SJER Scientific Journal of Engineering Research

futuristech PT. Teknologi Futuristik Indonesia

Date of publication December 26, 2025, date of current version December 26, 2025.
e-ISSN: 3109-1725
Digital Object Identifier 10.64539/sjer.v2i1.2026.365

Article

Design and Simulation of a Scalable IoT-Based Multi-Sensor Prototype for Pipeline Security Monitoring

Evander Chika Udeh[1,*], Michael Chukwuebuka Agwu[1], Pamilerin Samuel Akinrinde[1], Nnaemeka Sunday Ugwuanyi[1], Akudo Ogechi Nwogu[1]

[1] Department of Electrical/Electronic Engineering, Alex-Ekwueme Federal University, Ndufu-Alike Ikwo, Nigeria; evanderude@gmail.com
* Correspondence

The authors received no financial support for the research, authorship and publication of this article.

Abstract: Pipeline vandalism and leaks pose a significant threat to global energy infrastructure, leading to severe economic losses and environmental degradation. Traditional surveillance methods are often reactive and insufficient for monitoring vast, remote pipe-line networks in real-time. To address this gap, this study designs and simulates a multi-sensor Internet of Things (IoT) proto-type that integrates gas, vibration, and temperature monitoring for anomaly detection. The methodology employs a design-and-simulation ap-proach using an Arduino Uno and ESP8266 Wi-Fi module within the Proteus environment. Key findings demonstrate the functional correctness of the system's logic, achieving consistent alert triggering based on predefined heuristic thresholds with no failures in the simulated environment. These results imply that a low-cost, multi-modal sensor fusion approach provides a technically feasible foundation for future physical deployment in infrastructure security.

Keywords: Internet of Things (IoT); Pipeline Security; Sensor Networks; Nigeria; Anomaly Detection.

1. Introduction

Pipeline vandalism continues to pose a significant challenge within Nigeria's oil and gas sector, leading to substantial economic setbacks, environmental harm, and security threats [1], [2]. Annually, billions of naira are lost due to oil theft and the costs associated with repairing damaged pipelines [3]. Frequent spills contaminate water sources and agricultural lands, destroying ecosystems and displacing local communities [2], [4]. Moreover, recurrent explosions and fires caused by compromised pipelines have resulted in over 2,000 fatalities over the past twenty years [1], [5]. The expansive and often hard-to-reach ter-rain housing Nigeria's pipeline infrastructure renders tra-ditional surveillance methods—such as manual patrols and limited aerial monitoring—both expensive and inef-fective for real-time threat detection [2].

The emergence of Internet of Things (IoT) technology presents a promising solution to these persistent issues. IoT-based monitoring systems utilize sensors, communi-cation networks, and data analytics to enable continuous monitoring of critical parameters, including pressure, temperature, vibration, and gas levels [6]-[8]. Recent re-search highlights the potential of affordable, IoT-enabled prototypes tailored for pipeline monitoring in Nigeria and comparable environments [9], [10], alongside increased in-tegration of machine learning (ML) techniques for real-time anomaly detection and predictive maintenance [11]-[13]. Similarly, studies by Mulla and Liyakat [14] demon-strate that IoT sensors combined with ML algorithms can effectively monitor pipeline pressure and flow rate to de-tect leakages, reinforcing the practical applicability of such systems in complex environments. These systems enable the early identification of leaks, intrusions, or sabotage, al-lowing for swift responses that can mitigate damage. Nonetheless, deploying IoT and Wireless Sensor Net-works (WSNs) in Nigeria faces hurdles such as high costs, limited scalability, and environmental and security chal-lenges that hinder effective adaptation [6], [15]. Therefore, there is a pressing need for cost-efficient, scalable, and multi-modal architectures that are customized to local conditions.

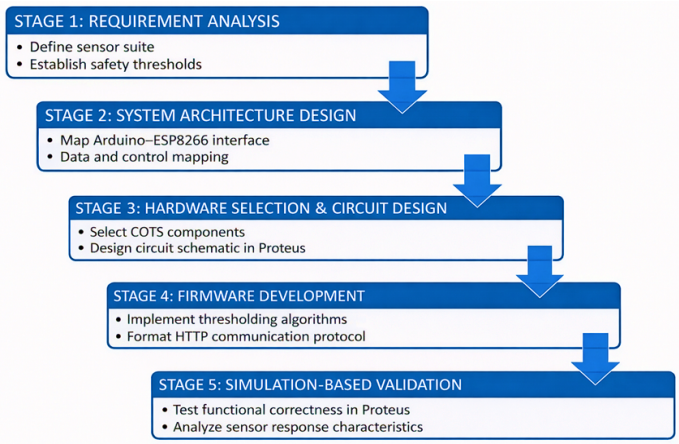

Figure 1. Five-stage workflow of the proposed system.

While existing research has explored various IoT solutions for pipeline safety, most are restricted to single-parameter detection or conceptual frameworks lacking real-time intelligence. For example, Bello et al. [16] developed a WSN for vandalism and leak detection but focused solely on single-sensor data. Odulaja and Rufai [17] proposed an expert system for intrusion alerts with manually set thresholds. Others, like Agbolade et al. [18], designed LoRaWAN-based systems integrating pressure and flow sensors for leak detection, while Bukie et al. [19] implemented multi-sensor setups emphasizing cloud data management. Similarly, Effiom et al. [20] utilized GSM communication for real-time monitoring, and Ojo et al. [21] employed MQ sensors with Wi-Fi for gas leak detection. Nigerian studies such as those by Ezeja and Nwobi [22], and Okorodudu et al. [23], contributed foundational ideas and WSN frameworks but lacked comprehensive multi-sensor integration and automated decision-making features. More recently, prototype systems tailored for Nigeria using low-cost multi-sensor fusion and IoT platforms have been explored [9], [24], and global IoT systems focusing on energy-efficient network performance provide insights for scalable implementations [25].

Although prior studies have advanced IoT-enabled pipeline security, most remain narrowly focused on single-parameter leak detection and lack integrated assessments of mechanical and environmental anomalies. High-performance global systems exist, but their costs hinder large-scale deployment in developing countries like Nigeria. To directly address these gaps, this research proposes a low-cost, resilient IoT-based prototype that fuses gas, vibration, and temperature sensing with automated anomaly detection. The three-prong sensor approach offers a compelling advantage by enabling the simultaneous detection of chemical leaks, mechanical tampering, and thermal anomalies, thereby ensuring broader coverage and reducing the risk of undetected events. Implemented on Arduino and ESP8266 platforms, the system consolidates sensor data to provide real-time alerts for leaks and vandalism. Simulated in Proteus under conditions representative of Nigeria's environment, the prototype achieved an excellent functional correctness, demonstrating its reliability and scalability. By combining multi-sensor fusion, edge-level processing, and affordable hardware, this work contributes a practical pathway toward intelligent pipeline security in Nigeria's oil and gas sector.

