"AI-Powered Fault Diagnosis System for Industrial Equipment"

Asst prof. Dr. G S Pushpalatha
Electronics and Communication Engineering
Dr. Ambedkar Institute of Technology, VTU
Bengaluru, India
pushpalathags.ec@drait.edu.in

Thrishul Gowda B N
Electronics and Communication Engineering
Dr. Ambedkar Institute of Technology, VTU
thrishulgymnast123@gmail.com

Sahana S
Electronics and Communication Engineering
Dr. Ambedkar Institute of Technology, VTU
Bengaluru, India
sahana.sj10@gmail.com

Amrutha M A
Electronics and Communication Engineering
Dr. Ambedkar Institute of Technology, VTU
Bengaluru, India
amruthama1969@gmail.com

Shilpa S
Electronics and Communication Engineering
Dr. Ambedkar Institute of Technology, VTU
Bengaluru, India
sshilpa78912@gmail.com

Abstract— **This paper presents an IoT-based AI-powered fault diagnosis system for real-time monitoring of industrial equipment. The proposed system uses an ESP32 microcontroller interfaced with vibration, temperature, and current sensors to continuously acquire machine condition data. The collected data are transmitted to a Firebase Realtime Database and processed using a Flask-based backend. A cloud-hosted web dashboard provides real-time visualization and fault alerts. Threshold-based anomaly detection is employed to identify abnormal operating conditions at an early stage. The system is low-cost, scalable, and remotely accessible, making it suitable for small- and medium-scale industries. Experimental results demonstrate effective fault detection with minimal latency, supporting predictive maintenance and improving equipment reliability.**

Keywords— IoT, Fault Diagnosis, ESP32, Firebase, Predictive Maintenance

I. INTRODUCTION

Industrial equipment is subjected to continuous mechanical, thermal, and electrical stress, leading to performance degradation and unexpected failures. Conventional maintenance methods such as reactive and preventive maintenance are inefficient and costly. Recent advancements in the Internet of Things (IoT) enable real-time monitoring of machine health using sensor data. This paper proposes an AI-powered IoT-based fault diagnosis system using an ESP32 microcontroller, cloud storage, and a web-based dashboard to detect abnormal operating conditions early and support predictive maintenance in industrial environments

II. PROBLEM STATEMENT

Industrial and laboratory machinery operates under varying loads and environmental conditions, which often results in mechanical wear, temperature fluctuations, and electrical faults. Traditional maintenance approaches such as reactive maintenance cause unplanned downtime, while preventive maintenance often leads to unnecessary servicing and increased operational costs. Manual inspection methods are time-consuming, prone to human error, and incapable of detecting early-stage faults. Furthermore, existing industrial monitoring systems are expensive, complex, and unsuitable for adoption by small- and medium-scale industries.

The absence of real-time monitoring and remote accessibility limits timely fault detection and corrective action. Therefore, there is a need for an integrated, low-cost, and scalable fault diagnosis system capable of continuously monitoring machine parameters and providing real-time alerts.

III. LITERATURE REVIEW

Industrial equipment monitoring has evolved significantly with the adoption of Internet of Things (IoT) and Artificial Intelligence (AI) technologies. Early research primarily focused on multi-sensor fault detection and cloud-based monitoring, demonstrating that real-time data acquisition improves machine health assessment and operational reliability. The integration of edge–cloud collaborative frameworks and on-device intelligence has further enhanced predictive maintenance by reducing latency and enabling low-power local computation. Real-time visualization and alerting mechanisms have also been shown to improve response time to anomalies, while standardization and interoperability remain critical challenges for integrating heterogeneous industrial devices.

Zhang *et al.* [1] proposed an unsupervised fault detection system for conveyor systems using multi-sensor data fusion. By combining vibration, temperature, and current signals and applying unsupervised learning techniques, the system successfully detected anomalies without requiring labeled datasets. The study demonstrated that sensor fusion improves fault detection accuracy and enables early identification of potential failures, thereby reducing unplanned downtime.

