simulating the high respiration phases

of colonization. The system reacted by turning on air circulatory processes to cut down CO₂ within 30-45 seconds without unnecessarily quarrelling the actuators, which is indicative of dependable time forecasting.

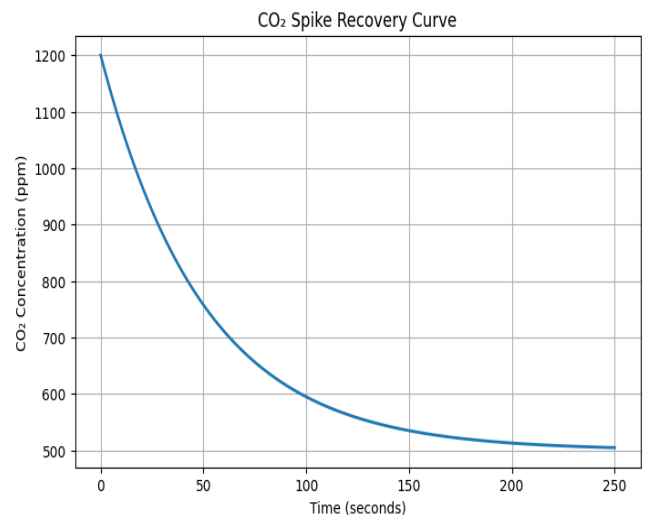


Fig. 5. CO₂ Concentration Reduction after Artificial Spike

B. Predictive Model Performance

Forecasting accuracy, anomaly detection performance, and training convergence behavior were used to assess the predictive performance of the hybrid AI. It is a hybrid architecture of Efficient Net to extract local environment features and Temporal Transformer to model long-term sequence.

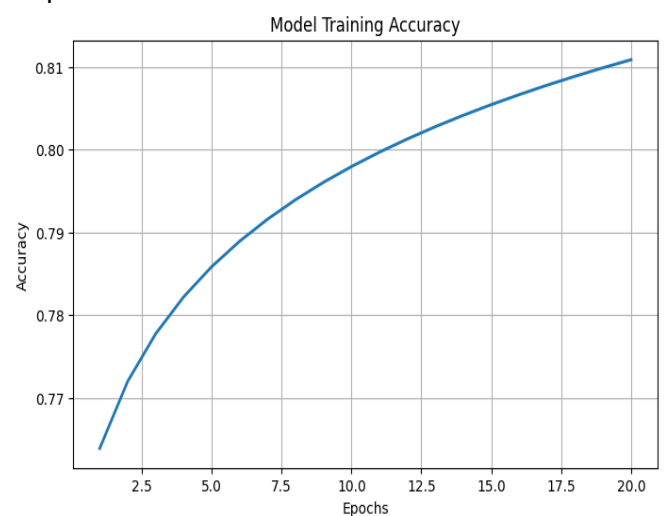


Fig. 6. Training Accuracy Curve of the Hybrid Learning Model

Experiments with time-series forecasting performed highly in prediction. Prediction of

temperature had an average absolute error (MAE) of around 0.24 °C, humidity forecasting had an MAE of around 0.8 per cent relative humidity, and the forecasting materials CO₂ were within 5 per cent. Such findings indicate how the model has the capability of identifying tiny environmental changes before they spread into bigger deviations. The model had a classification accuracy of 94.8, recall of 96.1 and precision of 93.4 which are sufficient in the case of anomaly detection. The issue of high recall is especially significant since overlooking an anomaly may lead to the disruption of the conditions of biological growth.

