



# IOT-Based Machine Health Monitoring System

Chavan Rutuja Umesh<sup>1</sup>, Babar Dnyaneshwari Namdev<sup>2</sup>, Gajare Pragati Dnyaneshwar<sup>3</sup>,  
Mrs. P. S. Gavali<sup>4</sup>

UG Student, Department of Computer Science and Engineering, SKN Sinhgad College of Engineering, Pandharpur,  
India<sup>1</sup>

UG Student, Department of Computer Science and Engineering, SKN Sinhgad College of Engineering, Pandharpur,  
India<sup>2</sup>

UG Student, Department of Computer Science and Engineering, SKN Sinhgad College of Engineering, Pandharpur,  
India<sup>3</sup>

Assistant Professor. Computer Science and Engineering, SKN Sinhgad College Of Engineering Korti Pandharpur<sup>4</sup>

**Abstract:** Motors, pumps, compressors, generators and conveyors underpin modern manufacturing, but each can break down without warning, leading to unplanned stoppages, expensive repairs and growing maintenance costs. This paper describes an Internet of Things (IOT) system that continuously supervises such equipment. An ESP32 microcontroller reads temperature and vibration sensors and transmits the readings over Wi-Fi to a FastAPI backend, which stores them in a MySQL database and serves them to a ReactJS dashboard. Each reading is evaluated against limits configured for the individual machine, and the machine is classified as Healthy, Warning or Critical. A Critical state triggers a local buzzer and a remote notification through Blynk Cloud. In addition, a machine learning module examines the stored sensor history to anticipate probable faults and so supports predictive maintenance. The result is a low-cost, intelligent approach to industrial condition monitoring.

**Keywords:** IOT, Predictive Maintenance, Machine Health Monitoring, Machine Learning, ESP32

## I. INTRODUCTION

Industrial machines work under changing loads and environmental conditions, and over time these stresses show up as wear, overheating and abnormal vibration. Most plants still rely on inspections carried out at fixed intervals, which leaves a window in which a developing fault can progress unseen until the next visit. Continuous acquisition and analysis of machine data closes this window and allows equipment to be maintained before it fails.

This paper proposes a monitoring system built for that purpose. An ESP32 microcontroller samples temperature and vibration sensors and forwards the values over Wi-Fi. A FastAPI service processes the incoming data and records it in a MySQL database, and a React dashboard shows each machine as Healthy, Warning or Critical. Critical conditions are signalled through a buzzer and Blynk notifications, while a machine learning component draws on the accumulated history to forecast faults and assist maintenance planning.

## II. RELATED WORK

The reviewed literature is organised under four themes: (i) the shift from manual to IOT-based monitoring, (ii) sensor-microcontroller monitoring systems, (iii) temperature and vibration as indicators of machine condition, and (iv) the platforms on which such systems are commonly built.

### A. From Scheduled Servicing to IOT-Based Monitoring

Machine maintenance has gradually moved from manual, calendar-driven servicing to intelligent IOT-enabled monitoring. Calendar-driven servicing depends on human intervention and cannot ensure that emerging faults are noticed early. Embedded systems later made it possible to log machine parameters automatically, but limited remote access restricted their value. Recent work therefore emphasises continuous monitoring with remote access and improved fault detection, aiming to reduce both downtime and maintenance cost.

### B. Sensor and Microcontroller-Based Monitoring Systems

Several studies pair sensors with microcontrollers to observe machines remotely. Sharma and Patel [1] present a predictive maintenance system for industrial machines in which an ESP32 is combined with machine learning to predict



failures. Kumar et al. [2] examine machine learning techniques for constructing predictive maintenance models in smart manufacturing and apply the Random Forest algorithm to failure prediction. Ali et al. [3] describe a cloud-connected ESP32 approach that enables real-time observation of machines and remote access to their data.

