Premier Journal of Engineering and Applied Sciences
Publication of Nigerian Society of Engineers, Ibadan Branch
P J E A S
Vol. 7, No. 1, 2026 ISSN:2734-3170

Development of an IoT-Enabled Fault-Tolerant Farm Intrusion Detection Control System Using Triple Modular Redundancy

I. O. Ojomuyide[a*], E. O. Omidiora[b], A. S. Falohun[c] and Y. O. Mojeed[d]

[a]Department of Computer Science, Osun State University, Osogbo, Nigeria
[b]Department of Computer Engineering, Ladoke Akintola University of Technology, Ogbomoso, Nigeria

***Corresponding Author:** ojomuyide2000@gmail.com*

ARTICLE INFO

Received: October, 2025
Accepted: December, 2025
Published: January, 2026

Keywords:
Fault-Tolerance
Hotspot
Internet of Things
Redundancy
Triple Modular.

ABSTRACT

Agriculture remains the backbone of global sustenance and economic development, providing essential food, raw materials, employment and economic stability. However, despite its importance, it faces challenges such as food scarcity and financial losses caused by security breaches. Baseline Technique (BT) used in existing studies were hindered by sensor failure and lack of fault tolerant circuit. Therefore, this study developed an IoT-based fault-tolerant farm intrusion detection control system using Triple Modular Redundancy (TMR). Dataset of 2,150 sample images, categorized into two primary classes: 1,300 human and 850 animal were acquired from kaggle.com. The acquired images underwent preprocessing to eliminate unwanted noise, normalized sensor reading and convert infrared radiation image detected into electrical signals. The signals were amplified and processed by an internal amplifier and comparator. The processed output signals were transmitted to a BT through a fault-tolerant circuit which was formulated using Triple Modular Redundancy (TMR). This handled sensor faults and ensured accurate outputs were maintained even when individual sensors failed. The output of the fault-tolerant circuit was integrated with a voter circuit which generated final decision output. The study was simulated using Proteus 8.12 and implemented in Arduino C. The performance of the TMR technique was evaluated using reliability, availability, uptime, downtime and mean time between failures. A comparative analysis between the fault-tolerant circuit and BT was carried out to evaluate the improvement achieved through the fault-tolerant circuit. The reliability, availability, uptime, downtime and mean time between failures for fault-tolerant circuit were 100%, 95%, 24 hrs, 0 hr and 0 hr, respectively, while the corresponding values for BT were 37%, 87%, 21 hrs, 3 hrs and 2 hrs. The research revealed that fault-tolerant circuit perform better than the Baseline Technique.

1. INTRODUCTION

Agriculture remains the backbone of global sustenance and economic development, providing essential food, raw materials, employment, and economic stability. However, despite its importance, it faces challenges such as food scarcity, reduced yields, and financial losses caused by security breaches. In Nigeria, insecurity has become a factor that compromised the performance of the agricultural sector because there are no proper surveillance mechanisms to check on the farmland against animals, intruders,

kidnappers, and other criminal activities in the rural farming communities. Lack of the proper monitoring mechanisms has left farmlands vulnerable to frequent intruders and grazing animals, armed bandits, and kidnappers and contributed to massive destruction of crops, loss of livestock, loss of productivity, and increased food insecurity (Aderayo and Omojowo, 2024). Movements of animals, particularly cattle, have caused massive destruction of crops, degradation of soil, and conflicts between farmers and herders in most of the agricultural areas in Nigeria. This has led to decrease in farming activities, thus undermining agricultural production (Amadi *et al.*, 2024).

More so, the absence of quality surveillance to keep track of animals, intruders, and criminal activities is one of the critical elements contributing to increasing insecurity in the agriculture sector in Nigeria (Oladoyin *et al.*, 2024). Several methods have been adopted to control the aforementioned, such as the Baseline Technique (BT), used in existing studies, which was hindered by sensor failure and a lack of fault-tolerant circuits. This study developed an Internet of Things (IoT)-based fault-tolerant farm intrusion detection control system using Triple Modular Redundancy (TMR). The need for this system lies in highly resilient, fault-tolerant, real-time monitoring and accurate detection of unauthorized intrusions. By leveraging TMR, the developed system ensures uninterrupted operation even in the event of hardware or sensor failures. This system adopted three objectives. The first objective is to design a fault-tolerant farm intrusion detection control system (FTFIDCS) using TMR. The second objective was to simulate the system using Proteus 8.12 and also implement the design using Arduino C programming with the IoT. The third objective is to test the performance of the developed system against the existing system on the basis of reliability, availability, uptime and downtime, mean time between repairs.

A number of studies have concentrated on the application of TMR in IoT-enabled fault-tolerant farm intrusion detection control systems. For example, Panda *et al.* (2022) discussed a field monitoring system that leverages the IoT to detect wild animal incursions on agricultural farms. The methodology involved deploying ultrasonic sensors at the corners of the field to detect intrusions. An E-vehicle equipped with a Node MCU Microcontroller and a mounted camera captured images of the intruder. An IoT application was used to send alert messages to the farmer, enabling timely response to potential threats. The results demonstrated the efficacy of the developed system in capturing images of intruders and sending timely alerts, allowing efficient identification of any kind of intrusion around the field. A limitation of this study was the reliance on ultrasonic sensors, lack of fault tolerant and an E-vehicle, which may not fully cover large fields or adapt well to diverse terrain conditions, potentially affecting system scalability and performance. Raghuvanshi (2022) conducted a study on intrusion detection in agriculture and implemented the model based on machine learning, which shows the promise of these data-driven models to detect threats. However, limitations included lack of redundancy and continued protection due to inconsistencies within the system. Furthermore, delayed detection and false alarms would usually deter the surveillance system of the IoT.