Chakraborty *et al.* [2] introduced a predictive maintenance architecture based on edge–cloud collaboration. In their approach, preliminary data processing was performed at the edge, while computationally intensive analytics were handled in the cloud. AI-based fault prediction algorithms enabled early detection of failures, reduced communication latency, and optimized bandwidth usage. The study highlighted the effectiveness of combining edge computing and AI for responsive and efficient industrial maintenance systems.

Patel and Kumar [3] presented a cloud-based condition monitoring system for industrial motors using IoT sensors to collect vibration, temperature, and current data. The cloud analytics platform enabled remote monitoring and early fault detection without the need for on-site inspection.

While the system demonstrated scalability and efficiency, it also highlighted the dependency on reliable internet connectivity for continuous operation.

Verma and Singh [4] developed a multi-sensor machine health monitoring system that integrated vibration, temperature, and current sensing. Their results showed that combining multiple operational parameters provides a more comprehensive view of equipment health and improves fault detection accuracy. The study emphasized the importance of multi-sensor integration for reliable predictive maintenance and reduced unexpected downtime.

Brown and Lee [5] discussed the importance of standardization and interoperability in industrial IoT systems. Their work emphasized that standardized communication protocols and unified frameworks are essential for seamless integration of sensors, AI models, and cloud platforms. Such standardization supports scalable deployment and improves system reliability in industrial environments.

Kumar and Sharma [6] proposed a real-time industrial equipment monitoring system using IoT sensors and cloud-based visualization. Continuous data acquisition and real-time alerts enabled timely fault detection and improved operational efficiency. The study also highlighted that cloud-based monitoring architectures provide a strong foundation for future AI-driven predictive maintenance applications.

Liu *et al.* [7] explored low-power on-device predictive maintenance using embedded machine learning and edge AI. Their system performed local fault prediction, reducing reliance on cloud computation and improving response time. The study demonstrated the feasibility of deploying energy-efficient fault diagnosis solutions on resource-constrained industrial devices.

Reddy *et al.* [8] implemented an ESP32-based industrial data monitoring system with Firebase cloud visualization. The system was cost-effective, scalable, and easy to deploy, enabling real-time remote monitoring of industrial equipment. The use of dashboards and alerts provided actionable insights for operators, highlighting the practicality of ESP32-based IoT solutions.

Roy *et al.* [9] developed a real-time visualization and alerting framework for industrial IoT applications using cloud-based dashboards. Their results showed that intuitive visualization and real-time alerts significantly improve situational awareness and maintenance decision-making, emphasizing the role of user interfaces in predictive maintenance systems.

Wang *et al.* [10] investigated multi-sensor data fusion techniques that combine vibration, temperature, and current signals to enhance diagnostic accuracy and fault localization. Their findings demonstrated that integrated sensing approaches are effective in detecting early signs of equipment degradation. Additionally, Chen *et al.* [11] highlighted the importance of data integrity and security in cloud-based IoT systems, emphasizing encryption, authentication, and access control mechanisms to protect sensitive industrial data.

Overall, the existing literature demonstrates that IoT-based multi-sensor monitoring, combined with cloud and edge intelligence, provides an effective foundation for real-time fault diagnosis and predictive maintenance. However, many existing solutions are either complex or costly. The proposed work builds upon these studies by developing a low-cost, ESP32-based AI-powered fault diagnosis system that integrates multi-sensor monitoring, cloud storage, backend analytics, and real-time visualization to support proactive industrial maintenance.

IV. METHODOLOGY

A. *Hardware:*

The system uses an **ESP32 microcontroller** interfaced with vibration, temperature, and current sensors to continuously acquire machine operating parameters in real time.

B. *Software:*

The ESP32 is programmed using **Arduino IDE** to transmit sensor data to a **Firebase Realtime Database**, while a **Flask-based backend** processes the data and a web dashboard provides real-time visualization.