The rest of this paper proceeds as follows: Section 2 outlines the design methodology and system architecture, Section 3 presents the simulation outcomes, and Section 4 offers conclusions, implications, and directions for future research.

2. Methodology

This section details the design, components, and operational logic of the proposed IoT-based pipeline security system.

2.1. Research Stages

The development of the proposed system followed a structured workflow to ensure technical feasibility and logical accuracy. The stages represented in a flowchart of Figure 1 are summarized below:

1) **Requirement Analysis**: Identification of critical sensor types (gas, vibration, thermal) suitable for pipeline threat detection.
2) **System Architecture Design:** Mapping the data flow from physical sensors to the IoT cloud interface.
3) **Hardware Selection and Circuit Design**: Selecting COTS (Commercial Off-The-Shelf) components and designing the circuit schematic in Proteus.
4) **Firmware Development**: Coding the thresholding algorithms and Wi-Fi communication protocols in C++.
5) **Simulation & Validation:** Testing the system response under idealized conditions to verify functional correctness.

2.2. Selecting a System Architecture

The system architecture illustrated in Figure 2 is structured in four layers: (1) The Sensing Layer, which acquires physical data; (2) The Network/Communication Layer, which transmits data; (3) The Data Processing/Cloud Layer (ThingSpeak), which analyzes and stores data; and (4) The Application/User Interface Layer, which displays alerts to the end-user.

A cost-effective and scalable hub-and-spoke architecture was developed for field deployment. In this model, a central hub equipped with a 4G-enabled WiFi router serves as the communication gateway, while individual sensor nodes—positioned within a 150 m radius—connect as WiFi clients. This configuration reduces the need for multiple cellular modules, thereby minimizing overall system cost.

Figure 2. Conceptual block diagram of the IoT system architecture.

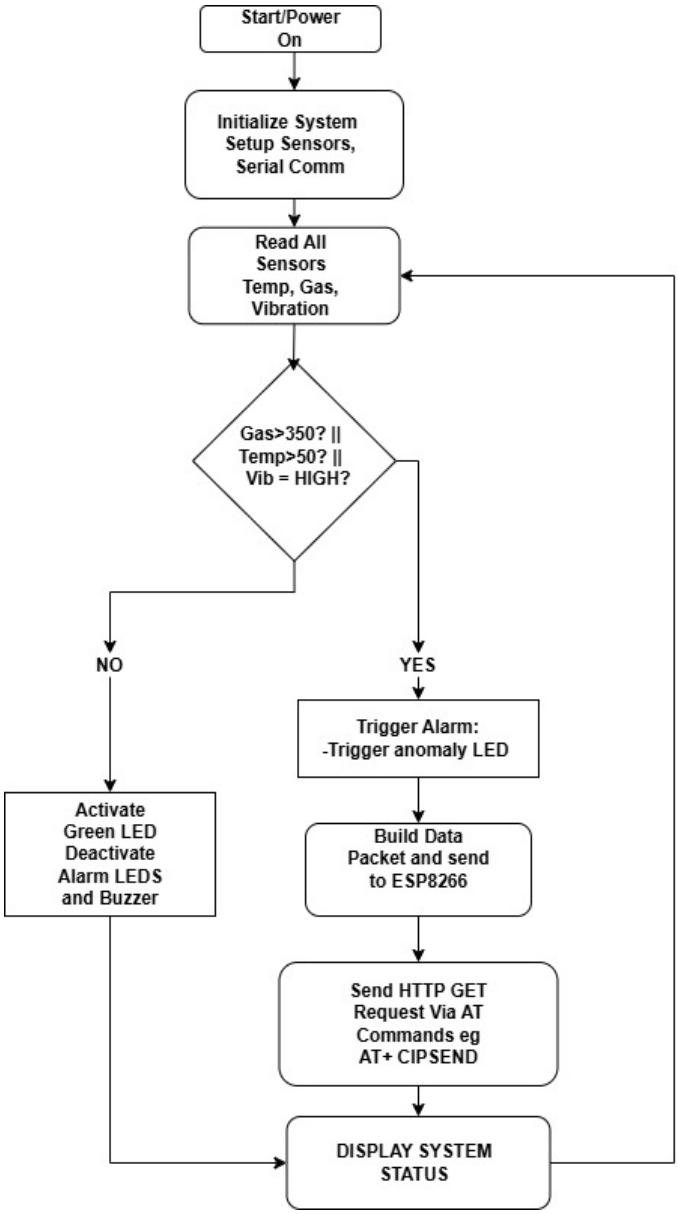

Figure 3. Flowchart of the system's operational logic.

2.3. Simulated Hardware Components

The core of each sensor node consists of low-cost, off-the-shelf components:

- **Microcontroller:** An Arduino Uno serves as the local processing unit, responsible for data acquisition and executing threshold-based logic.
- **Sensor Suite:** A multi-modal sensor suite was integrated to detect distinct vandalism signatures:
- **Gas Sensor (MQ-2):** Detects combustible gas leaks. Its resistance (Rs) decreases in the presence of hydrocarbons, increasing the output voltage (V_{out}). The sensor and a load resistor (RL) form a voltage divider. The Arduino's 10-bit ADC converts V_{out} (0-5V) to an integer 'ADC_Value' (0-1023).

The output voltage is given by:

$$V_{out} = V_{cc}\frac{R_L}{R_s + R_L} \quad (1)$$

The ADC value is:

$$ADC\_Value = V_{out} \times \frac{1023}{5v} \quad (2)$$

By combining these, the sensor's resistance Rs can be derived directly from the ADC reading: An alert threshold was set at an ADC_Value > 350 (corresponding to V_{out} > 1.71V), indicating a significant gas concentration. The threshold was set slightly above the calculated clean-air baseline (≈ 341 ADC for $R_{S_{clean\ air}} = 10k\Omega$) to distinguish minor environmental fluctuations from significant gas presence.

- **Temperature Sensor (LM35):** Monitors for thermal anomalies, such as fires from a breach. It provides a linear voltage output, V_{out}, directly proportional to the temperature in Celsius (TC):

$$V_{out} = 10mV/°C \times T_c \quad (3)$$

An alert threshold was set at > 50◦C (500 mV / 103 ADC), a value selected to represent an anomalous thermal event (e.g., fire, exothermic reaction) well above typical ambient operating temperatures.

- **Vibration Sensor (SW-420):** This digital (HIGH/LOW) output sensor was used to detect physical tampering, such as digging or cutting. Its operation is based on a spring–mass mechanism, with an onboard potentiometer used to set the vibration sensitivity (acceleration threshold). When external vibration exceeds this limit, the LM393 comparator outputs a digital HIGH signal. During simulation, the sensitivity was not numerically defined; instead, the output was manually toggled from LOW to HIGH to test the microcontroller's response.
- **Communication Module:** An ESP8266 WiFi module, connected to the Arduino's UART, handles data transmission to the cloud platform via TCP/IP.

