MORPH-X: Low-Cost Machine Behaviour Monitoring and Vibration Anomaly Detection Using ESP32 And MPU6050

Ms. B. Sangeetha Priya[1], Balamurugan S[2], Jayashree M[3], Kamalesh M[4], Praveen M[5], Rithika K[6]
[1,2,3,4,5,6]*Department of CSE (IoT & Cybersecurity Including BCT), SNS College of Technology, Coimbatore, India*

***Abstract*—Unexpected machine failures cause equipment damage, downtime and added maintenance cost, while commercial vibration-monitoring systems are often too expensive and too complex for small workshops, teaching laboratories and student projects. This paper presents MORPH-X, a low-cost machine behaviour monitoring prototype built with an ESP32 microcontroller and an MPU6050 motion sensor, tested on a 12 V DC geared motor driven through an L298N module. During calibration, the system records 100 readings while the sensor is still and stores their mean vibration value as a baseline of normal behaviour. Each later reading is compared with this baseline; a deviation greater than a threshold of 0.35 is reported as an anomaly through an on-board LED, a serial message and a live dashboard. The dashboard is hosted by the ESP32 on its own Wi-Fi network and shows the vibration value, baseline, deviation, a live graph and the status on a mobile phone, with no internet or router needed. The paper sets out the necessity of the project, the design, algorithm, implementation and evaluation procedure, and states the limits of the approach plainly. The method is baseline calibration with threshold detection rather than a trained machine-learning model, and it provides a practical base for future predictive maintenance work.**

***Index Terms*—anomaly detection, condition monitoring, ESP32, Internet of Things, MPU6050, predictive maintenance, vibration monitoring**

I. INTRODUCTION

A. Background

Rotating machines such as motors, pumps and fans are the working core of workshops, factories and laboratories. During healthy operation each machine produces a characteristic vibration pattern. When a fault such as imbalance, looseness, misalignment or wear begins to develop, this pattern changes. Detecting the change early allows maintenance to be planned before failure, which reduces unplanned downtime and repair cost [1], [2].

Industrial condition-monitoring systems perform this task well, but their cost and complexity can put them out of reach of small workshops, teaching laboratories and student projects. At the same time, low-cost microcontrollers with built-in wireless connectivity and small MEMS motion sensors have made it practical to build affordable monitoring prototypes, and IoT-based monitoring of machine vibration and condition has been demonstrated in several settings [2], [3].

B. Contributions and Organisation

The contributions of this work are: (i) a clear statement of the necessity for an affordable, self-contained monitoring tool; (ii) a low-cost ESP32 and MPU6050 design with on-device processing; (iii) a baseline calibration and threshold-based anomaly detection method that is simple to explain and reproduce; (iv) a self-hosted wireless dashboard; (v) an evaluation procedure with defined metrics for a real motor; and (vi) a clear statement of scope and limits. Section II explains why the project is necessary, Section III reviews related work, Section IV describes the system, Section V explains the method, Section VI covers the implementation, Section VII presents the evaluation, Section VIII discusses strengths and limitations, and Section IX concludes with future work.

II. NECESSITY OF THE PROJECT

A. Cost of Unplanned Machine Failure

A machine that fails without warning stops the work that depends on it, may damage connected equipment,

and is repaired under time pressure. Waiting for a breakdown (run-to-failure) is the simplest maintenance policy but the most costly when failures are sudden. Fixed-interval servicing avoids some failures but replaces parts that are still healthy. Condition-based and predictive maintenance instead act on the observed state of the machine, and vibration is one of the most informative signals for this purpose because developing faults change how a machine moves [1], [4]. Continuous observation of machine behaviour is therefore necessary if small failures are to be caught before they become large ones.

B. Cost and Complexity Barrier

Commercial vibration-monitoring equipment typically relies on dedicated industrial accelerometers, data-acquisition hardware, analysis software and trained staff to interpret the results. For a small workshop, a teaching laboratory or a student team this combination is often unaffordable or unnecessary for the machines concerned. These users still have motors, fans and pumps whose sudden failure is disruptive, yet they commonly have no monitoring at all. An affordable alternative built from widely available parts is therefore necessary.

C. Gap in Existing Low-Cost Approaches

Research prototypes show that IoT and microcontroller hardware can support useful monitoring [3], [6]– [8]. However, many such systems depend on a gateway, a computer or a cloud service to collect and display data, and many diagnostic methods rely on trained models that need labelled fault data that small users do not have [2], [4]. What is missing is a self-contained node that works out of the box, needs no fault dataset, and whose decision rule a student or technician can understand and adjust.