C. *Logic:*

Sensor readings are compared with predefined threshold values to detect abnormal conditions, and fault alerts are generated and displayed on the dashboard for timely maintenance action.

D. *Block Diagram:*

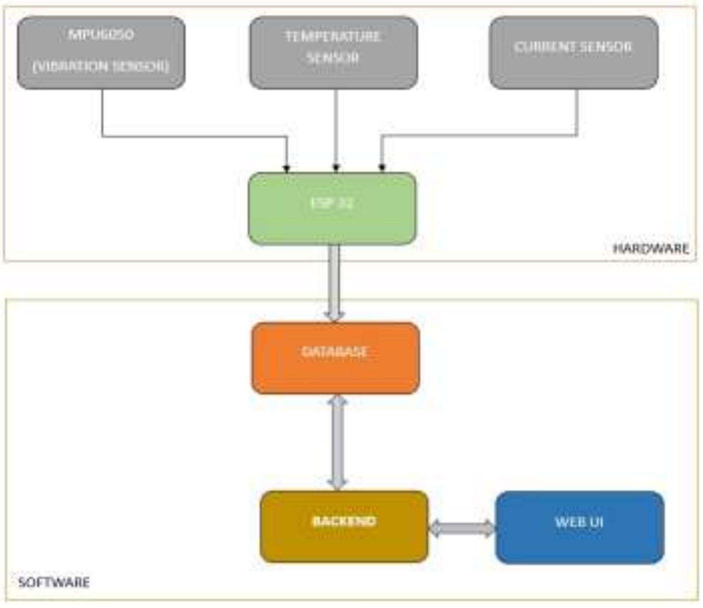


Fig.1: Block diagram of "AI-Powered Fault Diagnosis System for Industrial Equipment

This Fig.1 illustrates the complete workflow of the system, from data acquisition to visualization. It shows the sensors connected to ESP32, sending data to Firebase cloud storage, then to the Flask backend, and finally to a web dashboard for real-time monitoring. It explains both hardware and software flow clearly.

System Components and Flow:

1. Sensor Data Acquisition: Vibration, temperature, and current sensors continuously measure machine operating parameters.
2. Embedded Processing: The ESP32 microcontroller reads sensor data, performs basic filtering, and prepares the data for transmission.
3. Wireless Data Transmission: Processed sensor data is transmitted to the cloud using Wi-Fi and HTTP communication.
4. Cloud Data Storage: Sensor readings are stored in the Firebase Realtime Database with real-time synchronization.
5. Backend Data Processing: A Flask-based backend retrieves cloud data and evaluates it using threshold-based fault detection logic.

6. Visualization and Alerts: A web-based dashboard displays real-time graphs and generates alerts when abnormal conditions are detected.
7. User Monitoring: Maintenance personnel remotely monitor machine health and take preventive action.

V. OPERATIONAL LOGIC

1. The system initializes the ESP32 microcontroller, sensors, and Wi-Fi connection.
2. Vibration, temperature, and current sensors continuously collect machine operating data.
3. The ESP32 processes sensor readings and transmits them to the Firebase cloud database.
4. The backend server retrieves real-time data from the cloud.
5. Sensor values are compared with predefined threshold limits.
6. If values remain within limits, the system continues normal monitoring.
7. If any parameter exceeds its threshold, a fault condition is detected. Fault alerts and real-time status are displayed.