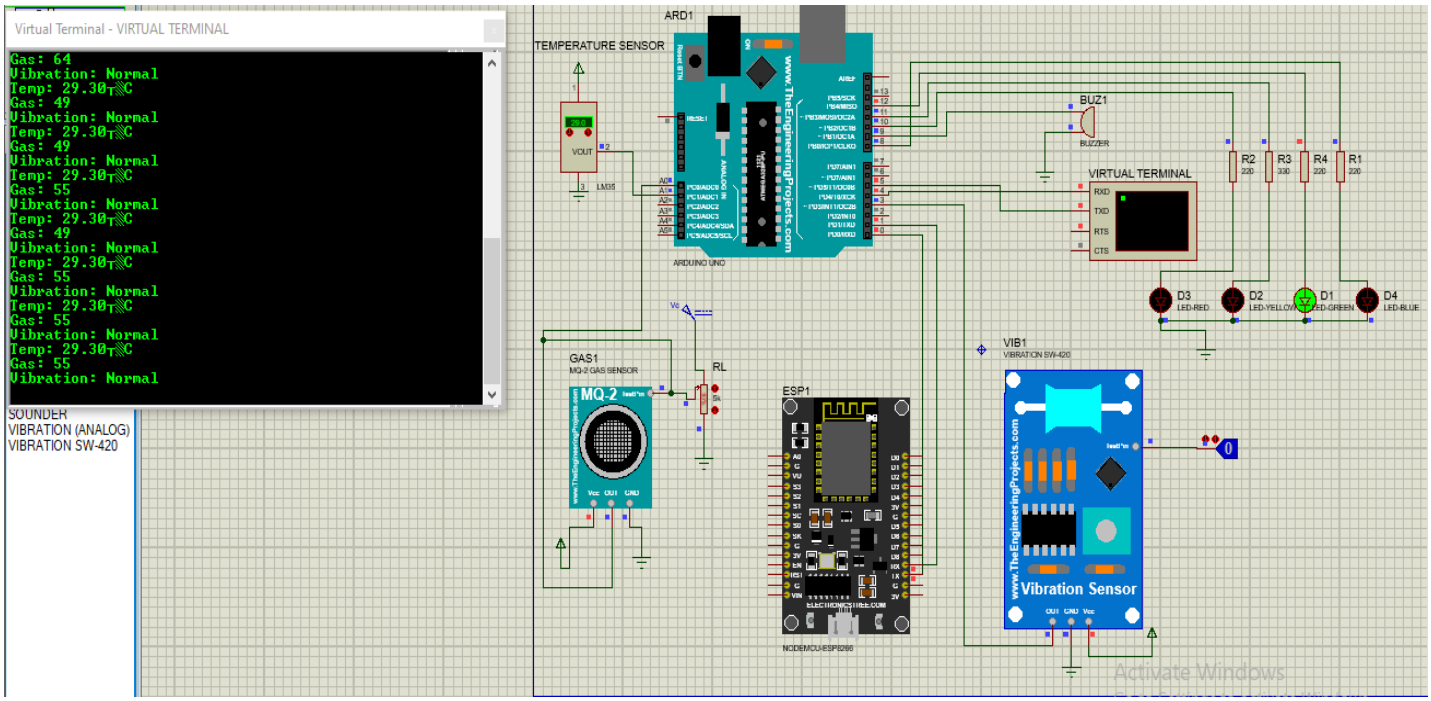

Figure 4. System in normal operating conditions. The simulation shows the green 'OK' LED (D1) is active, the buzzer is off, and the virtual terminal(left) displays sensor values that are all below their alarm thresholds, confirming baseline stability.

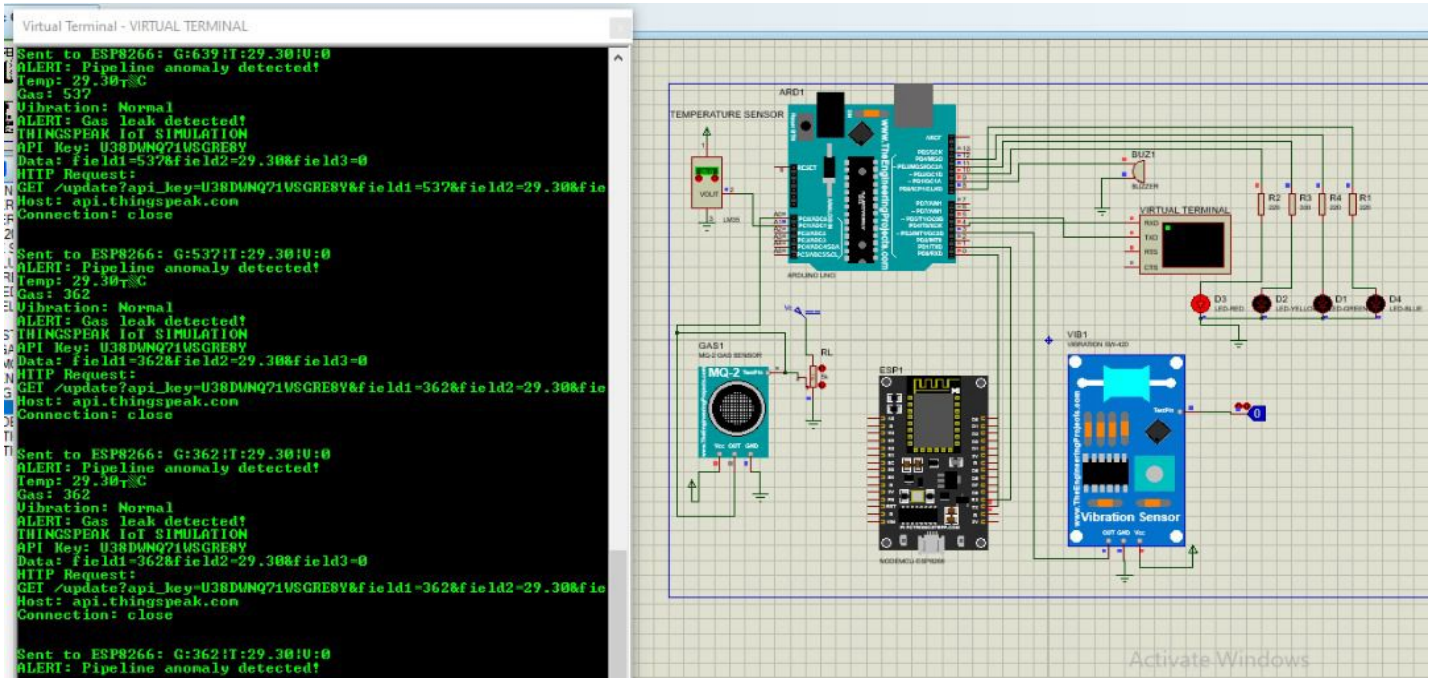

Figure 5. System response to a simulated gas leak. A sensor reading above the 350 ADC threshold (e.g., 362 ADC) successfully triggers the red 'Gas Alert' LED (D3), the audible buzzer, and a corresponding "ALERT: Gas leak detected!" message on the virtual terminal (left).