Kethineni and Gera (2023) presented an IoT-based privacy-preserving anomaly detection system for smart agriculture. The proliferation of IoT technology and its application in modern agriculture, known as smart farming, allow farmers to use data collected and stored using IoT capability to monitor crop health, productivity, and field conditions. Ibam *et al*. (2023) worked on IoT-based farm intrusion detection system was proposed to address the increasing cases of crop vandalisation, conflicts between farmers and herdsmen in Nigeria. The developed system employed RFID-based identification and image recognition methods, in which RFID sensors were used to authenticate authorized farm workers through tag detection, while surveillance cameras captured images of individuals entering the farmland. The captured images were processed using a convolutional neural network for face and object recognition to distinguish authorized users from intruders. The primary objective of the study was to provide an effective intrusion detection mechanism with real-time alerts and visual evidence to enhance farm security. The system achieved an accuracy of 90%, a precision of 70%, and a recall of 80%, demonstrating its effectiveness in controlling

illegal farm encroachment. The limitation of the work is lack of fault tolerant, poor image quality and environmental conditions which affect recognition performance evidence and real-time intrusion alerts to farm owners.

Ravindra *et al*. (2024) developed a wireless sensor network-based farm monitoring and security system to address the limitations of conventional fencing methods, ensuring efficient crop protection and farm management. The system employs a network of sensor nodes, surveillance cameras, and a central base station to monitor environmental parameters like soil moisture, temperature, and pest activity while detecting intrusions using motion detection algorithms. Microcontroller platforms like Arduino and ZigBee communication protocols enhance the system's scalability and interoperability. The developed system enables real-time monitoring, automates deterrent measures against intrusions, provides granular insights for precision agriculture, and reduces labor costs through remote monitoring and control. The system's effectiveness may be constrained by challenges such as network reliability, lack of fault tolerant, the initial cost of deployment, and the adaptability of small-scale farmers to advanced technology.

Oyelade *et al*. (2024) developed a farmland intrusion detection system utilizing both sensor and computer vision technology with the IoT. Farm intrusion is a major issue that can lead to unauthorized entries and losses for farmers. This study designed and implemented an unauthorized access detection and prevention system for farmland using advanced sensors and machine learning approaches. The approach consisted of arranging a set of video sensors and passive infrared (PIR) sensors around the farm. The limitation of the system is that it lacks fault tolerant. Shaik *et al*. (2024) worked on the IoT-Enabled Real-time Smart Agriculture Monitoring and Intrusion Detection System (IoT-SAMIS) for smart agriculture. ESP32 microcontrollers and sensor networks were used to monitor agriculture management conditions (soil moisture, temperature, humidity, and crop health) in this system. The data was sent to the cloud by a Wi-Fi module connected to the ESP32. The study relies on stable internet/multiple IoT; it may not work well in long-range and rural areas with limited connectivity. The limitation of the system is that it lacks fault tolerant. Miao *et al*. (2025) discussed a smart agriculture system aimed at detecting animal intrusion at the perimeter edge of the farm. The limitation was that the system performance could be affected by environmental factors and the search for further adaptations from a variety of agricultural contexts.

Summarily, fault tolerant mechanisms were not incorporated in most of the work that was done in past. The few instances of fault tolerance that were taken into account were mainly software-based, and were not well supported on the hardware or system architecture level. Consequently, such systems are susceptible to component failures and runtime failures thus lowering its overall reliability and robustness in operation. However, the FTFIDCS proposed based on TMR and IoT can also be used to overcome these shortcomings by providing an efficient fault-tolerant architecture that will maintain its functionality even when sensors or modules are broken down.

2. METHODOLOGY

The methodology describes the design and evaluation of an IoT fault-tolerant farm intrusion detection system. The system is divided into software module, hardware module, experimental setup, and performance evaluation sections. The Architectural design of the developed system is illustrated in Figure 1, which presents the graphical interconnection of the system's major components and their communication flow. Figure 2 depicts the system flowchart showing the logical operation behavior of a fault-tolerant motion detection process, in which three parallel modules simultaneously process input signals from the PIR sensor, and a majority-voting mechanism determines the final system decision to ensure accessibility, reliability, and accuracy.

