Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

InfraSense - Portable Structural Health Monitoring for Concrete Structures

AkhildevA[1], AparnaR[2], Athulanilkumar[3], Gauthamajay[4],SijuKoshy[5],DhanyaV Kurup[6]
student[1],student[2],student[3],student[4],AssistantP rofessor[5],AssistantP rofessor[6]
Department of Computer Science and Engineering
College of Engineering Aranmula
(Under CAPE, Estd by Govt. of Kerala)
Aranmula, Kerala, India

***Abstract*—InfraSense is a compact, portable device designed to simplify structural health monitoring of concrete buildings such as homes, schools, and community facilities. It integrates calibrated sensors to measure strain, vibration, and environmental conditions, enabling users to perform accurate, zone-by-zone assessments of structural elements. Paired with a cloud-based web application, it securely stores and analyzes inspection data over time, helping identify early signs of damage such as cracks, spalling, and material fatigue. The system features intuitive dashboards, threshold-based alerts, and automated report generation, making it accessible even to non-specialists like homeowners and local engineers. With capabilities such as multi-user access, offline data syncing, and optional open-data sharing, InfraSense enhances collaboration and community awareness. By promoting early detection, data-driven decision-making, and preventive maintenance planning, it significantly reduces the risk of structural failures and contributes to safer, more resilient infrastructure.**

***Index Terms*—Concrete Structures, Portable Sensor Device, Strain Measurement, Vibration Analysis, Environmental Monitoring, Cloud-Based Analytics, Zone-wise Inspection, Preventive Maintenance, Damage Detection, Crack Monitoring, Data Visualization, Infrastructure Safety, Smart Inspection System.**

I. Introduction

InfraSense is a portable and affordable structural health monitoring (SHM) system designed to assess the condition of concrete structures such as homes, schools, and small community buildings. Concrete, although widely used in construction, is prone to deterioration over time due to environmental exposure, load variations, moisture infiltration, and aging. These factors can lead to structural issues such as cracks, spalling, corrosion, and material fatigue, which may eventually result in serious safety hazards if not detected early.Traditional SHM systems are typically expensive, complex, and permanently installed, making them unsuitable for community-level applications. InfraSense addresses this limitation by introducing a compact, sensor-integrated device capable of performing zone-by-zone structural analysis. It utilizes sensors to measure key parameters such as strain, vibration, temperature, and humidity, and transmits the collected data to a cloud-based platform for storage and analysis.The system is supported by a user-friendly web dashboard that visualizes real-time data, tracks historical trends, and generates alerts when abnormal conditions are detected. Designed for accessibility, InfraSense enables non-specialist users like homeowners and local engineers to monitor structural health and take preventive maintenance actions. By combining portability, IoT technology, and data analytics, InfraSense transforms structural monitoring into an efficient, scalable, and user-oriented solution, ultimately enhancing safety and infrastructure resilience.

II. Literature Review

C. Scuro et al [1] IoT for masonry SHM:This paper demonstrates use of IoT and cloud platforms for masonry structure monitoring. The paper also focuses on digital twin integration. However, the system has certain limitations, like it is tailored for masonry buildings; lacks general-purpose architecture

M. Datoussa¨ıd et al.[2] Cloud-Based IoT Platform for Bridge Health Monitoring:This paper developed a digital twin model for infrastructure monitoring using IoT sensors. The major downsides for this development was: Implementation complexity and heavy reliance on cloud processing.

J. Hou et al[3] . Smart Monitoring of Civil Infrastructure Using Wireless Sensor Networks:This paper proposed the idea of using wireless sensor networks for vibration monitoring of civil structures. the major drawbacks for this paper was: It lacked integration with modern IoT or edge–cloud systems; wireless sensors are more costlier than wired sensors

III. System Specification

A. Hardware Requirements

This section outlines the design and selection of hardware components:

- **ESP32 Development Board:**The ESP32 Development Board acts as the main controller of the system. It collects data from different sensors and processes the information. The board also sends the collected data to the server through Wi-Fi, which enables real-time monitoring and communication in the system.
- **Arduino Nano:** The Arduino Nano is used for handling additional sensor inputs in the system. It helps in collecting data from multiple sensors and sending the data to the main controller for further processing
- **ADS1115:**The ADS1115 is a 16-bit analog-to-digital converter used in the system. It converts analog signals received from the sensors into digital data so that the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

microcontroller can process it. It also provides high-resolution measurement for accurate monitoring.