### C. Temperature and Vibration as Condition Indicators

Temperature and vibration are widely regarded as informative signs of machine condition. Reddy and Iyer [4] report real-time condition monitoring built on these two quantities to detect irregular behaviour. Wang et al. [5] propose a predictive maintenance model that joins IOT with artificial intelligence and highlight the importance of sensing in such schemes. Patel et al. [6] evaluate an ESP32-based industrial monitoring arrangement with wireless communication and show that it can be flexible as well as remotely accessible.

### D. Enabling Platforms

The supporting technologies are described in their own vendor and developer documentation. Espressif Systems documents the ESP32 platform, which provides data acquisition and wireless connectivity [7]. FastAPI can act as the backend that receives sensor data and exchanges it with a database [8]. Blynk Cloud supports remote observation and the delivery of notifications when abnormal conditions occur [9], and React can be used to build the front-end dashboard showing monitored parameters and health status [10].

Taken together, these works indicate that IOT, wireless links, cloud services, web technologies and machine learning can be combined into an effective predictive maintenance solution. They also point to a need for flexible systems that handle several machine types and apply predefined limits to each monitored parameter. Table I relates the reviewed work to the proposed system.

TABLE I  
MAPPING OF LITERATURE TO THE PROPOSED SYSTEM

| Reference | Focus                                                             | Contribution to Proposed System                                                    |
|-----------|-------------------------------------------------------------------|------------------------------------------------------------------------------------|
| [1]–[3]   | ESP32-based predictive maintenance and cloud-connected monitoring | Supports pairing an ESP32 with machine learning and with cloud-based remote access |
| [4]       | Temperature and vibration as condition indicators                 | Justifies the choice of the two sensed parameters                                  |
| [5]       | IOT and AI predictive maintenance model                           | Supports a sensing-driven prediction layer for failure forecasting                 |
| [6]       | ESP32-based industrial monitoring with wireless communication     | Supports a flexible, remotely reachable wireless design                            |
| [7]       | ESP32 platform documentation                                      | Basis for data acquisition and Wi-Fi connectivity                                  |
| [8]       | FastAPI documentation                                             | Basis for the backend that receives and stores readings                            |
| [9]       | Blynk Cloud documentation                                         | Basis for remote observation and push alerts                                       |
| [10]      | React documentation                                               | Basis for the dashboard showing readings and health status                         |

## III. PROBLEM STATEMENT

Industrial machines rarely announce a fault before it causes a stoppage, and periodic manual inspection cannot guarantee that a developing problem is caught between visits. Operators therefore need continuous, remotely accessible information about each machine, a clear indication of its condition, and timely warning when that condition deteriorates.

**Problem Statement:** To design an IOT-based health monitoring system for industrial machines that uses temperature and vibration sensors, supports several machine types with adjustable operating limits, raises alarms, keeps historical records, and predicts machine failure through machine learning.

### Key Challenges Addressed by the Proposed System:

- **Late Fault Detection:** replaces fixed-interval inspection with continuous observation.
- **Remote Visibility:** makes machine data available on a web dashboard.



- **Machine Diversity:** accommodates several machine types, each with its own adjustable limits.
- **Unattended Alerting:** signals critical states through a buzzer and Blynk notifications without manual checking.
- **Record Keeping:** retains sensor history in a database.
- **Reactive Maintenance:** uses machine learning on stored history to anticipate failures.

#### IV. OBJECTIVES

The main objective is to design and build an IOT-based system that observes the temperature and vibration of industrial machines in real time using an ESP32. The specific aims are:

- to present data for different machine types on a web dashboard so that status can be followed remotely;
- to label each machine as Healthy, Warning or Critical and to sound a buzzer when a Critical state is reached;
- to apply machine learning to anticipate failures and help prevent them; and
- to deliver a dependable, intelligent monitoring solution that improves the operation of industrial machinery.

#### V. PROPOSED SYSTEM AND METHODOLOGY

The proposed system supervises machine health by combining sensors, an ESP32, IOT communication, web technologies and machine learning. Its components are grouped as follows.