- **Local Alerts:** An active buzzer and colored LEDs (Red, Yellow, Blue, and Green) provide immediate on-site visual and audible alerts upon anomaly detection.

2.4. Software Design and Anomaly Detection

The system firmware was developed in C++ using the Arduino IDE. The logic, detailed in the system flowchart (Figure 3), follows an alert-driven protocol to conserve power. In the setup routine, all sensor pins (input) and alert pins (output) are initialized, along with serial communication (9600 baud) to the ESP8266 module.

The main loop function continuously performs local data acquisition and processing:

1) It reads the digital state from the SW-420, the analog value from the MQ-2, and the voltage from the LM35 (converting it to ◦C).

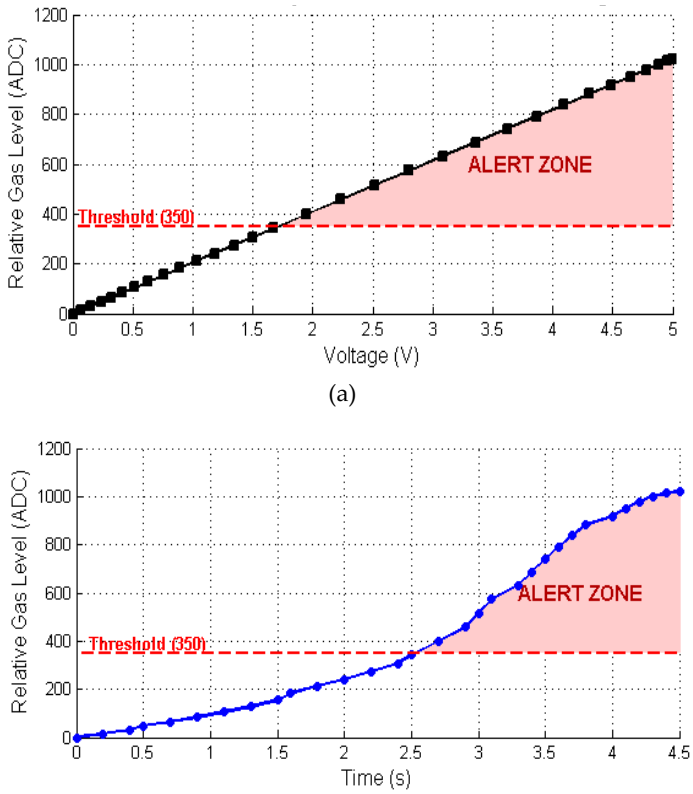

Figure 6. (a) Sensor Sensitivity: Relative Gas Level vs voltage, (b) System Response: Relative Gas Level vs Time.

2) These readings are compared against the predefined thresholds (Vibration = HIGH, Gas > 350 ADC, Temp > 50◦C).
3) If any threshold is breached, the triggerAlarm() function is called, activating the local buzzer and LEDs.
4) Simultaneously, the sendData() function is executed, which formats a data packet (e.g., "G:310|T:26.00|V:1") and transmits it via the ESP8266 to the ThingSpeak cloud platform, triggering a remote alert.
5) If no thresholds are breached, the system remains in a low-power monitoring state, and no data is transmitted.

The complete system, including the circuit design and firmware logic, was designed and validated using the Proteus Design Suite. The simulation environment was used to test the system's response to various scenarios, including normal operation (all sensor values within thresholds) and multiple anomaly conditions (simulated high gas, high temperature, and vibration). This allowed for thorough testing and validation of the circuit and code logic before physical prototyping.

3. Results and Discussions

The system's performance was validated by simulating three scenarios in the Proteus environment: normal operation, a gas leak, fire incidents, and a physical intrusion (vibration).

3.1. Normal Operating Conditions

Under normal operating conditions, all simulated sensor inputs remained below their respective alarm thresholds (Gas < 350 ADC, Temperature < 50 °C, Vibration = LOW). As illustrated in Figure 4, the system maintained a stable steady-state operation throughout the simulation.

During testing, incremental adjustments were made to the simulated gas sensor voltage while observing the live readings on the virtual terminal. The system accurately recorded varying gas concentrations (in ADC values) without triggering any false alarms. Throughout this phase, the green "OK" LED (D1) remained illuminated, the buzzer was inactive, and the virtual terminal continuously displayed normal sensor readings (e.g., "Gas: 49 | Temp: 29.3 °C | Vibration: Normal"). No alert packets were generated or transmitted, confirming the system's baseline stability and reliability under normal environmental conditions.

3.2. Anomaly Detection: Gas Leak

To simulate a gas leak, the potentiometer, which operates with the MQ-2 sensor, was adjusted until its output exceeded the 350 ADC threshold. The system responded immediately, as shown in Figure 5. The green LED turned off, the red "Gas Alert" LED (D3) illuminated, and the buzzer sounded. The virtual terminal confirmed the breach for all ADC values greater than 350 (for example, ADC Value: 362) and logged the message, "ALERT: Gas leak detected!" Additionally, the system formatted and sent the alert data packet (for example, "G: 362 | T: 29.30 | V: 0").

To evaluate system performance, a graphical analysis of the sensor response was conducted using data extracted from the simulation's virtual terminal (Figure 6a and Figure 6b). In Figure 6a (Relative Gas Level vs. Voltage), a linear regression analysis was performed to justify the fixed safety thresholds used in the firmware. The analysis revealed a sensitivity of 204.80 ADC units/Volt with a perfect linearity coefficient of $R^2 = 1.0000$. This confirms that the virtual 10-bit ADC is accurately calibrated to the 5V reference range (1024/5V = 204.8). Consequently, the heuristic threshold of 350 ADC is mathematically equivalent to 1.71V, providing a stable and predictable trigger point for anomaly detection.

Furthermore, Figure 6b (Relative Gas Level vs. Time) provides insight into the system's simulated latency. While the data shows a 100% functional detection rate, a sampling latency of approximately 200ms was observed between the gas level crossing the threshold and the software trigger at the 2.7s mark. This latency is a result of the sampling frequency set in the firmware, demonstrating that the system operates at a near-real-time resolution suitable for rapid alerting, albeit within a controlled simulation environment.