D. Need for Infrastructure-Free Access

Workshops and laboratories often have no dependable internet connection or spare network infrastructure near the machine. A monitor that needs a router or cloud account cannot be used there. Hosting the dashboard on the microcontroller itself, over its own Wi-Fi network, removes this dependency and lets any phone with a browser act as the display.

E. Educational Need

Condition monitoring combines sensing, signal processing, decision logic and visualisation. A low-cost, transparent platform lets students see the whole chain on real hardware and experiment with each stage, which is difficult with closed commercial products. The baseline-and-threshold method is deliberately simple so that it can be explained, reproduced and later replaced by more advanced techniques.

F. Summary of Needs

TABLE I NEEDS, GAPS AND THE MORPH-X RESPONSE

Need	Gap in current practice	MORPH-X response
Early warning of faults	Failures noticed only after breakdown	Continuous comparison of vibration with a learned baseline
Low cost	Industrial systems are costly and complex	ESP32 and MPU6050, inexpensive and easy to obtain
No fault dataset	Learned models need labelled faulty data	Baseline recorded from normal running only
No infrastructure	Gateway, router or cloud required	Dashboard served by the ESP32 on its own Wi-Fi
Transparency	Closed or opaque methods	Single threshold rule that is easy to explain and tune
Learning platform	Few open, hands-on kits	Open design usable for teaching and extension

G. Problem Statement

Unexpected machine failures and abnormal operating conditions can result in equipment damage, downtime and increased maintenance costs.

Traditional machine monitoring solutions may be expensive or unsuitable for small-scale applications. There is therefore a need for an affordable and accessible system that can continuously monitor machine motion and identify deviations from normal behaviour.

H. Objectives

- *Low cost:* use an ESP32 and an MPU6050, which are inexpensive and easy to obtain.
- *Real-time monitoring:* receive sensor readings continuously.

- *Anomaly detection:* identify motion or vibration that differs from the learned baseline.
- *Visual feedback:* show status, measurements and a graph on a mobile phone.
- *Wireless access:* host the dashboard on the ESP32 without internet or a router.
- *Future predictive maintenance:* provide a base for earlier detection of machine faults.

III. RELATED WORK

A. Vibration-Based Fault Diagnosis

Vibration analysis is one of the most widely used techniques for assessing the health of rotating machinery, because faults change how a machine vibrates. Work on intelligent fault diagnosis from vibration includes microprocessor-based analysis of roller-bearing vibration in turbine generators [1], machine-learning classification of rotor and bearing defects using experimental vibration data [4], and IoT-based diagnosis of induction motors using a convolutional neural network [2]. For induction motors, the envelope of the stator current has also been investigated using Hilbert spectrum analysis [5]. Classical approaches use time-domain measures such as RMS and peak values, and frequency-domain analysis based on the Fast Fourier Transform.

B. IoT-Based Vibration Monitoring

Low-cost sensing nodes and wireless links have been used to monitor machine condition remotely. For example, an IoT system for transformer condition monitoring combined vibration sensors, audio collectors, Arduino modules and LoRa links, with readings viewable on desktops and smartphones [3]. Other IoT and microcontroller-based systems include an AI-based dimensional measurement and material-transport system [6], an HVAC duct-cleaning bot monitored through the Blynk IoT platform [7], and an experimentally studied Arduino-based safety device [8]. Such systems show that affordable hardware can support useful monitoring, and they typically rely on a gateway, a computer or cloud services for display.

C. Positioning of MORPH-X

MORPH-X takes the low-cost direction but keeps everything on the microcontroller: it calibrates a baseline, evaluates each reading and serves its own dashboard over a local Wi-Fi network, so no gateway, router or cloud account is needed. It does not attempt fault classification. Instead, it provides a transparent baseline-and-threshold detector that a student team can build, understand and extend.

IV. SYSTEM DESIGN

A. Architecture

The machine produces vibration and motion, which the MPU6050 measures as acceleration and rotation. The ESP32 computes the baseline and the deviation of each reading. The status is Normal when the deviation stays within the threshold and Anomaly when it exceeds it, and both are shown on a live mobile dashboard (Fig. 1).