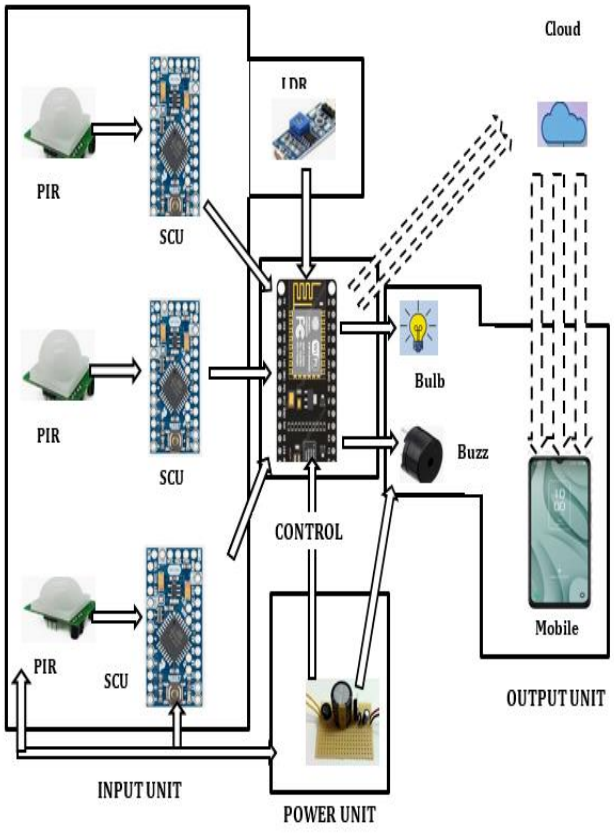


Figure 1: Architecture of the developed System

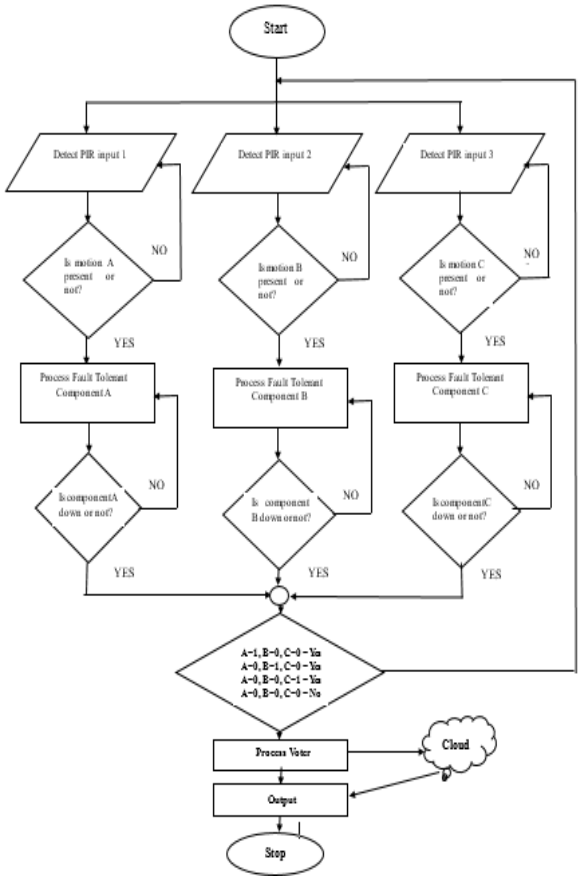


Figure 2: Flowchart of the developed system

2.1 Software Module

The software module leveraged the strengths of both Arduino IDE and MIT App Inventor to create a robust interactive system. The embedded software was developed to core the operation of the system and was deployed via the Arduino IDE. Codes were written and burned on the Micro Arduino microcontroller using Arduino C and run the fault-tolerant circuit using the TMR. MIT Inventor is an Android software available online that aids in the creation of Android applications; it comprises of many components that throughout the layout. It has code blocks that may be arranged by dragging and dropping arrangement block of code as shown in Figure 3. The interface was designed to have a user-friendly mobile application that act as the notification interface to the system, which was created using MIT App Inventor. The app communicates to the cloud service, Firebase, in order to receive real-time message sent by voter controller. With this mobile application, farmer will be notified the current security status in the farm remotely, know immediately when there is an intrusion. The cloud-based and graphics/block-based design system of MIT App Inventor facilitates easy communication between the mobile device and cloud, which increases system accessibility, usability, and real-time situational awareness for farm owners.

2.2 Hardware module

The hardware module comprises three major components: the PIR sensors, the ESP8266 microcontroller, microcontroller, and the TMR voter circuit as shown in Figure 4. The sensors are strategically installed around the farm perimeter to detect human and animal movements. Each PIR sensor transmits its signal to a corresponding processing unit within the TMR circuit. Three identical processing units perform simultaneous evaluation, evaluation, and their outputs are sent to the voter mechanism. The voter decides

the final output based on the majority logic, ensuring that the system remains accurate even if one unit fails. A 5VA 5V power supply generated from a 12V DC regulated circuit incorporating bridge diodes and capacitors for voltage stabilization was supplied to the aforementioned integrated circuit. The ESP8266 configured in shared mode connected Firebase over Wi-Fi through its host address, ensuring real-time synchronization and alert transmission to the mobile device.