##### A. System Components

- **Sensing:** temperature sensor (DS18B20/LM35) and vibration sensor (SW-420).
- **Edge unit:** ESP32 microcontroller, which reaches the backend over Wi-Fi.
- **Backend and storage:** FastAPI backend and MySQL database.
- **Presentation and alerts:** React web dashboard and Blynk Cloud.
- **Analytics:** machine learning module.

##### B. Operational Workflow

1. The sensors measure the temperature and vibration of the machine.
2. The ESP32 processes the readings and sends them over the internet.
3. The FastAPI backend receives the readings and saves them in the MySQL database.
4. The readings are compared with the thresholds defined for that machine.
5. Based on the comparison, the machine is classified as Healthy, Warning or Critical.
6. The React dashboard displays the machine data together with its health status.
7. For a Critical status, the buzzer is activated and a Blynk Cloud notification is sent.
8. The machine learning module analyses the sensor history to anticipate possible failures so that maintenance can be carried out beforehand.

#### VI. IMPLEMENTATION AND ARCHITECTURE

The hardware consists of an ESP32 microcontroller, temperature and vibration sensors, and a buzzer connected to the ESP32, which together track machine condition in real time. Over Wi-Fi, the ESP32 uploads readings to a server built with the FastAPI framework; the server saves them in a MySQL database and supplies them to the React application on the user's device. The application displays the uploaded data and derives each machine's health condition from the threshold values set for it. When a fault condition arises, the buzzer sounds and a Blynk notification reaches the user, and the machine learning model uses the stored data to predict machine failure. Fig. 1 shows the overall methodology of the system.

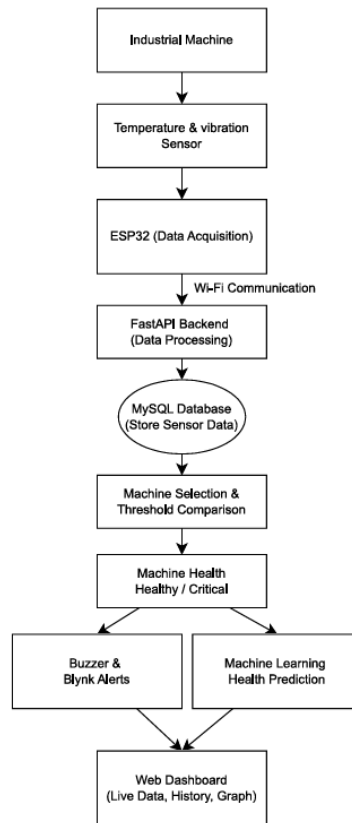


Fig. 1 Methodology of the Proposed System

## VII. RESULTS

The implemented system was able to follow machine parameters in real time. Normal operation is presented clearly, and abnormal readings lead to alerts without manual checking, which reduces the manual effort involved in supervision. The observations reported here are qualitative; this paper does not provide numerical measures of response time or prediction accuracy.

### A. Normal Condition

While temperature and vibration remain within their predefined safe limits, the dashboard continues to show live values and no alert is generated, indicating that the machine is running in a stable state (Fig. 2).