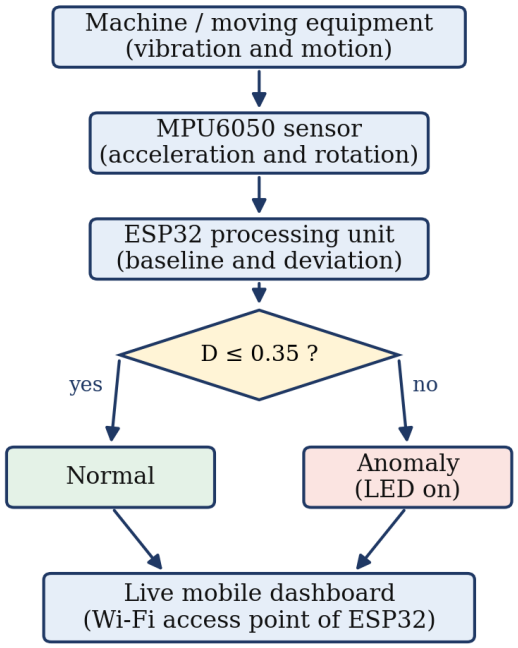


Fig. 1. Block diagram of the MORPH-X system.

B. Hardware Components

Table II lists the components used in the prototype. The ESP32 handles processing, anomaly logic, the Wi-Fi access point and the web server.

The MPU6050 is a six-axis motion sensor with a 3-axis accelerometer and a 3-axis gyroscope that communicates over the I²C bus [9]; the present firmware uses the acceleration readings only.

The motor under test is driven through an L298N motor driver module [11] from a separate 12 V adapter, so that the motor current never passes through the ESP32.

TABLE II COMPONENTS USED IN THE PROTOTYPE

Component	Specification	Role
ESP32 dev module	ESP-WROOM-32, Wi-Fi and Bluetooth [10]	Processing, anomaly logic, access point, web server
MPU6050	6-axis accelerometer and gyroscope, I²C	Measures vibration and motion
DC geared motor	12 V, 100–300 RPM	Machine under test
Motor driver	L298N module	Drives the motor from the 12 V supply
Power supply	12 V adapter	Powers the motor driver and motor
Breadboard, wires	Solderless breadboard	Connections between modules
Mobile phone	Any phone with a web browser	Dashboard client

C. Connections

The MPU6050 is connected to the ESP32 with four wires (Table III). The sensor is fixed rigidly on the body of the motor so that it feels the machine's vibration rather than the motion of loose wires.

TABLE III MPU6050 TO ESP32 CONNECTIONS

MPU6050 pin	ESP32 pin	Purpose
VCC	3V3	Power
GND	GND	Ground
SDA	GPIO 21	I²C data
SCL	GPIO 22	I²C clock

The L298N driver is connected as in Table IV. The motor supply and the ESP32 share a common ground.

TABLE IV L298N DRIVER CONNECTIONS

L298N pin	ESP32 / component	Description
ENA	GPIO 25 (PWM)	Motor speed control (enable)
IN1	GPIO 26	Motor direction 1
IN2	GPIO 27	Motor direction 2
IN3, IN4, ENB	Not used	Optional
OUT1	Motor +	DC motor terminal 1
OUT2	Motor −	DC motor terminal 2
+12V	12 V adapter +	Motor power supply
GND	12 V adapter − and ESP32 GND	Common ground

D. Power and Safety

The motor is powered through the motor driver from the separate 12 V adapter and is never connected directly to the ESP32, because the motor draws far more current than a microcontroller pin can supply. The driver and the ESP32 share a common ground so that the control signals have a common reference. The ESP32 and MPU6050 are powered from the USB connection. Wiring is changed only with the power off.

V. METHODOLOGY

A. Vibration Value

The MPU6050 provides acceleration values (a_x, a_y, a_z) along its three axes. Each sample is reduced to a single vibration value v, the magnitude of the resultant acceleration:

$$v = \sqrt{(a_x^2 + a_y^2 + a_z^2)} \quad (1)$$

where a_x, a_y and a_z are the accelerations along the X, Y and Z axes, expressed in g, and v is the vibration value used throughout this paper.

B. Baseline Calibration

At start-up, with the sensor kept still, the system records $N = 100$ vibration values v_i and stores their mean as the baseline B:

$$B = (1/N) \Sigma v_i , \; i = 1, \ldots, N \quad (2)$$

C. Deviation and Threshold Detection

After calibration, each new reading v is compared with the baseline to give the deviation

$$D = | v - B | \quad (3)$$

The status is Normal if $D \leq T$ and Anomaly if $D > T$, where $T = 0.35$ is a fixed threshold. On an anomaly, the on-board LED turns on and the serial monitor reports the event.