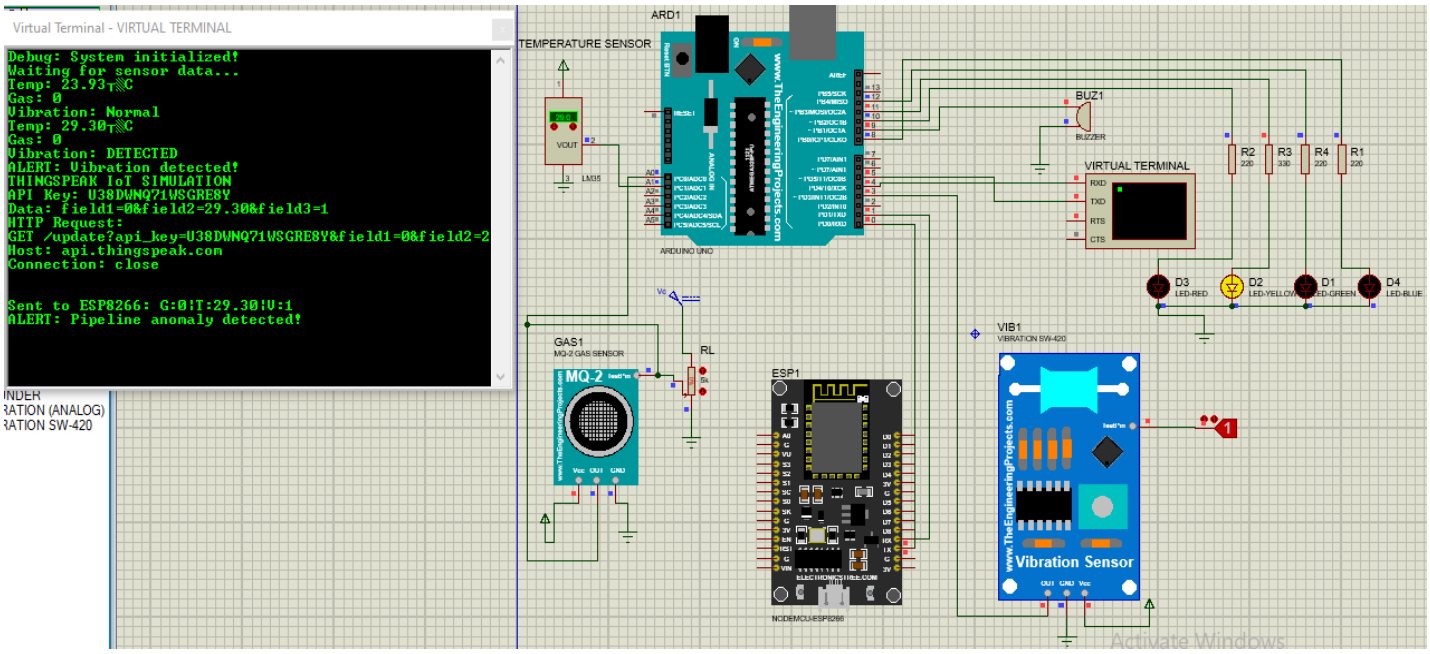

Figure 7. System response to a simulated vibration/intrusion. Activation of the vibration sensor (Logic HIGH) correctly triggers the yellow 'Vibration Alert' LED (D2), the buzzer, and an "ALERT: Vibration detected!" message, confirming the system's intrusion detection capability.

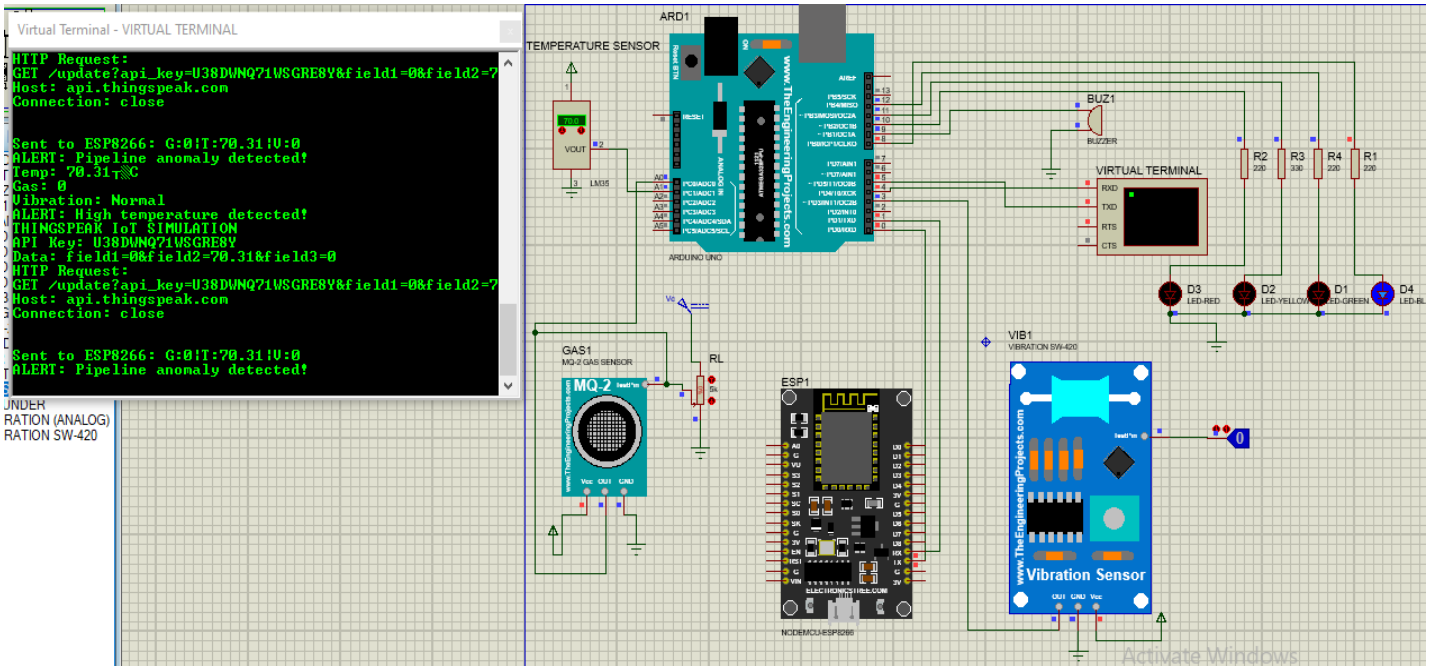

Figure 8. System response to a simulated fire/exothermic reaction. A temperature reading of 70.31 °C, which exceeds the 50 °C threshold, successfully activates the blue 'Temperature Alert' LED (D4), the buzzer, and generates a corresponding alert data packet.

3.3. Anomaly Detection: Vibration/Intrusion

To simulate a physical intrusion or mechanical tampering event, the pushbutton representing the SW-420 vibration sensor was momentarily activated. The system accurately identified this disturbance as a high-priority anomaly. As shown in Figure 7, activation of the sensor caused the green "OK" LED to turn off, while the yellow "Vibration Alert" LED (D2) illuminated, and the buzzer was triggered to provide an immediate on-site alarm.

Simultaneously, the virtual terminal displayed the warning message "ALERT: Vibration detected!" and generated a corresponding data packet formatted as "G: 0 | T: 29.30 | V: 1" for remote transmission. This demonstrates the system's capability to promptly detect and classify mechanical intrusion events, initiate a multi-layered alert response, and communicate relevant data in real time.