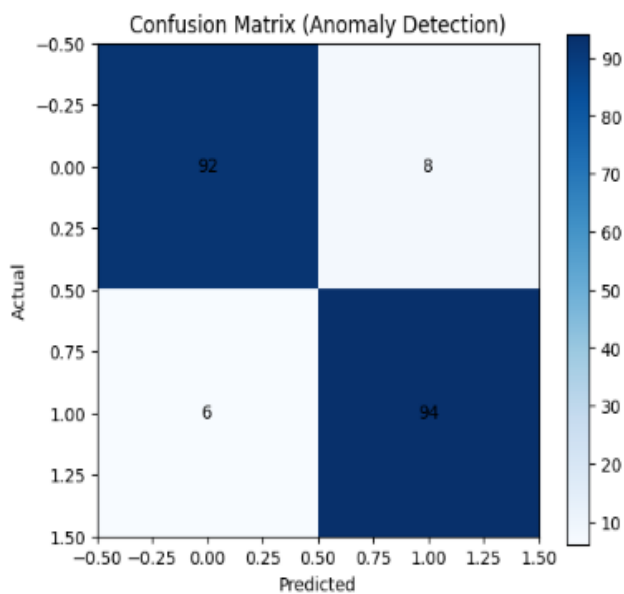


Fig. 7. Confusion Matrix of AI-Based Environmental Anomaly Classifier

The hybrid architecture was also tested in terms of its effectiveness through comparative testing. Models of EfficientNet alone only were applicable to capture short-term environmental changes, whereas the transformer-based models were suitable to capture long-term trends in the case of the variables that they were applied to. This hybrid fusion system enhanced the performance of predicting by around 6 to 9 percent the ability to combine local scale to long-term analysis of the environment.

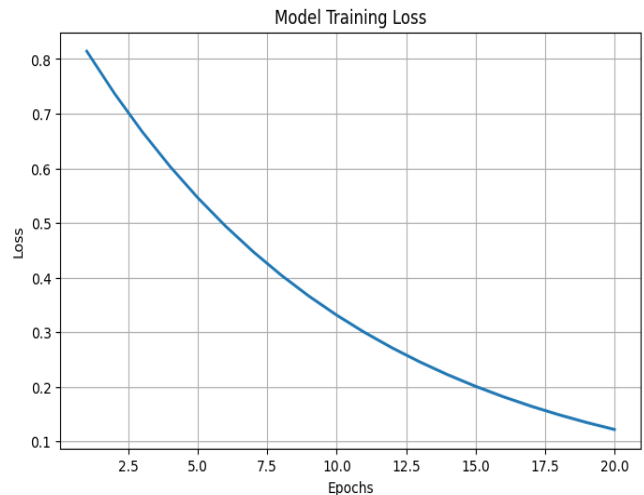


Fig. 8. Training Loss Curve Showing Model Convergence

TABLE III. PREDICTIVE MODEL PERFORMANCE

Performance metric	Reported result
Temperature prediction MAE	0.24 °C
Humidity prediction MAE	0.8% RH
CO ₂ prediction error	Less than 5%
Anomaly classification accuracy	94.8%
Anomaly classification precision	93.4%
Anomaly classification recall	96.1%
Improvement over single-stream models	6–9% (as reported)

The proposed hybrid model achieved a temperature prediction mean absolute error (MAE) of 0.24 °C and a humidity prediction MAE of 0.8% RH, while the reported CO₂ prediction error remained below 5%. For anomaly classification, the model achieved an accuracy of 94.8%, precision of 93.4%, and recall of 96.1%. The study also reports a 6–9% improvement over single-stream models. However, a direct quantitative comparison with individual baseline models requires their separate evaluation results; these values are not specified in the current report.

C. Digital Twin Synchronisation Fidelity.

The Digital Twin that applied to the AWS TwinMaker was tested in terms of the accuracy of synchronization, the possibility of real-time mirroring, and the availability of predictive simulation. The mean end to end latency between edge sensors, cloud processing and Digital Twin updates were about 0.9 seconds, which allowed visualisation of cultivation conditions in near real-time. Actuator states, temperature gradients, humidity dynamics, and CO₂ dynamics of the physical and virtual environments were accurately monitored by the Digital Twin with a less than 1 per cent desynchronization error. Even predictive simulation opportunities such as reduction in humidity, blockage of airflow, and failure of actuators were tried. The Digital Twin environment has demonstrated that it is possible to use simulation to predict at fidelity levels of about 92 per cent and with humidity errors of less than 4 and temperature errors of less than 2 °C, showing that the Digital Twin environment can be used to empower predictive experimentation and/or system diagnostics.

representation.

