Team name: Voltaura

Team number: 16

Team members:

Ananya M Murthy	ananyammurthy232@gmail.com	7975976508
Deeksha Ramappa Gani	deekshagani77@gmail.com	9019703864
Shriya S	shriyasatish012@gmail.com	9632073794
Varsha TM	varshatm09@gmail.com	9945803290

Github link : https://github.com/Shriya-012/VoltAura-Team-16-

Title of the project: AI-Based Smart Monitoring System with Self-Healing through Fault Isolation and Load Control

Problem statement:

Electrical systems and machines often fail due to abnormal conditions such as overheating, excessive vibration, or electrical fluctuations. Traditional protection systems react only after faults occur and lack real-time intelligence, prediction, and autonomous response.

Objectives:

- To develop a smart system for real-time monitoring of system parameters.
- To detect faults using Digital Twin and AI-based techniques.
- To analyze system behaviour and identify abnormal conditions.
- To improve system reliability and safety through intelligent decision-making.

Existing work:

- Traditional Circuit Breakers and Relays → Automatically disconnect power when a fault occurs, but only react after damage and have no intelligence or prediction capability.
- IoT-Based Monitoring Systems → Monitor system parameters remotely using the internet, but depend on cloud connectivity and lack real-time local decision-making.
- AI-Based Fault Detection (Cloud-Based)→ Use machine learning for fault prediction, but require continuous internet and have latency issues.

- Digital Twin Systems (Industrial Level)→ Create virtual models for monitoring and analysis, but are complex, expensive, and not suitable for small-scale or real-time embedded systems.

Gap analysis:

•Existing systems react only after faults occur and lack real-time intelligent decision-making.

•Most AI-based systems depend on cloud, causing delay and reliability issues.

•Digital twin implementations are complex and not integrated with embedded systems.

•Lack of combined approach integrating monitoring, AI, and automatic system response.

•No efficient low-cost solution for real-time fault detection and control.

Novelty:

•Integration of Digital Twin, Edge AI, and intelligent control in a single system.

•Use of the K-Nearest Neighbours (KNN) algorithm for fault classification, achieving high accuracy in detecting normal and abnormal conditions.

•Real-time fault detection using multi-sensor data (temperature + vibration) and audio-based AI model.

•Implementation of Edge AI, where all processing and decision-making happen on the device without the internet.

•Development of a Node-RED dashboard as a digital twin, providing real-time visualisation of system parameters.

•Incorporation of priority-based load control, enabling intelligent system response during faults.

Project Description:

This project presents an intelligent monitoring system that integrates sensor-based data acquisition, machine learning, and digital twin-based analysis for effective fault detection. The system continuously collects real-time data such as temperature and vibration using sensors like DHT11 and MPU6050. This data represents the actual operating condition of the system. A digital twin model is defined using normal operating values, which serves as a reference to compare real-time data and identify deviations. Machine learning is implemented using the K-Nearest Neighbours (KNN) algorithm. The model is trained using labelled datasets in Google Colab, achieving high classification accuracy. Additionally, Edge Impulse is used for vibration and audio-based classification, enhancing the system's ability to detect abnormal conditions. The decision-making process is carried out on the embedded system by combining threshold-based digital twin logic and machine learning predictions.

Based on the system state (normal or fault), appropriate actions such as alerts and load control are triggered using relay modules. A web-based dashboard is developed using the WARP AI tool to visualize real-time system parameters such as temperature, voltage, and system status. This dashboard provides an intuitive interface for monitoring, while the digital twin logic operates within the system for comparison and analysis. Overall, the system provides a real-time, edge-based solution for intelligent monitoring and fault detection, improving system reliability and enabling efficient control without dependence on cloud infrastructure.