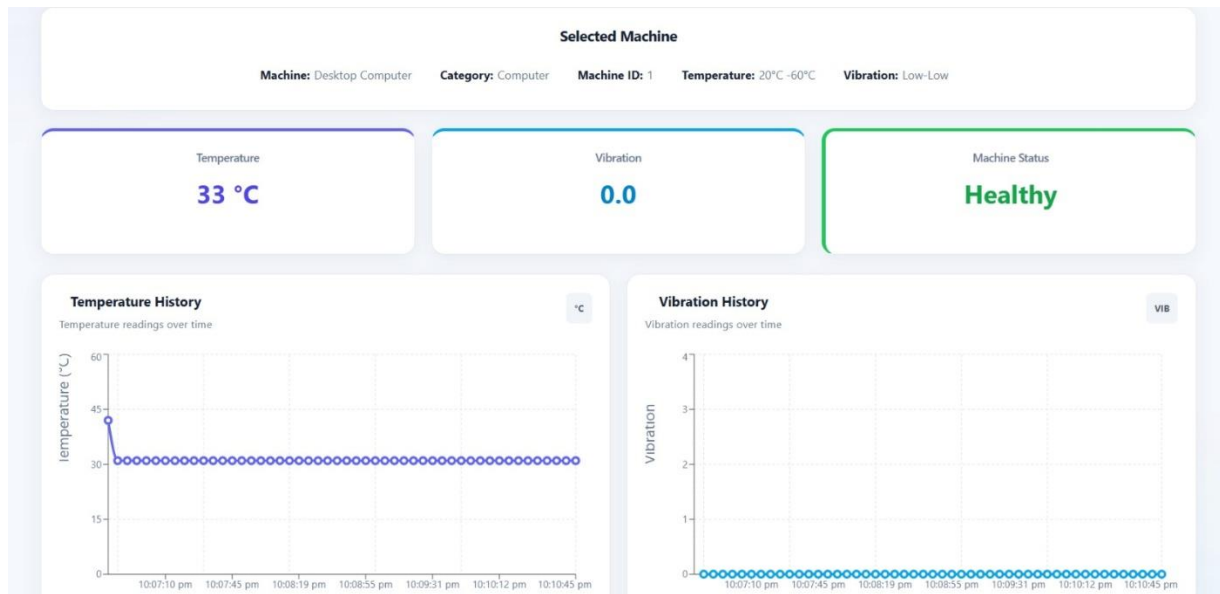


Fig. 2 Dashboard View under Normal Condition

### B. Abnormal Condition

When a sensor value crosses its threshold, the fault is detected and the system triggers the buzzer or a notification (Fig. 3). Such early detection can help prevent damage to the machine and shorten downtime.