Hardware Requirements:

Sl. No.	Component	Specification	Purpose
1	ESP32 Microcontroller	Dual-core MCU with integrated Wi-Fi and Bluetooth	Data acquisition, processing, and wireless transmission
2	Vibration Sensor (MPU6050 / SW-420)	3-axis accelerometer with digital output	Detect mechanical vibration and imbalance
3	Temperature Sensor (LM35 / DS18B20)	Analog/Digital temperature sensor	Monitor machine temperature and overheating
4	Current Sensor (ACS712)	Hall-effect based current sensor	Measure current consumption and detect overload
5	Firebase Realtime Database	Cloud-based NoSQL database	Store and synchronize sensor data in real time
6	Flask Framework	Lightweight Python web framework	Backend processing and fault detection
7	Web Dashboard	HTML, CSS, JavaScript	Real-time visualization and alert display
8	Power Supply	5V regulated DC source	Provide stable power to hardware components

Software Tools:

Sl. No.	Software	Purpose
1	Arduino IDE	Programming the ESP32 microcontroller
2	Firebase Realtime Database	Cloud data storage and synchronization
3	Flask Framework	Backend data processing and logic
4	Python	Backend development and data handling
5	HTML, CSS, JavaScript	Web dashboard development
6	Render Cloud Platform	Hosting backend and dashboard

VI. IMPLEMENTATION

The implementation of the proposed AI-powered fault diagnosis system was carried out by integrating embedded hardware, wireless communication, cloud computing, backend processing, and web-based visualization into a unified IoT framework. The system was designed to operate continuously and reliably under real-time industrial conditions, ensuring accurate monitoring and timely fault detection. The core hardware component of the system is the ESP32 microcontroller, which was selected due to its dual-core processing capability, integrated Wi-Fi module, low power consumption, and suitability for real-time IoT applications. The ESP32 was interfaced with vibration, temperature, and current sensors to acquire essential machine health parameters.

During system initialization, the ESP32 establishes a Wi-Fi connection and configures communication with the Firebase Realtime Database using HTTP-based REST APIs. Sensor data acquisition is performed continuously, where vibration data is captured using a MEMS-based sensor, temperature values are obtained from a thermal sensor, and electrical current variations are measured using a Hall-effect-based current sensor. The ESP32 reads analog and digital sensor outputs through its built-in analog-to-digital converter and communication interfaces. To improve data reliability, preliminary filtering is applied at the microcontroller level to reduce noise and transient fluctuations before transmission.

The processed sensor data is formatted into structured JSON packets and transmitted wirelessly to the Firebase Realtime Database. Firebase acts as the central cloud repository, enabling real-time synchronization and reliable storage of sensor readings. Each parameter is updated continuously with timestamped values, allowing both real-time monitoring and historical analysis. The use of Firebase eliminates the need for a dedicated server infrastructure while ensuring scalability and low-latency data access.

A Flask-based backend application serves as the intermediate processing layer between the cloud database and the user interface. The backend periodically retrieves sensor data from Firebase and performs threshold-based evaluation to determine the operational status of the equipment. Predefined safe operating limits are used to identify abnormal conditions such as excessive vibration, overheating, or abnormal current draw. When any parameter exceeds its threshold, the backend flags a fault condition and updates the system status accordingly.

This logic enables early detection of faults while maintaining computational simplicity and efficiency. The processed data and fault status are presented to the user through a web-based dashboard developed using HTML, CSS, and JavaScript. The dashboard is hosted on the Render cloud platform, ensuring remote accessibility and continuous availability. Real-time graphs display sensor parameters dynamically, allowing maintenance personnel to observe trends and identify deviations instantly. Visual alerts are generated when fault conditions are detected, enabling prompt corrective action. The dashboard interface is responsive and accessible from any internet-enabled device, supporting remote monitoring in distributed industrial environments.

To validate system performance, the complete implementation was tested under controlled operating conditions by introducing variations in vibration, temperature, and current parameters. The system demonstrated reliable data acquisition, stable cloud synchronization, and accurate fault detection with minimal latency. The ESP32 maintained continuous operation and successfully re-established connectivity in the event of temporary network disruptions. Overall, the implementation confirms that the proposed system provides a practical, low-cost, and scalable solution for real-time industrial equipment monitoring and fault diagnosis.