- **7805 Voltage Regulator:**The 7805 Voltage Regulator is used to maintain a stable5V power supply for the electronic components in the circuit. It protects the system-from voltage fluctuations and ensures proper functioning of the hardware.
- **Rechargeable Li-ion Battery:**The rechargeable Li-ion battery provides a portable power supply for the system. It allows the device to operate without an external power source, which makes the system suitable for field usage.
- **Printed Circuit Board (PCB):**Printed Circuit Boards and jumper wires are used to connect the electronic components in the circuit. They help in building and organizing the prototype hardware setup of the system
- **Portable Enclosure (Casing):**The portable enclosure or casing is used to protect the electronic components from dust, damage, and environmental conditions. It also makes the device portable and safe for field deployment.

Sensors Required:

- **Y3 AH130 Strain Gauge -**Detects bending or strain in concrete structures.
- **ADXL345 Accelerometer -**Measures vibration and movement of structures.
- **Sliding Potentiometer -**Used to measure crack width in concrete surfaces.
- **Resistive Surface Moisture Sensor -**Detects moisture or water presence in concrete.
- **DS18B20 Temperature Sensor -**Measures temperature variations in the structure.

B. Software Environment

This section outlines the software tools and technologies used for system development:

- **Backend Processing:**Developed using Python with Flask framework, the backend handles data processing, API communication, and server-side operations. It manages data validation, storage, and interaction between the device and the cloud.
- **Frontend Interface:**The interactive dashboard is built using HTML, CSS, and JavaScript, enabling real-time visualization of sensor data, alerts, and system status through graphs and charts.
- **Development Environment:**Arduino IDE is used for programming the microcontrollers, allowing sensor integration, data acquisition, and communication setup.
- **Database and Cloud Services:**Firebase or MySQL is used for storing and managing sensor data securely, enabling real-time access, synchronization, and historical data tracking.
- **Libraries and Tools:**Various libraries such as sensor libraries (ADXL345, Dallas Temperature), communication libraries (Wi-Fi, HTTP), and data visualization tools are used to ensure smooth system functionality.

IV. Proposed Architecture

The proposed architecture of the InfraSense system is based on a multi-layered IoT framework designed to ensure efficient data acquisition, processing, storage, and user interaction. The architecture is divided into five main layers: the device layer, data layer, cloud layer, application layer, and user layer. The device layer consists of various sensors such as strain gauges, accelerometers, temperature, and moisture sensors connected to a microcontroller (ESP32), which collects real-time structural and environmental data. The data layer performs initial processing, including filtering, calibration, and formatting of sensor data before transmission. This processed data is then sent via Wi-Fi to the cloud layer, where it is securely stored and further analyzed using backend technologies like Python and Flask. The application layer provides an interactive web-based dashboard developed using HTML, CSS, and JavaScript, allowing users to visualize real-time readings, analyze historical trends, and receive automated alerts when values exceed predefined thresholds. The user layer includes stakeholders such as homeowners, engineers, and administrators who interact with the system to monitor structural conditions and make informed maintenance decisions. Additionally, the architecture supports offline data storage and later synchronization, ensuring reliability even in low-connectivity environments.

V. Implementation & Output

A. Code Implementation

The following snippets demonstrates the model's Backend Server Code (Flask API).

VI. Implementation

```python
from flask import Flask, jsonify, request, Response
from flask_cors import CORS
import datetime
import csv
import io
from reportlab.lib.pagesizes import A4
from reportlab.pdfgen import canvas

app = Flask(__name__)
CORS(app)

DEVICE_ID = "INFRA-001"
ZONE = "Column_A1"
HISTORY = []
LATEST_DATA = None
LAST_UPDATE_TIME = None
OFFLINE_THRESHOLD = 15
ACTIVE_ALERTS = {}
ALERT_HISTORY = []
```

Listing 1. Initialization and Configuration

```python
def get_utc_now():
    return datetime.datetime.utcnow().replace(
        tzinfo=datetime.timezone.utc
    )

def evaluate_overall(data):
    status = {}

    if data["strain"] < 250:
        status["strain"] = "SAFE"
    elif data["strain"] < 350:
        status["strain"] = "WARNING"
    else:
        status["strain"] = "CRITICAL"

    if data["vibration"] < 0.05:
        status["vibration"] = "SAFE"
    elif data["vibration"] < 0.08:
        status["vibration"] = "WARNING"
    else:
        status["vibration"] = "CRITICAL"

    if data["temperature"] < 35:
        status["temperature"] = "SAFE"
    elif data["temperature"] < 45:
```