Fig. 3 Dashboard View under Abnormal Condition

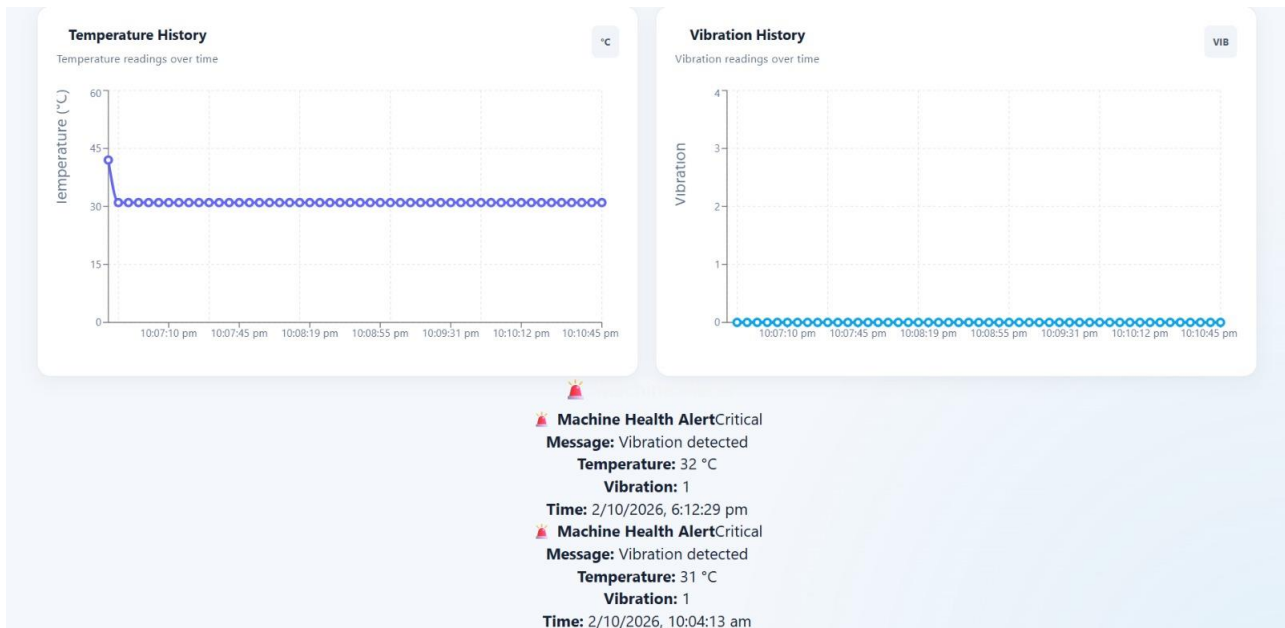


Fig. 3 Dashboard View under History

## VIII. FEASIBILITY ANALYSIS AND DISCUSSION

### A. Technical Feasibility

The system uses readily available components: an ESP32 with temperature and vibration sensors at the edge, a FastAPI backend with a MySQL database, a React dashboard, and Blynk Cloud for notifications. Because these are widely documented platforms [7]–[10], the design can be reproduced and extended.

### B. Economic Feasibility

The system is positioned as a low-cost route to industrial condition monitoring. Automatic alerting also reduces the manual effort needed for supervision. No cost figures were measured in this work, so the economic benefit is described qualitatively.

### C. Operational Feasibility

Thresholds are set per machine, and the dashboard presents each machine's state as Healthy, Warning or Critical, so operators can follow status remotely without taking readings by hand.

## IX. CONCLUSION AND FUTURE SCOPE

This paper presented an IOT-based machine health monitoring system built around an ESP32 and temperature and vibration sensors. By combining threshold-based status classification, alerting and machine learning, the system provides early warning of irregularities and predicts failures, making it an efficient and economical option for smart monitoring of industrial machines.

Future work may proceed in the following directions:

- adding further sensors and newer machine learning techniques;
- developing a mobile application for remote supervision;
- integrating additional cloud services and automated maintenance recommendations; and
- adapting the system to other kinds of industrial machines and to large-scale Industry 4.0 production environments.

## REFERENCES

- [1]. R. Sharma and A. Patel, "IOT-based predictive maintenance system for industrial machines using machine learning," IEEE Access, vol. 13, pp. 15421–15435, 2025.
- [2]. S. Kumar, P. Verma, and R. Singh, "Machine learning techniques for predictive maintenance in smart manufacturing," IEEE Internet Things J., vol. 11, no. 4, pp. 4251–4264, 2024.



- [3]. M. Ali, F. Khan, and S. Ahmed, "Industrial IOT-based machine health monitoring using ESP32 and cloud computing," in Proc. Int. Conf. Smart Computing and Communication (ICSCC), IEEE, 2024, pp. 112–120.
- [4]. K. Reddy and V. Iyer, "Real-time machine condition monitoring using temperature and vibration sensors," Int. J. Eng. Res. Technol. (IJERT), vol. 12, no. 6, pp. 45–52, 2023.
- [5]. Y. Wang, H. Chen, and J. Liu, "Predictive maintenance framework for industrial equipment based on IOT and artificial intelligence," Sensors, vol. 23, no. 8, p. 3812, 2023.
- [6]. D. Patel, R. Mehta, and P. Shah, "ESP32-based smart industrial monitoring system with wireless data communication," Int. J. Sci. Res. Comput. Sci., Eng. Inf. Technol., vol. 8, no. 3, pp. 210–218, 2022.
- [7]. Espressif Systems, "ESP32 Series Technical Reference Manual," Shanghai, China, 2024.
- [8]. FastAPI Development Team, "FastAPI: A modern, high-performance web framework for building APIs with Python," FastAPI documentation, 2024.
- [9]. Blynk Inc., "Blynk IOT platform for remote monitoring and notification in industrial applications," Blynk documentation, 2024.
- [10]. React Development Team, "React: A JavaScript library for building interactive user interfaces," Meta Platforms documentation, 2024.