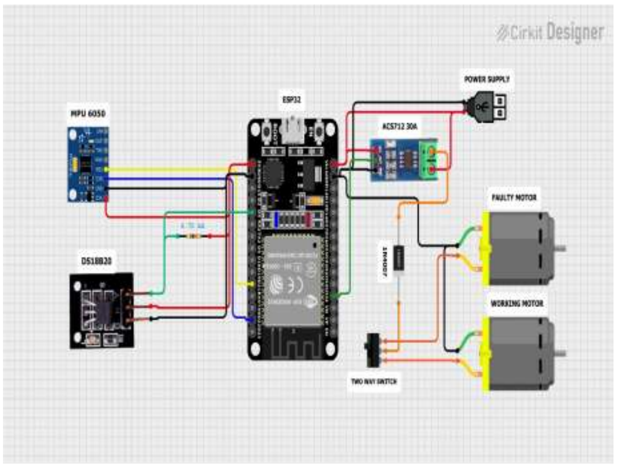


Fig.2: Circuit Design of AI-Powered Fault Diagnosis System for Industrial Equipment

VII. RESULTS AND DISCUSSION

The proposed AI-powered IoT-based fault diagnosis system was implemented and evaluated under controlled operating conditions to verify its effectiveness for real-time industrial monitoring. The ESP32 microcontroller reliably acquired vibration, temperature, and current data and transmitted the readings to the Firebase Realtime Database with low communication latency. The cloud infrastructure maintained stable synchronization, and the Flask-based backend processed incoming data efficiently, enabling continuous system operation without interruptions. The web-based dashboard accurately reflected real-time sensor values, confirming correct end-to-end data flow from sensing to visualization.

Under normal operating conditions, the monitored parameters remained within predefined threshold limits, and the dashboard displayed stable trends with smooth variations. When abnormal conditions were introduced by increasing load, inducing vibration, or elevating temperature, the system promptly detected deviations from normal ranges.

Fault conditions were identified immediately after threshold violations, and corresponding alerts were displayed on the dashboard in real time. This behavior demonstrates the effectiveness of the threshold-based detection approach for early fault identification.

System responsiveness was assessed by observing the delay between fault occurrence and alert visualization. The measured latency was minimal, indicating that the integration of ESP32, Firebase, and Flask is suitable for real-time industrial applications. Wireless communication reliability was also validated, as the system maintained consistent data transmission and successfully re-established connectivity after temporary network interruptions without loss of critical information.

The results indicate that the proposed system provides a cost-effective and scalable alternative to conventional industrial monitoring solutions. By utilizing low-cost sensors, a Wi-Fi-enabled microcontroller, and cloud-based services, the system achieves reliable fault detection without the complexity and expense of traditional diagnostic equipment. Although the current implementation relies on threshold-based logic, the observed performance confirms its suitability for real-time monitoring and early-stage fault detection. Furthermore, the

architecture supports future enhancements, including the integration of machine learning models for predictive maintenance.
Overall, the experimental evaluation validates that the proposed system delivers accurate real-time monitoring, timely fault detection, and reliable remote accessibility, making it suitable for deployment in small- and medium-scale industrial environments.

.