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

```
        status["temperature"] = "WARNING"
    else:
        status["temperature"] = "CRITICAL"

    if data["humidity"] < 70:
        status["humidity"] = "SAFE"
    elif data["humidity"] < 85:
        status["humidity"] = "WARNING"
    else:
        status["humidity"] = "CRITICAL"

    if data["crack"] < 0.3:
        status["crack"] = "SAFE"
    elif data["crack"] < 0.5:
        status["crack"] = "WARNING"
    else:
        status["crack"] = "CRITICAL"

    overall = "SAFE"
    if "CRITICAL" in status.values():
        overall = "CRITICAL"
    elif "WARNING" in status.values():
        overall = "WARNING"

    return overall, status
```

Listing 2. Time Handling and Safety Evaluation Logic

```
@app.route("/api/device", methods=["GET", "POST"])
def receive_device_data():
    global LATEST_DATA, LAST_UPDATE_TIME

    strain = float(request.args.get("value1"))
    temperature = float(request.args.get("value2"))
    crack = float(request.args.get("value3"))
    vibration = float(request.args.get("value4"))
    humidity = float(request.args.get("value5"))

    now = get_utc_now()
    now_iso = now.isoformat()

    LATEST_DATA = {
        "device_id": DEVICE_ID,
        "zone": ZONE,
        "timestamp": now_iso,
        "strain": strain,
        "vibration": vibration,
        "temperature": temperature,
        "humidity": humidity,
        "crack": crack
    }

    LAST_UPDATE_TIME = now

    HISTORY.append({
        "time": now_iso,
        "strain": strain,
        "vibration": vibration,
        "temperature": temperature,
        "humidity": humidity,
        "crack": crack
    })

    return jsonify({"status": "data received"})
```