D. Algorithm

Fig. 2 shows the complete procedure: initialise, calibrate once, then loop over readings, classify each and update the dashboard.

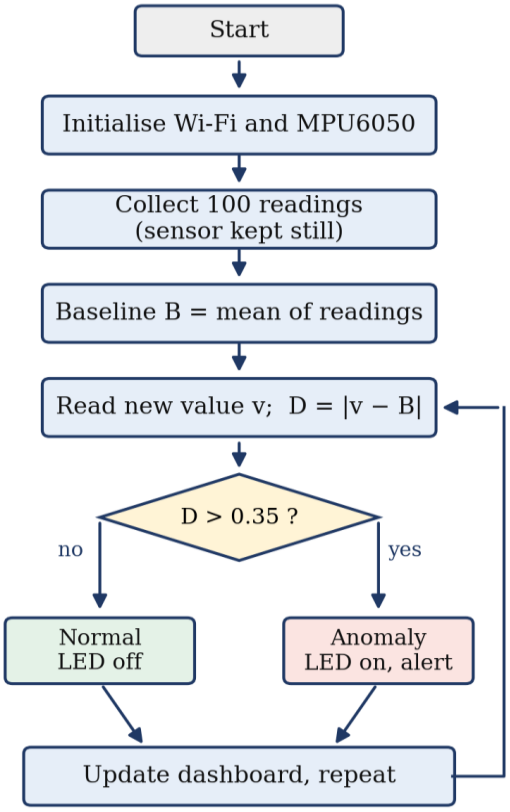


Fig. 2. Flowchart of baseline calibration and anomaly detection.

E. Meaning of the Threshold

Fig. 3 illustrates the idea. After calibration, readings that stay within a band of width *T* around the baseline are treated as normal, and readings outside the band are flagged. A threshold that is too small causes false alarms from ordinary noise, while one that is too large can miss small but real changes. *T* should therefore be checked against the spread of readings seen during healthy operation of the specific machine.

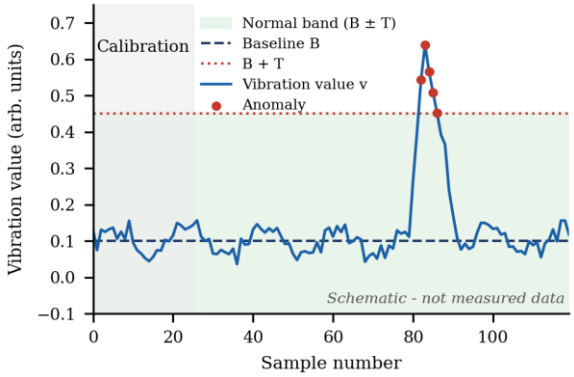


Fig. 3. Schematic of baseline, threshold band and anomaly (illustration only, not measured data).

F. Scope of "Learning"

In this work, learning means calibrating a baseline and detecting deviation from it. It is not a trained machine-learning model, and the system cannot name the cause of an anomaly.

The structure allows learned models to replace the fixed threshold in later versions.

VI. IMPLEMENTATION

A. Firmware

The firmware, written for the ESP32, performs five tasks: reading the MPU6050, running the calibration, evaluating every sample, controlling the LED and serial output, and serving the dashboard. Calibration runs once at start-up; the evaluation loop then runs continuously.

B. Wireless Access

The ESP32 creates its own Wi-Fi network (access point mode). A phone joins this network and opens the dashboard page in a browser, so the system works anywhere, including places without internet or a router. The range is limited to the ESP32's Wi-Fi coverage.

C. Dashboard

The dashboard shows the current vibration value, the baseline, the deviation, a live graph and the Normal or Anomaly status, so that a user can see at a glance whether the machine is behaving as it did during calibration (Fig. 4). The status indicator is green for Normal and red for Abnormal, and a log lists recent events.



Fig. 4(a). Dashboard, Normal state (motor running normally).



Fig. 4(b). Dashboard, Abnormal state (after adding load or loosening the mount).

D. Test Rig

For the evaluation, the MPU6050 was mounted on the body of the 12 V geared motor, the motor was driven through the L298N module from the 12 V adapter, and the ESP32 was connected to the sensor as in Table III. Fig. 5 shows the hardware layout of the prototype: the 12 V adapter powers the motor driver and motor, the MPU6050 and the ESP32 are on the breadboard, and the ESP32 is linked to the sensor and to the driver.