3.4. Anomaly Detection: Fire/Exothermic reaction

To assess the system's response to excessive heat or potential fire conditions, the simulated temperature input was deliberately raised beyond the predefined threshold

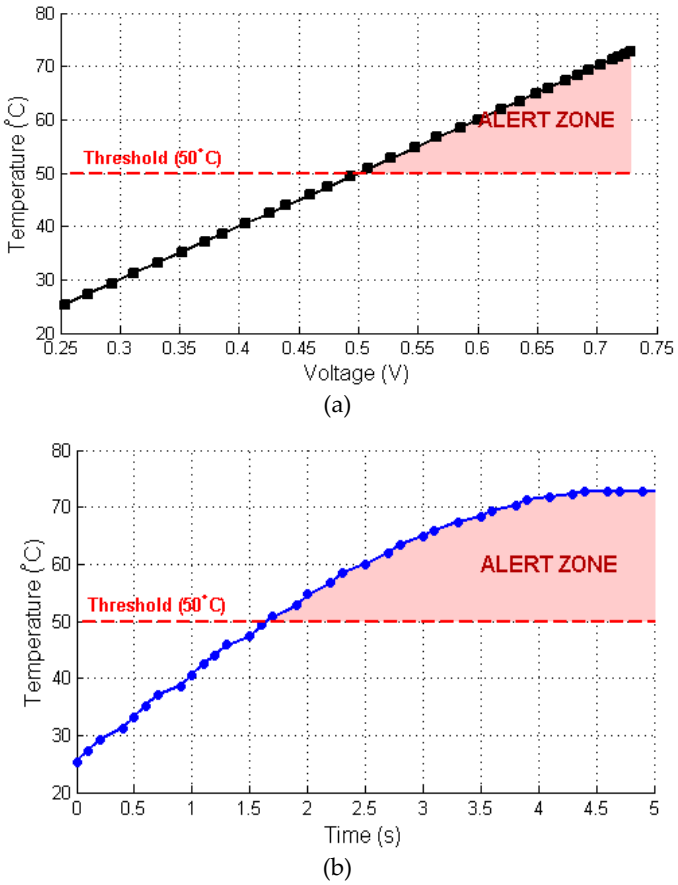

Figure 9. (a) Sensor Sensitivity: Temperature vs voltage, (b) System Response: Temperature vs Time.

of 50 °C to 70.31 °C. As shown in Figure 8, the system accurately detected this thermal anomaly and immediately initiated the programmed alert sequence.

Upon detection, the blue "Temperature Alert" LED (D4) was illuminated, the buzzer emitted an audible alarm, and a corresponding alert data packet ("G: 0 | T: 70.31 | V: 0") was generated and prepared for wireless transmission. This coordinated response demonstrates the reliability and functional integrity of the temperature monitoring and alert subsystem, confirming its effectiveness in identifying exothermic or fire-related events.

To evaluate temperature sensor performance, a graphical analysis of the thermal response was conducted using data extracted from the simulation's virtual terminal (Figure 9a and Figure 9b). The sensor calibration curve (Figure 9a exhibits a linear response, where each 0.1 V increase corresponds to a 10 °C rise ($R^2 = 1.000$), enabling the firmware to define a stable trigger threshold of 50 °C at 0.5 V. The response profile (Figure 9b) shows the temperature crossing 50 °C at approximately 1.65 s, while the alert is generated at about 1.70 s, resulting in a 50 ms detection latency. This delay is governed by the firmware sampling frequency, which is adjustable to optimize response time without introducing signal noise or jitter, thereby supporting near-real-time alerting within the simulated environment.

3.5. Communication Protocol Validation

A critical aspect of the system's validation was its ability to communicate reliably with the cloud platform. Although Proteus does not support live internet connectivity, a virtual terminal was employed to emulate and monitor serial data transmitted from the Arduino to the ESP8266 Wi-Fi module, as illustrated in Figure 10. This test scenario simulated a condition in which all three sensors—temperature, vibration, and gas—simultaneously exceeded their respective thresholds.

The simulation confirmed that, upon anomaly fields. This result verifies that the system's communication protocol and data formatting logic are functionally sound and ready for physical deployment.

The system's reliability was evaluated based on its functional correctness under controlled simulation conditions. The detection thresholds (e.g., Gas > 350 ADC, Temp > 50 °C) were established as heuristic baselines to trigger alerts. While these values proved effective in simulation, they represent a proof-of-concept and would require field calibration to account for environmental variability.

3.6. Discussions

The simulation results presented in Section 3 confirm both the technical feasibility and functional correctness of the proposed IoT-based pipeline security system. This section evaluates the findings, their implications, and the limitations of the study.

The achieved 100% functional correctness in the simulation validates that the core system architecture—comprising a multi-modal sensor array integrated with an Arduino-based edge processor—is sound and effective. The system successfully demonstrated its ability to (1) differentiate between normal operating conditions and genuine anomaly events (such as gas leakage, mechanical vibration, and abnormal temperature) and (2) trigger a multi-layered response mechanism that includes immediate on-site alerts and remote notifications. This dual-alert structure is strategically important: local alarms can deter vandals in real time, while remote alerts ensure that a coordinated security response is activated promptly.

A key outcome of this work is the validation of the edge-processing and alert-driven protocol. By executing anomaly detection and thresholding locally on the Arduino and transmitting data only when a confirmed event occurs, the design significantly reduces power consumption and data bandwidth usage. These characteristics are critical for low-cost, field-deployable solutions in Nigeria's oil and gas regions, where both power supply and cellular network reliability are often constrained.

This study contributes a practical and economically scalable solution to the field. While previous research—such as Bello et al. [16], Odulaja and Rufai [17], and Agbolade et al. [18]—has explored IoT and WSN approaches to pipeline monitoring, many of these rely on high-cost