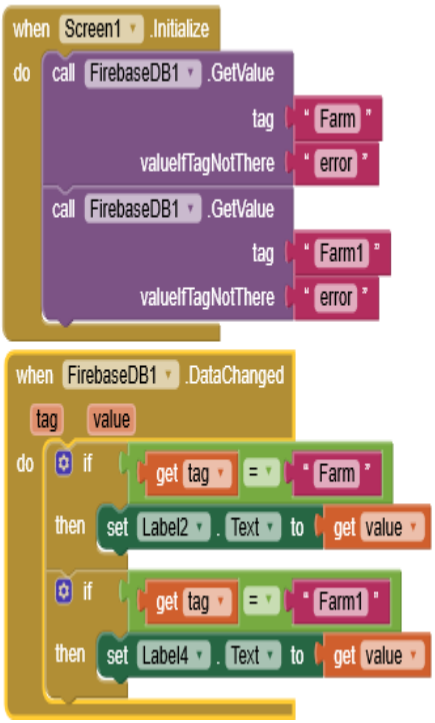


Figure 3: MIT block codes

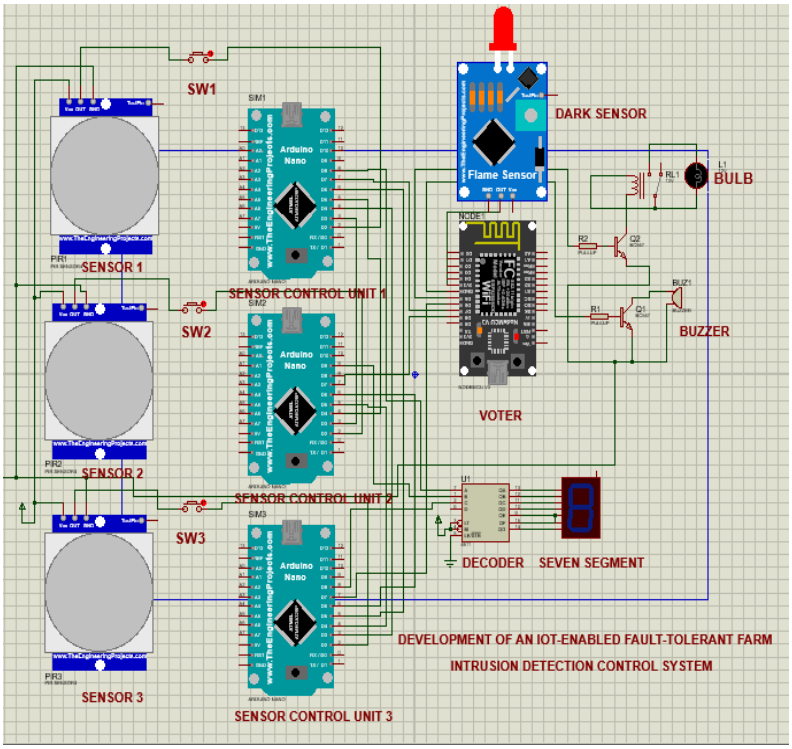


Figure 4: The circuit of the system

The NODEMCU8266 microcontroller was designed as a voter incorporated into the system by linking its GND pin to the power supply ground terminal while its VCC pin was connected to the +5V output of the power supply. The D4 of each fault tolerant represented by Arduino micro were connected to the D0, D1 and D2 of NODEMCU8266 while output of LDR was connected to D4, D5 was connect to positive pin of buzzer and its GND pin link to ground of power supply and the D7 pin was linked to the base of a C1815 transistor which acts as a buffer to energize a 12V relay to control bulb. The PIR sensor was connected to the Arduino microcontroller to enable motion detection within the system. The VCC pin of the PIR sensor was connected to the 5V output pin of the Arduino to supply power, while the GND pin of the sensor was connected to the Arduino's ground to complete the circuit. The OUT pin of the PIR sensor, which generated a digital signal, connected to the digital input pin D2 of the Arduino. This configuration allowed the Arduino to read the HIGH or LOW logic signals from the PIR sensor and respond accordingly during system operation. This connection represented a fault-tolerant unit arranged in parallel form of three components synchronized to the D3 of components A, B, and C for proper communication of which component responded when one is faulty.

The PIR is the major sensor that detect both human and animal that intrude the farm land, the sensor uses data set of about 2,150 contains human and animal as shown in Figure 5. This images gave clear picture of what the sensor capable of determine. Once the sensor senses acquired images, the image underwent preprocessing to eliminate unwanted noise, normalized sensor reading and convert infrared radiation image detected into electrical signals. The signals were amplified and processed by an internal amplifier and

comparator. The processed output signals was later used in fault tolerant circuit for further processing. Figure 6 shows TMR configuration which was implemented using three identical components operating in parallel to ensure system fault tolerance. Each hardware component was triplicated, designated as Component A, B, and C, and executed the same task simultaneously. The outputs from the three modules were compared using a voting mechanism, which selected the correct output based on majority agreement. When all three outputs matched, the system maintained an idle and stable state, ensuring reliability and accuracy in operation.