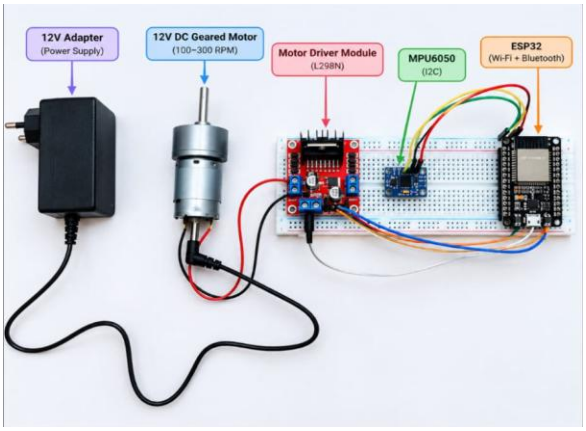


Fig. 5. Hardware layout of the MORPH-X prototype: 12 V adapter, 12 V DC geared motor (100–300 RPM), L298N motor driver, MPU6050 (I²C) and ESP32 on a breadboard.

VII. EVALUATION

A. Procedure

The system was first calibrated with the sensor still. The sensor was then mounted on the running motor, and the vibration value and status were recorded for five trials under each of two conditions: the motor running normally, and the motor with a deliberate change (added load or a slightly loosened mounting).

B. Metrics

For the test runs, the detection rate and false-alarm rate are defined as

$$DR = TP / (TP + FN) \times 100\% \quad (4)$$

$$FAR = FP / (FP + TN) \times 100\% \quad (5)$$

where TP is the number of abnormal samples correctly flagged, FN the number of abnormal samples missed, FP the number of normal samples wrongly flagged, and TN the number of normal samples correctly passed.

C. Results

[Provisional values: replace with the final recorded measurements before submission.]

Table V gives the vibration values and detected status for five trials with the motor running normally, and Table VI gives the same for the changed operating condition.

TABLE V NORMAL OPERATING CONDITION

Trial	Vibration value (g)	Detected status
1	0.06	Normal
2	0.05	Normal
3	0.07	Normal
4	0.06	Normal
5	0.05	Normal

Average vibration: 0.058 g. Detection accuracy: 5/5 (100%).

TABLE VI CHANGED OPERATING CONDITION

Trial	Vibration value (g)	Detected status
1	0.38	Abnormal
2	0.42	Abnormal
3	0.47	Abnormal
4	0.39	Abnormal
5	0.44	Abnormal

Average vibration: 0.42 g. Detection accuracy: 5/5 (100%).

Across the ten trials, TP = 5, FN = 0, FP = 0 and TN = 5, which gives DR = 100% and FAR = 0% from (4) and (5). Fig. 6 plots the measured vibration value against time with the 0.35 threshold line.

D. Discussion of Results

During the normal condition, the vibration values stayed below the 0.35 threshold and the system correctly reported the status as Normal.

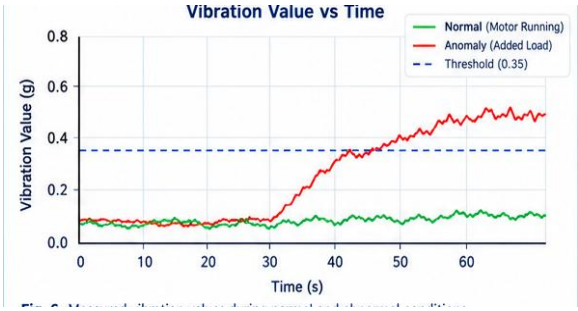


Fig. 6. Measured vibration values during normal and abnormal conditions.

When an additional load was applied, or the motor mount was slightly loosened, the vibration value rose beyond 0.35 and the system reported the status as Abnormal. The live dashboard updated the status in real time and the LED alert was triggered.

These results show that the simple baseline-and-threshold method can detect changes in machine behaviour on this motor. They come from one motor and a small number of trials, and they do not establish performance across machine types, speeds or fault kinds (Section VIII-B).

VIII. DISCUSSION AND LIMITATIONS

A. Strengths

The prototype shows that a baseline-and-threshold method on very inexpensive hardware can flag changes in a machine's motion in real time, and that the result can be presented on a phone without extra infrastructure. This directly answers the needs set out in Section II. The logic is simple enough to be understood, reproduced and modified by students.