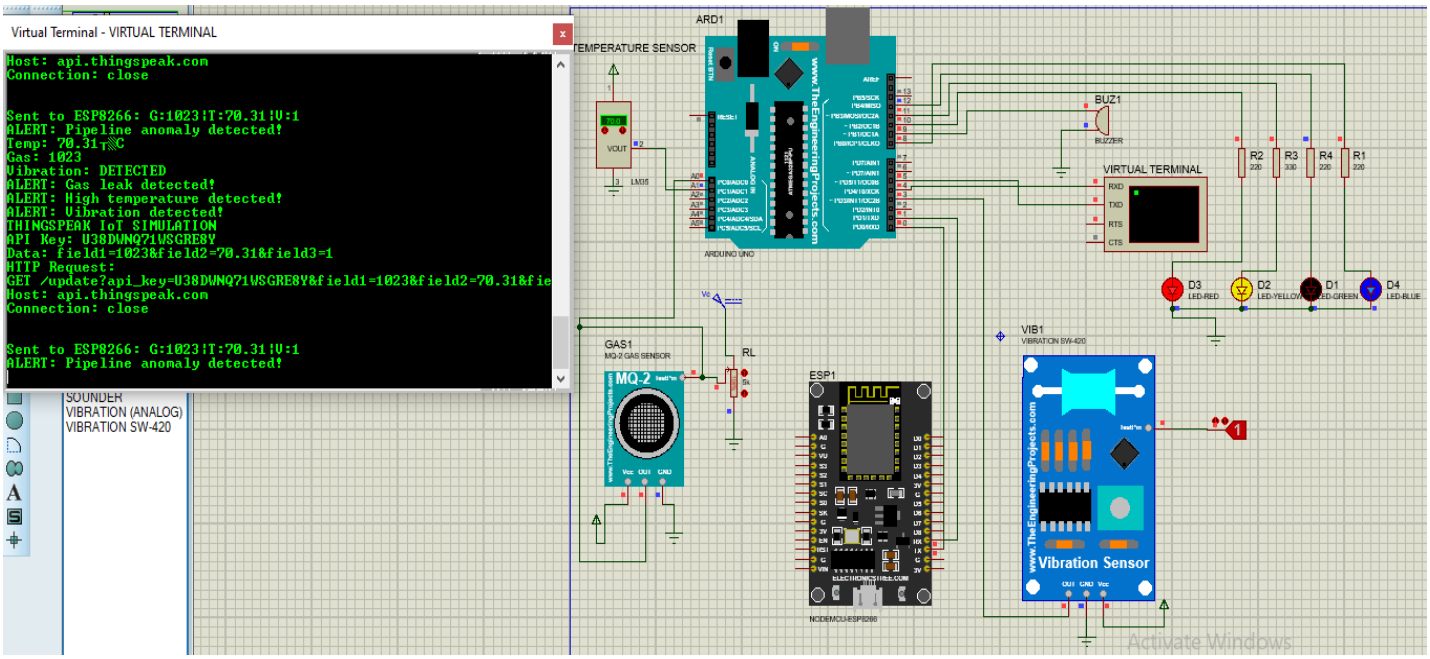

Figure 10. Validation of the HTTP communication protocol format. The virtual terminal monitor displays the serial output, showing the firmware correctly constructing the HTTP GET request. This request is properly formatted with the ThingSpeak API key and sensor data fields, verifying the data packet structure for remote transmission.

Table 1. Comparative Analysis of Related Works and Position of Current Study.

Study	Focus Area	Sensor/Technology Used	Limitations	Contribution of Current Work
Bello et al. [16]	IoT/WSN for leak & vandalism detection	Single sensor + WSN	Limited to single-parameter sensing	Introduces multi-modal sensing (gas, vibration, temperature) for improved detection accuracy
Odulaja & Rufai [17]	Expert system for vandalization detection	Generic sensor + alert module	Manual threshold setup; limited integration	Implements automated thresholding and anomaly discrimination
Agbolade et al. [18]	LoRaWAN leak detection	Pressure and flow sensors	Leak-only detection; no intrusion analysis	Adds vibration sensing for mechanical intrusion detection
Bukie et al. [19]	IoT-based leak topology monitoring	Multi-sensor setup	Focused on cloud data handling; limited on-site intelligence	Incorporates edge-level decision logic for real-time-like response
Effiom et al. [20]	Real-time leak monitoring	GSM-based IoT	No multi-sensor integration	Expands monitoring scope through sensor fusion and local processing
Ojo et al. [21]	IoT gas leakage detection	MQ sensors + Wi-Fi	Limited to gas detection	Extends functionality to vibration and temperature sensing
Ezeja & Nwobi [22]	WSN-based pipeline monitoring	Wireless sensor nodes	No automated analytics	Adds local automation and intelligent decision-making
Okorodudu et al. [23]	Conceptual Niger Delta monitoring	Conceptual (non-IoT)	No implementation or communication framework	Provides a validated, IoT-enabled prototype

(e.g., DAS-based) or high-bandwidth (e.g., 4G video streaming) technologies, which limit their deployment feasibility. In contrast, this work demonstrates how low-cost, off-the-shelf components (such as the Arduino Uno, ESP8266 Wi-Fi module, MQ-2 gas sensor, and vibration/temperature sensors) can effectively be combined to create a multi-sensor fusion system with intelligent decision-making at the edge.

Moreover, the system's hub-and-spoke Wi-Fi architecture offers a pathway for scalable deployment, allowing multiple sensor nodes to communicate with a local access point or gateway before relaying critical data to the cloud. This hybrid model balances local responsiveness with remote data accessibility, enhancing reliability under field conditions.

Table 1 presents a comparative analysis of key related works, emphasizing how the present study advances current knowledge through its multi-modal sensing, edge analytics, and alert-driven communication approach.

4. Conclusion

This research has successfully designed and validated, through simulation, a low-cost, multi-modal Internet of Things (IoT)-based system for pipeline security in Nigeria. The system integrates gas, vibration, and temperature sensors with a local microcontroller and cloud communication, providing a dual-layer alert mechanism. The simulation results demonstrated 100% functional correctness in detecting anomalies and triggering appropriate alarms. The design emphasizes low-cost, off-the-shelf components, edge processing, and a scalable network architecture, making it a technically feasible and economically viable solution to combat the ongoing challenges of pipeline vandalism. While further validation through physical prototyping and field testing is required, this study serves as a solid proof of concept for enhancing the security and integrity of Nigeria's critical pipeline infrastructure.

5. Limitations and Future Work

The current study represents a simulated prototype focused on logical validation and communication protocol formatting. A critical limitation is the absence of physical environmental stressors such as network latency, signal attenuation, or power-consumption profiles typical of remote Nigerian terrain. Future research will focus on transitioning from software simulation to physical hardware deployment to test the system's sensitivity under real-world noise and varying climatic conditions.

6. Declarations

6.1. Author Contributions

Evander Chika Udeh: Methodology, Software, Investigation, Formal analysis, Resources, Data Curation, Writing - Original Draft; **Michael Chukwuebuka Agwu:** Resources, Data Curation,Writing - Original Draft; **Pamilerin Samuel Akirinde:** Resources, Writing - Original Draft**; Nnaemeka Sunday Ugwuanyi**: Conceptualization, Writing -Original draft, Review & Editing, Formal analysis, Visualization, Validation, Supervision, Project administration. **Akudo Ogechi Nwogu:** Review & Editing.

6.2. Institutional Review Board Statement

Not applicable.

6.3. Informed Consent Statement

Not applicable.

6.4. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.5. Acknowledgment

Not applicable.

6.6. Conflicts of Interest

The authors declare no conflicts of interest.

7. References

[1] L. Olu-Adeyemi, "The Political Ecology of Oil Pipeline Vandalism in Nigeria," *International Journal of Research and Innovation in Social Science*, vol. 4, no. 5, pp. 239–245, 2020. [Online]. Available: https://rsisinternational.org/journals/ijriss/Digital-Library/volume-4-issue-5/239-245.pdf.

[2] S. A. Edun, T. K. Olaniyi, and K. Lawani, "Modelling the Implications of Oil Pipeline Vandalism on the Nigeria Economy: A Case Study of Niger Delta Region," *International Journal of Innovative Business Strategies*, vol. 9, no. 2, 2023, https://doi.org/10.20533/ijibs.2046.3626.2023.0075.