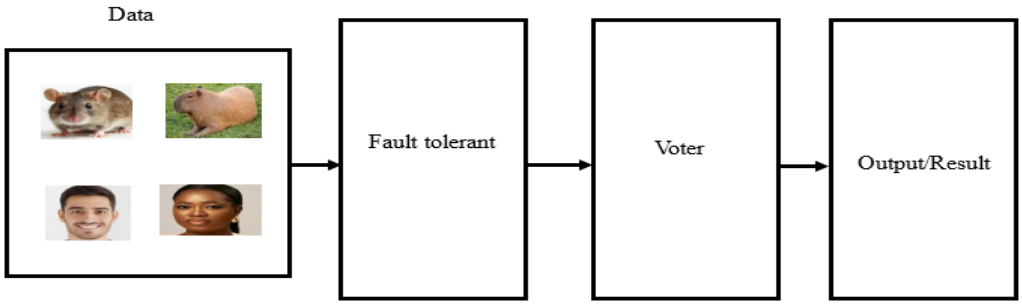


Figure 5: Conceptual view of the system

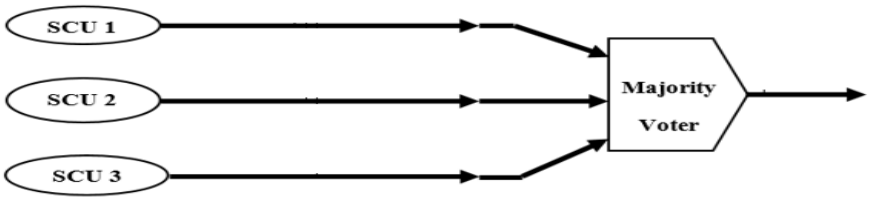


Figure 6: The developed TMR architecture

2.3 Experimental Setup

The simulation for the developed IoT-enabled fault-tolerant farm intrusion detection control system was developed using the Arduino C programming environment to evaluate the system's performance, fault tolerance, and reliability in detecting farm intrusions as shown in Figure 4. The simulation was performed on a 64-bit Windows 10 operating system with 4 GB RAM and a 2 GHz Intel Core i5 CPU to ensure effective processing and execution of the program. The experimental procedure was done in a controlled simulation with the use of Proteus 8.12 software and was confirmed using an ESP8266 microcontroller. The experimental set consists of two stages, namely, the power supply unit, the ESP8266 and Arduino micro. Despite the fact that much of the system validation involved simulation appliances like the Proteus and the Arduino development environment to test the functionality of the circuits, and system integrity, the system was also applied and functionally tested in a farm-related environment to represent the actual operation conditions as shown Figure 7.

The hardware was then assembled and installed at strategic positions, which represent the three entry points on the farm and open field boundaries, and is comprised of the PIR sensors, microcontroller, ESP8266 module, and communication interfaces after the verification of their functionality in simulation-based tests. Sensors were observed to react to human and animal movement, and intrusion alerts were sent to the cloud and mobile application in real time, thus confirming the behavior of the system in a realistic farm-like setting. The deployment was, however, not a large-scale farm deployment but a pilot one due to the limitations in place, like access to large-scale farmland, cost, and time. As a result, though simulations were much utilized to maintain the stability of the system and guarantee fault tolerance, the outcomes were backed by real-life physical hardware testing in a farm environment, which showed that the system can be applied to actual farm security situations.

Figure 7: Farm deployment

2.4 Performance Evaluation

The performance metrics of the IoT fault-tolerant farm intrusion detection control system, which had been developed, were assessed in terms of availability, reliability, mean time between failures (MTBF), and downtime. The assessment was conducted by simulating operational conditions, including normal operation, failure of single components, and environmental disturbances. These metrics formulae are stated in equations 1 to 6.

a. Reliability (R (t)) $=e-\lambda/t$ (1)

Where R (t) – Reliability at time *t*, λ =Failure rate, t =Time of operation, e =Mathematical constant=2.71828

b. Mean time between failures (MTBF): Average time a system operates before failure occurs. (2)

c. Failure Rate (λ): The frequency at which failures occur in a system.

Failure rate (λ) = $\text{Number of failure} \div \text{Uptime}$ (3)

d. System Uptime: Time the system is operational.

e. Downtime: Time the system is unavailable due to failure or maintenance.

f. Time: Time of operation.

g. Availability (A) = $(\text{Uptime} \div (\text{Uptime} + \text{Downtime}))$ (4)

h. Average Availability: Availability aggregate divided number of occurrence. (5)

i. Average Reliability: Reliability aggregate divided number of occurrence. (6)

Illustration of Reliability

Reliability R (t) $=e-\lambda/t$

Step 1: Deduce failure rate

Failure rate (λ) = $\text{Number of failure} \div \text{Uptime} = 0/24=0$

Time of operation (t) = 24hrs. (Uptime)

e =Mathematical constant=2.71828

Step 2: Deduce reliability

R (t) $=e-\lambda/t = 2.71828^{(-0x24)}$

R (t) = $2.71828^{(0)}$, recall that anything raise to power zero in indices is 1

R (t)=1

Step 3: Convert to percentage

Reliability (%) = 1x100=100%

Illustration of Availability

Availability (A) = $(\text{Uptime} \div (\text{Uptime} + \text{Downtime}))$

Step 1: Deduce availability

A= (24/(24+0))

A=24/24=1

Step 2: Convert to percentage

A (%) =1x100= 100%

Illustration of average reliability

Average Reliability= Reliability aggregate divided number of occurrence.

Average Reliability= 100+37.10+38.50+100+37.30+100+36.70/7

Average reliability=4496/7

Average reliability =64.22

llustration of average availability

Average availability= Availability aggregate divided number of occurrence.