D. Resourcefulness and System Strength.

Experimental tests were conducted on the TwinGrow framework to assess its resource efficiency and operational stability with predictive actuator control. Results showed that predictive control was able to save water used for misting by about 18-22% and also saved about 30% in fan running time when compared with the baseline control method. The system was able to adapt the operation of the actuators based on the forecasted environment, thereby minimising wasteful misting and ventilation cycles, which could lead to decreases in energy use and mechanical wear. The system was evaluated for stability and capable of keeping the target environmental conditions for over 72 consecutive hours without man-involvement with the introduction of environmental disturbances. This outcome indicates that the system can be made autonomous and indicates that the architecture is appropriate to be deployed in more significant mushroom cultivation facilities.

TABLE IV. DIGITAL TWIN PERFORMANCE METRICS

Metric	Reported value
Synchronization latency	0.9 s
State desynchronization	Less than 1%
Simulation fidelity	Approximately 92%
Temperature simulation error	Less than 2 °C
Humidity simulation error	Less than 4%

The Digital Twin performance was evaluated in terms of synchronization latency, state consistency, and simulation fidelity. The reported synchronization latency was 0.9 s, while state desynchronization remained below 1%. The simulation achieved approximately 92% fidelity, with temperature simulation error below 2 °C and humidity simulation error below 4%. These metrics indicate the reported level of agreement between the physical cultivation environment and its digital

E. Limitations and Future Work

While the TwinGrow framework also provides environmental sensing, predictive analysis, automated control and Digital Twins synchronization, there are a number of limitations. The experimental assessment is only partly reported in terms of the size of the dataset, the mushroom species, the dimensions of the cultivation chamber, the duration of the mushroom cultivation and the number of repeated trials. Further documentation of detailed sensor calibration procedures and the entire training configuration for the hybrid model (EfficientNet-Temporal Transformer) is also needed. Moreover, the reported performance should be verified by experiments in longer time periods under varying cultivation conditions to verify the reliability and generalizability of the system.

Future work will involve the gathering of more and varied environmental data in several cultivation cycles and mushroom varieties. In the next step, the hybrid architecture will be evaluated against

individual components of the model as well as against traditional threshold-based control strategies, under identical evaluation criteria. Repeated trial statistical analysis, sensor calibration, etc. will also be included to ensure reliability of measurements. The framework can further be enriched with enhanced forecasting, predictive maintenance and energy efficient actuator scheduling for scalable and reliable mushroom cultivation.

V. CONCLUSION

TwinGrow is, therefore, this AIoT Digital Twin System Cloud based, which targets smarter mushroom farming. It attracts high-frequency IoT sensors, performs a small amount of preliminary processing at the edge, and routes all of it to a hybrid deep-learning pipeline on the cloud. The entire message is to maintain the expanding condition constant and provide us with almost real-time analogies of what is occurring so that we do not need to continually leap in and make tailor-made alterations. The article goes on to discuss the process of discovering elusive micro-environmental fluctuations and deep, structural incompatibilities that have the potential to disrupt mycelium development, nearly, a forensic audit of deepfake technology. The tests we conducted indicate that the framework is effective. The IoT-edge-cloud loop-maintained temperature, humidity, CO₂ and light always close to optimal. With EfficientNet, we constructed the AI stack to identify the small patterns and a Temporal Transformer to identify the long-term trends, and it hit the nail on the head, with a high degree of accuracy and low error. It implies that the system will be able to notice the speedy deviations and even detect seasonal deviations so that the mushrooms can be grown in due time and bad spikes could be reduced as much as possible. And the Digital Twin on AWS TwinMaker provided us with a hyper-realistic view of the real chamber, therefore, allowing us to run scenarios and make proactive decisions based on an intuitive interface. The best thing is the fact that it is resource efficient. The prediction of when to actuate rather than react enables us to use less water by more than 20% and turning fans and pumps off only when necessary and effectively

make the hardware last longer. That sort agrees with SDG 2, 9 and 12-zero hunger, industry and infrastructure and responsible consumption. The system also ran continuously over a few days, which is encouraging if we wish to achieve expansion into real farms or multi-chamber configurations. Nothing is not so good though. The system is currently concentrating on the ecological aspect and has not continued with the biological imaging or the contamination scan. In my opinion, the future directions might include computer vision to develop mycelial growth, anomaly detection at the spore level, soil sensor information, or even reinforcement learning to determine the optimal environmental policies independently. We could also develop a more detailed Digital Twin system that spans across several zones and cross environments interfaces so that farmers would receive analytics on the entire farm level. TwinGrow in a nutshell is the smartest agriculture system as the author balances AIoT, Digital Twins, and hybrid ML together as a complete autonomous farm system.