B. Limitations

TABLE VII CURRENT CAPABILITIES AND LIMITATIONS

Current capability	Limitation
Reads acceleration from the MPU6050	The gyroscope readings are not used
Establishes a baseline	Baseline comes from calibration readings, not a full healthy operating cycle
Detects deviation with a fixed threshold	Cannot identify the cause of an anomaly; may need tuning per machine
Shows readings on a Wi-Fi dashboard	Range limited to the ESP32's Wi-Fi coverage
Tested on one motor	Results do not establish performance across machine types or speeds
Runs on low-cost hardware	Not a certified industrial monitoring device

C. Possible Applications

- Monitoring small motors, fans and pumps in workshops and laboratories.
- A teaching platform for sensors, IoT and condition monitoring.
- A starting point for a predictive maintenance system.

IX. CONCLUSION AND FUTURE WORK

MORPH-X is a low-cost prototype that learns a baseline of a machine's normal motion, detects deviations from it and displays the status live on a phone using an ESP32 and an MPU6050. It responds to a practical need for affordable, infrastructure-free and understandable machine monitoring, and it demonstrates sensor data acquisition, baseline-based anomaly detection and wireless visualisation on inexpensive hardware. It was evaluated on a 12 V geared motor. In the tests, the prototype monitored the motor's vibration and detected abnormal behaviour when the vibration value exceeded the defined threshold, and the dashboard and LED alert gave real-time status. This indicates that a low-cost IoT solution can support early anomaly detection in rotating machines. Its limits are stated openly.

Future work includes: using gyroscope data and multi-axis features such as RMS, peak and crest factor; adding frequency analysis (FFT) to capture the vibration frequencies linked with particular faults; recording datasets under normal and faulty conditions such as imbalance, looseness and misalignment; replacing the fixed threshold with adaptive or statistical thresholds and machine-learning classifiers; combining extra sensors into a health index; and adding data logging, cloud connectivity and remote alerts for predictive maintenance

DECLARATION

A generative AI tool was used solely for grammar checking of the manuscript text. The authors reviewed

all content and take full responsibility for the manuscript.

REFERENCES

[1] L. Mubaraali, N. Kuppuswamy, and R. Muthukumar, "Intelligent fault diagnosis in microprocessor systems for vibration analysis in roller bearings in whirlpool turbine generators real time processor applications," *Microprocess. Microsyst.*, vol. 76, Art. no. 103079, 2020.

[2] L. Mubaraali, N. Kuppuswamy, and R. Muthukumar, "Internet of Things-based induction motor diagnosis using convolutional neural network," *Electr. Power Compon. Syst.*, vol. 52, no. 2, pp. 265–276, 2024, doi: 10.1080/15325008.2023.2217434.

[3] T. N. H. Thinh, P. D. Lam, H. Q. Tran, L. H. C. Tien, and P. H. Thai, "Transformer vibration and noise monitoring system using internet of things," *IET Commun.*, vol. 17, no. 7, pp. 815–828, 2023, doi: 10.1049/cmu2.12585.

[4] K. M. Almutairi and J. K. Sinha, "Experimental vibration data in fault diagnosis: A machine learning approach to robust classification of rotor and bearing defects in rotating machines," Machines, vol. 11, no. 10, Art. no. 943, 2023.

[5] W. R. Babu, R. S. Kumar, and R. S. Kumar, "Rigorous investigation of stator current envelope of an induction motor using Hilbert spectrum analysis," Mater. Today, Proc., vol. 45, pp. 2474–2478, 2021, doi: 10.1016/j.matpr.2020.11.054.

[6] R. Senthil Kumar, C. Mohan Raj, W. Rajan Babu, K. Uday, V. Sudha, and V. Anandhakumar, "Smart dimensional measurement and material transportation (SDMMT) system using artificial intelligence," in Proc. Int. Conf. Intelligent Data Communication Technologies and Internet of Things (IDCIoT), 2023, doi: 10.1109/IDCIoT56793.2023.10053429.

[7] R. Senthil Kumar et al., "HVAC duct cleaning bot and monitoring using Blynk IoT," in [Book/Conference Title], CRC Press, doi: 10.1201/9781003770497-5.

[8] W. Rajan Babu et al., "Arduino based women safety device: An experimental study," Premier J. Sci., 2025, doi: 10.70389/PJS.100241.

[9] InvenSense, "MPU-6000 and MPU-6050 product specification," datasheet.

[10] Espressif Systems, "ESP32 series datasheet," datasheet.

[11] STMicroelectronics, "L298 dual full-bridge driver," datasheet.