Average Reliability= 100+91.67+95.83+100+87.50+100+83.33/7

Average Reliability= 658.33/7

Average Reliability=94.047

3. RESULTS AND DISCUSSION

As presented in Tables 1–4**,** the fault-tolerant system demonstrated consistently high availability, frequently achieving 100% uptime and reliability on days with zero downtime, particularly on Mondays, Thursdays, and Saturdays across all weeks. These peak performance periods coincided with stable power supply and moderate ambient temperatures (15°C–25°C)**,** conditions that enhanced sensor accuracy and hardware stability. However, notable performance degradation was observed on Fridays and Sundays, where increased downtime of 3–4 hours resulted in availability dropping to 87.50%–83.33% and reliability declining sharply to approximately 37.20%–36.70%**.** These performance dips were strongly associated with unstable electricity supply and elevated ambient temperatures above 30°C, which adversely affected sensor sensitivity and processing stability. This recurring pattern confirms that although the fault-tolerant system maintains high uptime, its reliability remains highly sensitive to fault duration and environmental stressors, with even short outages significantly reducing the probability of uninterrupted fault-free operation.

In contrast, the baseline system (Tables 6–9) exhibited substantially lower performance across all weeks, with availability rarely exceeding 87.50% and reliability consistently remaining below 40%**,** even during periods of reduced downtime. Extended downtime events of 7–8 hours, particularly on Fridays and Sundays, caused availability to fall to 70.83%–66.67% and reliability to approximately 33.80-32.50%**,** demonstrating high vulnerability to faults. These degradations were further intensified by high ambient temperatures and poor power stability**,** which the baseline system lacked the architectural resilience to mitigate. Table 5 and Table 10, in terms of availability**,** the fault-tolerant technique consistently achieved high performance, with values ranging between 91.07% and 94.05%**,** indicating that the system remained operational for most of the evaluation period. In contrast, the baseline technique recorded significantly lower availability values, ranging from 73.24% to 78.57%**,** which reflects frequent system downtime. This shows that the absence of fault-tolerance mechanisms in the baseline system negatively affected its ability to maintain continuous operation.

Table 1: Week 1 performance evaluation result for seven days (Fault tolerant)

Day	Uptime (Hours)	Downtime (Hours)	Availability (%)	Failure rate	MTBF (Hours)	Reliability (%)
Mon	24	0	100	0	N/A	100.00
Tue	22	2	91.67	0.090	22	37.10
Wed	23	1	95.83	0.043	23	38.50
Thur	24	0	100	0	N/A	100.00
Fri	21	3	87.50	0.142	21	37.30
Sat	24	0	100	0	N/A	100.00
Sun	20	4	83.33	0.20	20	36.70

Table 2: Week 2 performance evaluation result for seven days (Fault tolerant)

Day	Uptime (Hours)	Downtime (Hours)	Availability (%)	Failure rate	MTBF (Hours)	Reliability (%)
Mon	23	1	95.83	0.043	23	38.50
Tue	21	3	87.50	0.142	21	37.30
Wed	22	2	91.67	0.090	22	37.10
Thur	24	0	100	0	N/A	100.00
Fri	20	4	83.33	0.20	20	36.10
Sat	24	0	100	0	N/A	100.00
Sun	19	5	79.17	0.26	19	35.80

Table 3: Week 3 performance evaluation result for seven days (Fault tolerant)

Day	Uptime (Hours)	Downtime (Hours)	Availability (%)	Failure rate	MTBF (Hours)	Reliability (%)
Mon	24	0	100	0	N/A	100.00
Tue	23	1	95.83	0.043	23	38.50
Wed	22	2	91.67	0.090	22	37.10
Thur	23	1	95.83	0.044	23	38.50
Fri	21	3	87.50	0.142	21	37.30
Sat	24	0	100	0	N/A	100.00
Sun	20	4	83.33	0.20	20	36.70

Table 4: Week 4 performance evaluation result for seven days (Fault tolerant)

Day	Uptime (Hours)	Downtime (Hours)	Availability (%)	Failure rate	MTBF (Hours)	Reliability (%)
Mon	23	1	95.83	0.043	23	38.50
Tue	23	2	91.67	0.086	22	37.10
Wed	21	3	87.50	0.142	21	37.30
Thur	24	0	100	0	N/A	100.00
Fri	20	4	83.33	0.20	20	36.30
Sat	23	1	95.83	0.043	23	38.50
Sun	24	0	100	0	24	100.00

Table 5: Monthly Summary of Availability and Reliability (Fault tolerant)

Week	Average Availability	Average Reliability
Week1	94.05	64.22
Week2	91.07	54.97
Week3	93.45	55.37
Week4	93.45	55.38

Table 6: Week 5 performance evaluation result for seven days (Baseline Technique)

Day	Uptime (Hours)	Downtime (Hours)	Availability (%)	Failure rate	MTBF (Hours)	Reliability (%)
Mon	20	4	83.83	0.20	20	36.2
Tue	18	6	75.0	0.33	18	34.10
Wed	19	5	75.50	0.26	19	35.00
Thur	21	3	87.5	0.142	21	37.00
Fri	17	7	70.83	0.411	17	33.80
Sat	20	4	83.33	0.20	20	36.00
Sun	16	8	66.67	0.50	16	32.50