[3] F. I. Johnson, R. Laing, B. Bjeirmi, and M. Leon, "Examining the causes and impacts of pipeline disasters in Nigeria," *AIMS Environmental Science*, vol. 9, no. 5, pp. 636–657, 2022, https://doi.org/10.3934/environsci.2022037.

[4] M. M. Nuhu, E. R. Rene, and A. Ishaq, "Remediation of crude oil spill sites in Nigeria: Problems, technologies, and future prospects," *Environmental Quality Management*, 2021, https://doi.org/10.1002/tqem.21793.

[5] V. A. Semawon and N. A. Eduvwie, "Petroleum pipeline fire outbreak in communities of Delta State, Nigeria," *International Journal of Innovative Research and Development*, vol. 11, no. 4, p. 111, 2022, https://doi.org/10.24940/ijird/2022/v11/i4/APR22030.

[6] M. M. Umar *et al*., "Enhancing Security and Efficiency in IoT-Based Oil & Gas Pipeline Monitoring Systems with a Novel Lightweight Cryptography Framework," *in International Conference on Computing and Advances in Information Technology (ICCAIT 2023)*, 2023. [Online]. Available: https://iccait.com.ng/wp-content/uploads/2025/01/56.pdf.

[7] M. Yıldırım, U. Demiroğlu, and B. Şenol, "An in-depth exam of IoT, IoT core components, IoT layers, and attack types," *European Journal of Science and Technology*, no. 28, pp. 665–669, 2021, https://doi.org/10.31590/ejosat.1010023.

[8] E. N. Aba *et al*., "Petroleum pipeline monitoring using an internet of things (IoT) platform," *SN Applied Sciences*, vol. 3, no. 180, 2021, https://doi.org/10.3934/environsci.2022037.

[9] B. F. Ekeu-Wei and I. T. Ekeu-Wei, "Development of a Low-Cost Prototype System for Pipeline Operational and Vandalism Spillage Detection and Validation Framework," *Advances in Internet of Things*, vol. 14, no. 2, pp. 21–35, 2024, https://doi.org/10.4236/ait.2024.142002.

[10] M. A. Zurkanain and S. K. Subramaniam, "Investigation and Implementation of IoT-Based Oil & Gas Pipeline Monitoring System," *International Journal of Recent Technology and Applied Science*, vol. 5, no. 1, 2023, https://doi.org/10.36079/lamintang.ijortas-0501.477.

[11] O. C. Nwokonkwo *et al*., "Machine Learning Framework for Real-Time Pipeline Anomaly Detection and Maintenance Needs Forecast Using Random Forest and Prophet Model," *Automation, Control and Intelligent Systems*, vol. 12, no. 2, pp. 25–34, 2024, https://doi.org/10.11648/j.acis.20241202.11.

[12] V. A. Parjane, T. Arjariya, and M. Gangwar, "Corrosion Detection and Prediction for Underwater Pipelines Using IoT and Machine Learning Techniques," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 2S, pp. 162–167, 2023. [Online]. Available: https://www.ijisae.org/index.php/IJISAE/article/view/2626.

[13] S. S. Aljameel *et al*., "An Anomaly Detection Model for Oil and Gas Pipelines Using Machine Learning," *Computation*, vol. 10, no. 8, p. 138, 2022, https://doi.org/10.3390/computation10080138.

[14] N. R. Mulla, K. Kazi and S. Liyakat, "IoT sensors to monitor pipeline pressure and flow rate combined with ML-algorithms to detect leakages," *Recent Trends in Fluid Mechanics*, vol. 12, no. 2, pp. 40–48, 2025

[15] S. Ahmed, F. Le Mouél, and N. Stouls, "Resilient IoT-based monitoring system for crude oil pipelines," in *Proc. 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, pp. 1–7, 2020, https://doi.org/10.1109/IOTSMS52051.2020.9340197.

[16] S. Bello, M. D. Amadi, and A. H. Rawayau, "Internet of Things-Based Wireless Sensor Network System for Early Detection and Prevention of Vandalism/Leakage on Pipeline Installations in the Oil and Gas Industry in Nigeria," *Fudma Journal of Sciences*, vol. 7, 2023, https://doi.org/10.33003/fjs-2023-0705-1927.

[17] G. O. Odulaja and K. I. Rufai, "Remotely-Monitored Anti-Pipeline Vandalization Detection Expert System," *Journal of Science and Information Technology*, vol. 16, no. 1, pp. 79–88, 2021. [Online]. Available: https://journals.tasued.edu.ng/index.php/josit/article/view/34.

[18] O. A. Agbolade *et al*., "A LoRaWAN-Based IoT System for Leakage Detection in Pipelines," *European Journal of Engineering and Technology Research*, vol. 8, no. 5, 2023, https://doi.org/10.24018/ejeng.2023.8.5.3078.

[19] P. T. Bukie, I. E. Eteng, and E. E. Essien, "Development of Internet of Things-Based Petroleum Pipeline Topology Leak Monitoring and Detection System Using Sensors," *Journal of the Nigerian Society of Physical Sciences*, vol. 7, no. 4, art. 2407, 2025, https://doi.org/10.46481/jnsps.2025.2407.

[20] S. O. Effiom, G. A. Fischer, and E. J. Akpama, "Novel System Design Model for an IoT-Based Real-Time Oil and Gas Pipeline Leakage Monitoring System," *International Journal of Engineering and Technology*, vol. 13, no. 2, 2024. [Online]. Available: https://www.sciencepubco.com/index.php/ijet/article/view/32875.

[21] P. O. Ojo *et al*., "IoT-Based Gas Leakage Detection and Monitoring System," *Journal of Science and Information Technology*, vol. 19, no. 1, 2025. [Online]. Available: https://journals.tasued.edu.ng/index.php/josit/article/view/198.

[22] O. M. Ezeja and C. G. Nwobi, "A Fuel Pipeline Monitoring and Security System Using Wireless Sensor Networks (WSN)," *Nigerian Journal of Technology*, vol. 43, no. 3, 2023. [Online]. Available: https://www.nijotech.com/index.php/nijotech/article/view/3808.

[23] F. O. Okorodudu, P. O. Okorodudu, and L. O. Atumah, "A Monitoring System for Petroleum Pipeline Vandalism in the Niger Delta," *International Journal of Research – Granthaalayah*, vol. 6, no. 6, 2018, https://doi.org/10.29121/granthaalayah.v6.i6.2018.1359.

[24] S. R. G. Sumetha and R. Praba, "Water Pipeline Leakage Detection Using IoT," *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, vol. 13, no. 3, Mar. 2025, https://doi.org/10.17148/IJIREEICE.2025.13342.

[25] G. S. Kuaban *et al*., "Energy performance of Internet of Things (IoT) networks for pipeline monitoring," in *Proc. 20th Int. Wireless Commun. & Mobile Comput. Conf. (IWCMC)*, Ayia Napa, Cyprus, Jul. 2024, pp. 1490–1497, https://doi.org/10.1109/IWCMC61514.2024.10592530.