REFERENCES

- [1] M. A. Akkaş and R. Sokullu, "An IoT-based greenhouse monitoring system with MicaZ motes," *Procedia Computer Science*, vol. 113, pp. 603–608, 2017.
- [2] J. M. Talavera et al., "Review of IoT applications in agro-industrial and environmental fields," *Computers and Electronics in Agriculture*, vol. 142, pp. 283–297, 2017.
- [3] M. F. Mohammed et al., "IoT-based monitoring and environment control system for indoor cultivation of oyster mushroom," *Journal of Physics: Conference Series*, vol. 1019, pp. 1–7, 2018.
- [4] I. Gunawan et al., "Prototype system of temperature and humidity automatic control in oyster mushroom cultivation using Arduino Uno," *Journal of Physics: Conference Series*, vol. 1539, pp. 1–7, 2020.
- [5] A. Najmurrokhman et al., "Development of temperature and humidity control system in IoT-based mushroom cultivation," in *Proc. ISRITI*, pp. 551–555, 2020.

- [6] M. A. M. Ariffin et al., "Automatic climate control for mushroom cultivation using IoT approach," in *ICSET*, pp. 123–128, 2020.
- [7] S. D. Riskiono et al., "Control and realtime monitoring system for mushroom cultivation fields based on WSN and IoT," *Journal of Physics: Conference Series*, vol. 1655, pp. 1–10, 2020.
- [8] A. Salam, "Internet of Things in agricultural innovation and security," in *IoT for Sustainable Community Development*, Springer, pp. 71–112, 2020.
- [9] E. Adi, A. Anwar, Z. Baig, and S. Zeadally, "Machine learning and data analytics for the IoT," *Neural Computing and Applications*, vol. 32, no. 20, pp. 16205–16233, 2020.
- [10] S. Velliangiri, R. Sekar, and P. Anbhazhagan, "Using MLPA for smart mushroom farm monitoring based on IoT," *International Journal of Networking and Virtual Organisations*, vol. 22, no. 4, pp. 334–346, 2020.
- [11] Z. Jiang and Z. Xu, "The remote monitoring of agricultural information system design and implementation," *Journal of Agricultural Network Information*, vol. 11, pp. 40–43, 2010.
- [12] R. Lakshmanan et al., "Automated smart hydroponics system using Internet of Things," *IJECE*, vol. 10, no. 6, pp. 6389–6398, 2020.
- [13] A. Bhowmik et al., "IoT-based smart garbage management system," *AJSE*, vol. 19, no. 1, pp. 33–40, 2020.
- [14] A. Mehedi et al., "IoT-based healthcare middleware," in *Proc. Computing Advancements*, pp. 1–4, 2020.
- [15] M. S. U. Miah, A. Bhowmik, and R. T. Anannya, "LCDF: A unified framework for mobile data management," in *Proc. Computing Advancements*, pp. 1–5, 2020.
- [16] Apache Foundation, "XAMPP Installers and Downloads," 2021. [Online]. Available: <https://www.apachefriends.org>
- [17] Postman Inc., "Introduction to Postman," 2021. [Online]. Available: <https://learning.postman.com>
- [18] Google Brain Team, "EfficientNet: Rethinking model scaling for CNNs," *arXiv preprint arXiv:1905.11946*, 2019.
- [19] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," *arXiv:2010.11929*, 2020.
- [20] AWS, "AWS IoT Core, AWS SageMaker, and AWS TwinMaker Documentation," 2024. [Online]. Available: <https://aws.amazon.com>