Table 7: Week 6 performance evaluation result for seven days (Baseline Technique)

Day	Uptime (Hours)	Downtime (Hours)	Availability (%)	Failure rate	MTBF (Hours)	Reliability (%)
Mon	19	5	79.17	0.26	19	35.0
Tue	17	7	70.83	0.41	17	33.8
Wed	18	6	75.00	0.33	18	34.10
Thur	20	4	83.33	0.20	20	36.20
Fri	16	8	66.67	0.50	16	32.50
Sat	19	5	79.17	0.26	19	35.10
Sun	15	9	62.50	0.60	15	31.90

Table 8: Week 7 performance evaluation result for seven days (Baseline Technique)

Day	Uptime (Hours)	Downtime (Hours)	Availability (%)	Failure rate	MTBF (Hours)	Reliability (%)
Mon	21	3	87.50	0.142	21	37.00
Tue	19	5	79.17	0.263	19	35.00
Wed	18	6	75.00	0.333	18	34.10
Thur	20	4	83.33	0.20	20	36.20
Fri	17	7	70.83	0.411	17	33.80
Sat	21	3	87.50	0.142	21	37.10
Sun	16	8	66.67	0.50	16	32.50

Table 9: Week 8 performance evaluation result for seven days (Baseline Technique)

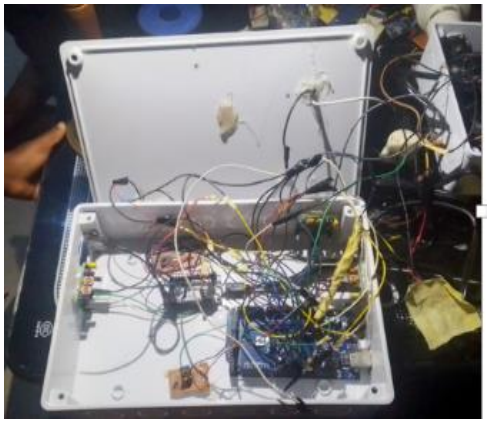
Day	Uptime (Hours)	Downtime (Hours)	Availability (%)	Failure rate	MTBF (Hours)	Reliability (%)
Mon	20	4	83.33	0.20	20	36.20
Tue	18	6	75.00	0.333	18	34.10
Wed	17	7	70.83	0.411	17	33.80
Thur	21	3	87.50	0.142	21	37.00
Fri	16	8	66.67	0.50	16	32.50
Sat	20	4	83.33	0.20	20	36.80
Sun	18	4	75.00	0.222	18	34.10

Table 10: Monthly Summary of Availability and Reliability (Baseline Technique)

Week	Average Availability	Average Reliability
Week1	77.69	34.80
Week2	73.24	33.80
Week3	78.57	35.10
Week4	77.24	34.80

In all the tables presented below the reliability was deduced using equation 1, MTBF was calculated using equation 2, failure rate was calculated using equation 3, availability was calculated using equation 4, average availability and average reliability were calculated using equation 5 and 6. Figure 8 presents the internal control structure, the internal power structure, the active mode design, and the idle mode design structure of the developed system. The active mode shows that the system is in operation, while idle mode describes when the system is not operating. Figure 9 shows mobile app that send notification of all activities within the farm to the farmer and also display current fault tolerant active. Figure 10 presents a graphical comparison of the average availability and reliability for both the fault-tolerant and baseline techniques over the four-week evaluation period. The fault-tolerant method recorded an average availability of more than 91.07% as well as an average reliability of 54.97%, whereas the baseline method registered an average availability of less than 78.57% and an average reliability of about 35.10%.

These findings help draw the conclusion that fault tolerance is an important factor that enhances the performance of the system, especially in regard to availability and reliability. To conclude, the gathered data indicate that the fault-tolerant method is better than the baseline method in terms of availability and reliability. The graph clearly validates the numerical results by demonstrating that the fault-tolerant technique consistently outperforms the baseline technique in both availability and reliability. This confirms that integrating Triple Modular Redundancy into the IoT-based farm intrusion control system substantially enhances system robustness, fault resilience, and operational stability.