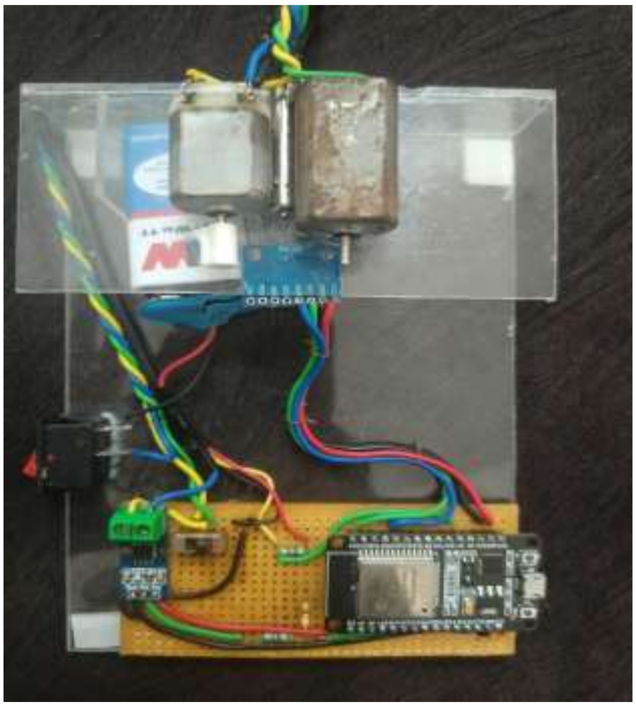
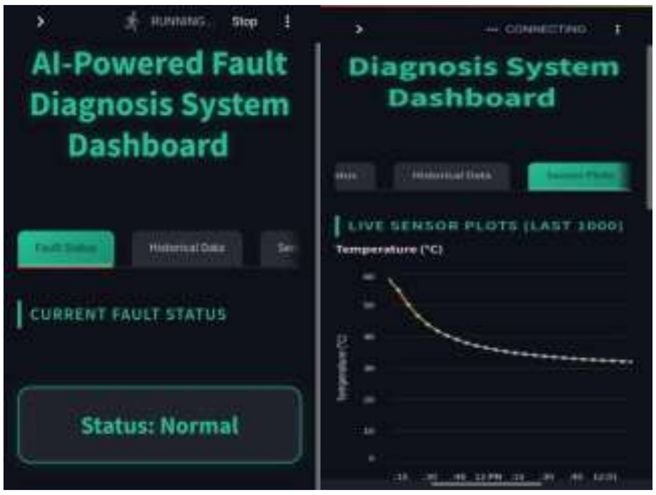
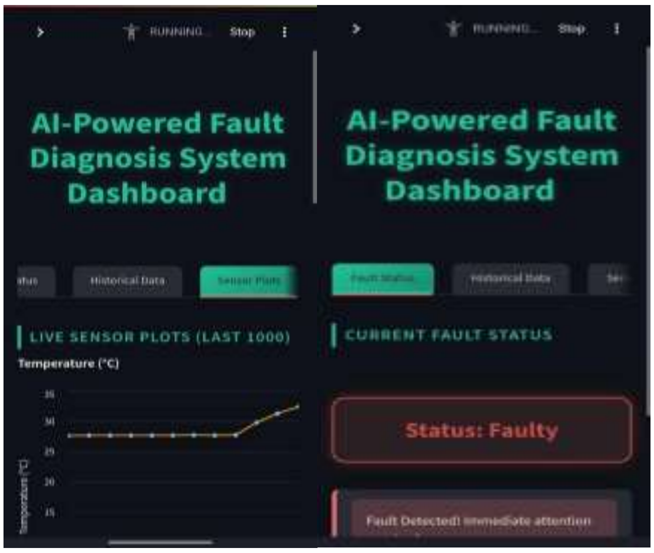


Fig.3: AI-Powered Fault Diagnosis System for Industrial Equipment

This Fig.3 presents the overall developed system, including hardware setup, cloud connectivity, and data pipeline. It visually summarizes how the ESP32 collects real-time data, sends it to Firebase, processes it through Flask, and displays it on the dashboard for fault analysis.




RUNNING Stop
AI-Powered Fault Diagnosis System Dashboard
Historical Data
Sensor Plots
LIVE SENSOR PLOTS (LAST 1000)
Temperature (°C)
Fault Status
CURRENT FAULT STATUS
Status: Faulty
Fault Detected! Immediate attention
Fig.4: Fault Status Dashboard

This Fig.4 displays the graphical user interface (GUI) of the developed dashboard used for real- time machine health monitoring. The dashboard retrieves live temperature, vibration, and current parameters from Firebase through the Flask backend and presents them in a structured, user-friendly format. Each sensor parameter is updated dynamically, enabling continuous supervision of the equipment.