Listing 3. API Endpoint for Receiving IoT Data

```
@app.route("/api/live", methods=["GET"])
def live_data():
    return jsonify(LATEST_DATA)

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

Listing 4. Live Data API and Server Execution

A. *System Outputs*

The system offers a centralized dashboard that enables real-time monitoring and control for administrators.

The output design of the InfraSense system focuses on presenting the processed sensor data in a clear, meaningful, and user-friendly manner. The system displays the monitored structural parameters through a web-based dashboard that provides real-time readings of strain, vibration, crack width, moisture, and temperature. Visual representations such as graphs, charts, and status indicators help users easily understand the condition of the concrete structure. Historical data comparison is also available to analyze trends and detect gradual structural changes over time.

The system automatically generates alerts whenever any monitored parameter exceeds predefined safety thresholds.

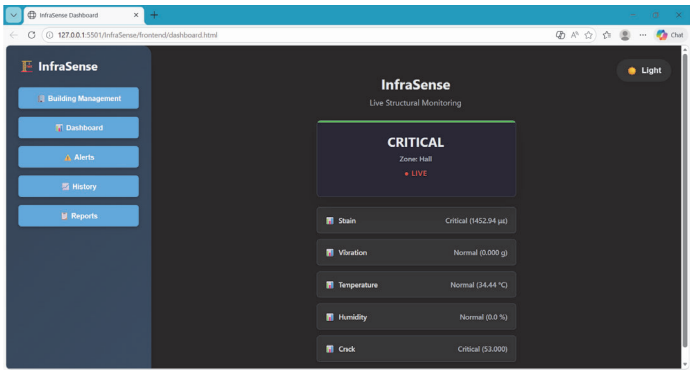

Fig. 1. Interactive Dashboard of InfraSense.

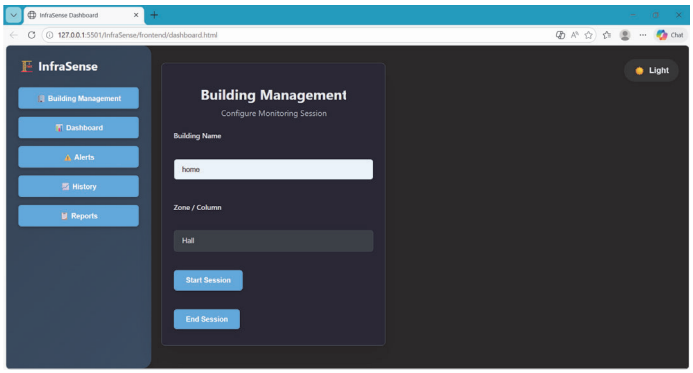

Fig. 2. Building Management Section

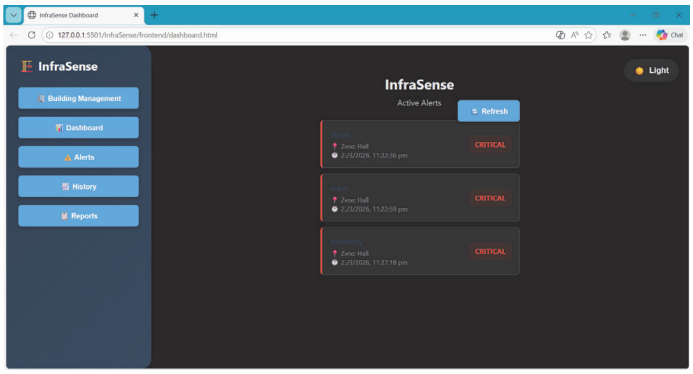

Fig. 3. Alerts Section.

Fig. 4. History Section.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

These alerts notify users about possible structural abnormalities such as excessive vibration, crack expansion, or high moisture levels, allowing timely preventive maintenance actions. The dashboard also supports report generation and data visualization, enabling engineers, maintenance staff, and administrators to monitor structural health efficiently and make informed decisions regarding building safety.

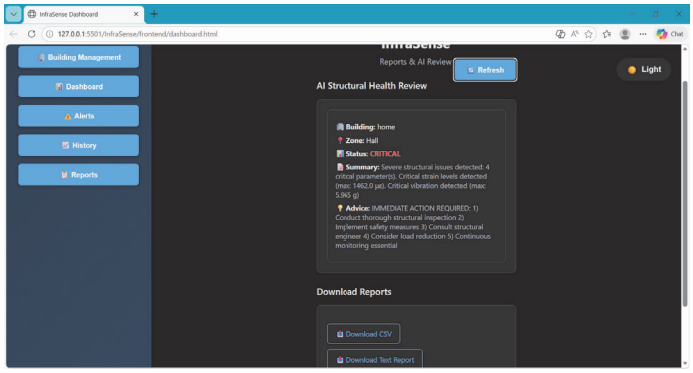

Fig. 5. Report Section.

VII. Conclusion

The InfraSense system provides a practical and efficient approach to monitoring the structural health of concrete buildings, offering a modern and accessible solution for community-level infrastructure safety. By integrating IoT-based sensors and cloud analytics, the system enables users to measure strain, vibration, temperature, and humidity in different concrete zones and store the readings in real time. This approach ensures that early signs of structural deterioration, such as cracks or corrosion, can be identified before they lead to major damage. The system's cloud-connected dashboard provides visual analytics, historical comparisons, and alerts, helping users take preventive maintenance actions quickly and effectively. InfraSense's portability and affordability make it an ideal solution for residential buildings, schools, and small-scale public structures that cannot afford traditional SHM systems. The project promotes awareness about structural safety and demonstrates how modern IoT and data analytics can be used to enhance building resilience. By empowering communities to participate in monitoring their own infrastructure, InfraSense contributes to safer living environments and sustainable urban development. The system represents a step toward democratizing structural health monitoring and making it more inclusive, scalable, and environmentally responsible.

VIII. Acknowledgment

The authors would like to express their sincere gratitude to Mr.Siju Koshy, Head of the Department of Computer Science and Engineering, and our project guide, Ms. Dhanya V Kurup, for their invaluable guidance, constant encouragement, and for providing the necessary facilities at the College of Engineering Aranmula to complete this research.

References

[1] L. Gigli et al., "Next Generation Edge-Cloud Continuum Architecture for Structural Health Monitoring," IEEE Transactions on Industrial Informatics, vol. 20, no. 4, pp. 5874-5887, April 2024
[2] M. Norouzi and N. Masoumi, "Crasen: A phase variation passive sensor node for metallic structural health monitoring," IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1-11, 2023
[3] Y. Zhang, V. Adin, S. Bader, and B. Oelmann, "Leveraging Acoustic Emission and Machine Learning for Concrete Materials Damage Classification on Embedded Devices," IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1-8, 2023
[4] X. Hong, D. Yang, L. Huang, B. Zhang, and G. Jin, "Vibration adaption deep convolutional transfer learning method for stranded wire structural health monitoring using guided wave," IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1-10, 2023
[5] T. H. Dinh et al., "Toward vision-based concrete crack detection: Automatic simulation of real-world cracks," IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1-15, 2023
[6] B. Ando et al., "Toward an autonomous sensor node exploiting a nonlinear energy harvester," IEEE Transactions on Instrumentation and Measurement, vol. 73, pp. 1-10, 2024