A: Internal structure of control

B: Internal structure of the power

C: Design in active mode

E: Design in idle mode

Figure 8: Internal structure of developed system

Figure 9: Mobile App

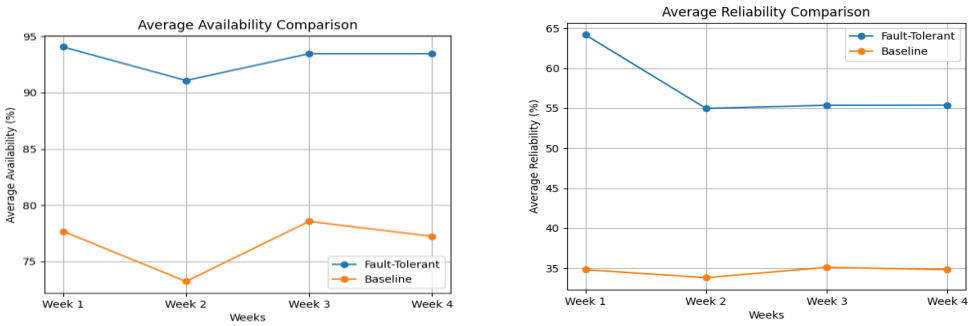


Figure 10: Average Availability and Average Reliability Comparison

4. CONCLUSION

This paper constructed and tested an IoT-based fault-tolerant system and compared its functionality with a baseline method without fault tolerance on the basis of the metrics of both availability and reliability. The findings indicated that fault tolerance increases the overall system reliability, specifically in terms of raising operational availability and decreasing service failures. The research proves that fault tolerance is a successful method of improving the performance of the system in an IoT-based application. The findings, however, also pointed out that additional optimization is required to reach greater levels of reliability. The results of this work offer a viable basis for future enhancements in the IoT system design, which is robust and reliable.

References

Abeysinghe, S. (2022). Project quality management. In Managing Information Technology Projects: Building a Body of Knowledge in IT Project Management (pp. 245-281).

Amadi-Robert, C. C., Thankgod, T., Agbagwa, K. S., and Chukwuemeka, C. S. (2024). Impact of insecurity in the performance of agriculture sector in Nigeria. International Journal of Economic, Finance and Management (IJEFM), 9(4): 142157.

Aderayo, A. A., and Omojowo, S. T. (2024). Food security and rural banditry in Oyo state (2019–2023). Journal of political Studies, 31: 45.

Ibam, E., Boyinbode, O. K., and Aladesiun, H. (2023). IoT protected farmland intrusion detector. Buana Information Technology and Computer Sciences (BIT and CS), 4(2): 3953).

Kethineni, K., and Gera, P. (2023). IoT-based privacy-preserving anomaly detection system for smart agriculture. *Systems*, 11(6): 304.

Nong, N. B., & Tarek, M. (2023). Surveillance approaches to intrusion detection, crop health, and Disease prevention in agriculture. *Quarterly Journal of Emerging Technologies and Innovations*, 8(3), 1–1.

Oche, O. E., Nasir, S. M., and Muhammed, A. H. (2020). Internet of Things-based agriculture: A review of security threats and countermeasures. *International Journal of Scientific Research and Publications*, 10(8).

Oyelade, I. M., Boyinbode, O. K., and Adewale, O. (2024). A review of existing farmland intrusion detection systems. *International Journal of Computer Applications*, 975: 88–87.

Oladoyin, A. M., Osimen, G. U., Adi, I., and Dada, O. (2024). The increasing insecurity in Nigeria: Questioning the connection between poverty and banditry. Educational Administration: Theory and Practice.

Onwuka, I. E., Afolabi, M. O., John, I. O., and Idowu, A. (2018). Design and implementation of farm monitoring and security system. *International Journal of Computer Applications*, 181(9): 10–15.

Olajide, O. B., Ayodeji, O. O., James, O., Buffington, F. N., Olatunji, L. M., and Yakubu, Y. (2022). Development of a reliable fault-tolerant traffic light system controller system. *Asian Basic and Applied Research Journal*, 131–139.

Panda, P. K., Kumar, C. S., Vivek, B. S., Balachandra, M., & Dargar, S. K. (2022, February). Implementation of a wild animal intrusion detection system based on Internet of Things. *In 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, 6(7): 1256–1261.

Raghuvanshi, A., Singh, U. K., Sajja, G. S., Pallathadka, H., Asenso, E., Kamal, M., ... and Phasinam, K. (2022). Intrusion detection using machine learning for risk mitigation in IoT-enabled smart irrigation in smart farming. *Journal of Food Quality*, 1: 3955514.

Ravindra, N. A., Bapusaheb, P. N., Vilas, R. K., and Niranjan, R. A. (2024). Farm security system.

Shaik, S., Adhil, M. D., Jennifer, M., Krishna, M., Kalyan, B., and Begum, H. (2024). An IoT-enabled real-time smart agriculture monitoring and intrusion detection system. *In 2024 IEEE International Conference on Information Technology*.

Song, C., Ma, W., Li, J., Qi, B., & Liu, B. (2022). Development trends in precision agriculture and its management in china based on data visualization. *Agronomy*, *12*(11): 2905.

Sishodia, R. P., Ray, R. L., and Singh, S. K. (2020). Applications of remote sensing in precision agriculture: A review. *Remote sensing*, *12*(19): 3136.

Shaheb, M. R., Sarker, A., and Shearer, S. A. (2022). Precision agriculture for sustainable soil and crop management. In *Soil Science-Emerging Technologies, Global Perspectives and Applications*. Intech Open.

Thakur, D., Kumar, Y., and Vijendra, S. (2020). Smart irrigation and intrusions detection in agricultural fields using IoT. *Procedia Computer Science*, 167: 154–162.

Vemuri, N., Thaneeru, N., and Tatikonda, V. M. (2023). Smart Farming Revolution: Harnessing IoT for Enhanced Agricultural Yield and Sustainability. *Journal of Knowledge Learning and Science Technology*, 2(2): 143–148.

Yu, W., Chen, Z., Wang, H., Miao, Z., and Zhong, D. (2025). Industrial network intrusion detection in open-set scenarios. *International Journal of Information Security*, 24(1): 39.