The interface includes multiple widgets such as numerical indicators, colored status bars, and time-series graphs that visualize the trend of each sensor signal. A dedicated status panel performs machine-learning-based fault classification, displaying whether the equipment is operating in "Normal Condition" or "Fault Detected" mode. The system also generates warning messages when any sensor value exceeds the predefined threshold limits.

In addition, the dashboard offers clear color-coded cues—such as green for safe conditions and red for critical states—allowing operators to make quick decisions. The purpose of this dashboard is to provide intuitive, real-time visibility into machine health so that faults can be identified early, reducing downtime and improving maintenance efficiency.

VIII. CONCLUSION

This paper presented an AI-powered IoT-based fault diagnosis system for real-time monitoring of industrial equipment. The proposed system integrates vibration, temperature, and current sensors with an ESP32 microcontroller, cloud-based data storage using Firebase Realtime Database, a Flask-based backend, and a web-based visualization platform. By enabling continuous monitoring and threshold-based fault detection, the system effectively identifies abnormal operating conditions at an early stage and supports timely maintenance decisions.

Experimental evaluation demonstrated reliable data acquisition, low-latency cloud communication, accurate fault detection, and stable real-time visualization. The use of low-cost hardware components and cloud services makes the system economically viable and scalable, particularly for small- and medium-scale industrial environments. The results confirm that the proposed approach provides a practical alternative to conventional maintenance strategies by reducing downtime, improving equipment reliability, and supporting predictive maintenance practices.

Overall, the developed system aligns with Industry 4.0 principles by combining embedded sensing, IoT connectivity, cloud computing, and web-based analytics into a unified framework. The proposed architecture also offers flexibility for future enhancements, including the integration of machine learning techniques for advanced fault prediction and automated maintenance planning, thereby extending its applicability to more complex industrial scenarios.

REFERENCES

[1] F. Zhang, M. Wu, and L. Wang, "Unsupervised fault detection of industrial systems using sensor fusion techniques," *IEEE Access*, vol. 7, pp. 92883–92892, 2019.

[2] A. Chakraborty, B. Roy, and P. Bera, "IoT-enabled predictive maintenance architecture using edge and cloud collaboration," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7206–7214, Aug. 2020.

[3] M. Patel and P. Kumar, "Cloud-based condition monitoring and fault detection in industrial motors using IoT," *Int. J. Eng. Res. Technol.*, vol. 9, no. 7, pp. 145–150, 2020.

[4] S. Verma and R. Singh, "IoT-based multi-sensor data acquisition system for machine health monitoring," *IEEE Sensors Journal*, vol. 20, no. 14, pp. 8012–8020, Jul. 2020.

[5] P. Zhang, G. Li, and Y. Wang, "Real-time industrial equipment monitoring based on IoT and cloud computing," *IEEE Access*, vol. 6, pp. 73835–73845, 2018.

[6] R. Kumar and S. Sharma, "Design and implementation of IoT-based real-time industrial equipment monitoring system," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 4120–4128, May 2021.

[7] Y. Liu, D. Zhang, and K. Wu, "Low-power on-device machine learning for real-time predictive maintenance," *IEEE Embedded Systems Letters*, vol. 13, no. 1, pp. 15–18, Mar. 2021.

[8] N. S. Reddy, R. Kiran, and H. S. Rao, "Implementation of ESP32-based industrial data monitoring and cloud visualization using Firebase," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 10, no. 8, pp. 5520–5527, 2021.

[9] A. Roy, M. Das, and T. Dey, "Real-time data visualization and alerting using Firebase for industrial IoT systems," in *Proc. Int. Conf. Smart Electronics and Communication (ICOSEC)*, 2021, pp. 112–118.

[10] T. Wang, J. Wan, D. Li, and C. Zhang, "Implementing machine learning for industrial fault diagnosis in IoT environments," *IEEE Network*, vol. 33, no. 5, pp. 64–69, Sep./Oct.