List of components and the software used:

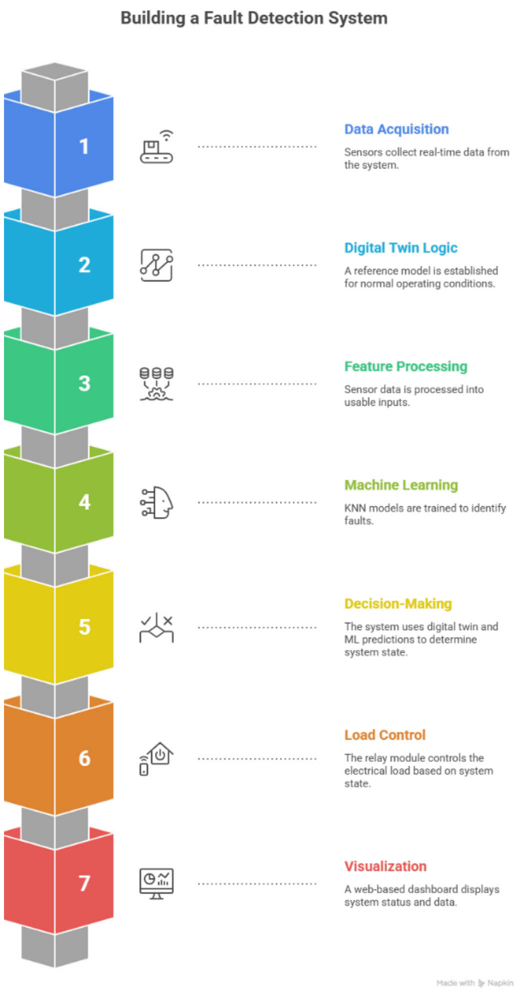
Hardware Components

Sl. No	Component	Specification
1	Glyph C6 Development Board	ESP32-C6 (WiFi enabled microcontroller)
2	DHT11 Sensor	Temperature & Humidity Sensor
3	MPU6050 Sensor	Accelerometer & Gyroscope (Vibration)
4	2-Channel Relay Module	5V Relay Module
5	OLED Display	0.91 inch I2C Display
6	Buzzer	5V Active Buzzer
7	LEDs	Red & Green LEDs
8	Push Buttons	Tactile Switch
9	Potentiometer	100kΩ
10	Capacitive Touch Sensor	CAP1203 Module
11	Breadboard	Standard
12	Jumper Wires	Male-Male & Male-Female
13	Power Supply	5V 2A DC Adapter

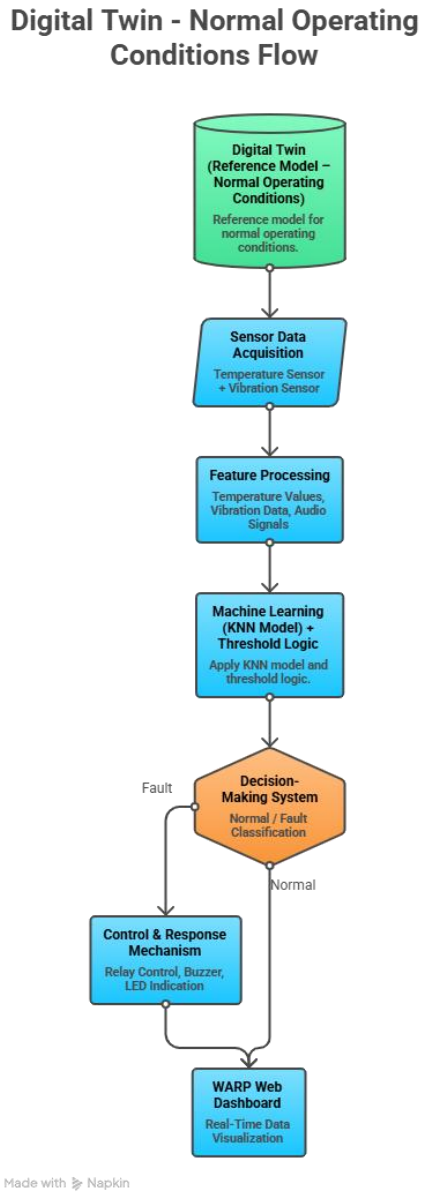
Software & Tools Used

Sl. No	Software / Tool	Purpose
1	Arduino IDE	Programming and uploading code to microcontroller
2	Google Colab	Training KNN machine learning model
3	EDGE IMPULSE	Data collection and ML model testing (vibration/audio)
4	WARP AI Tool	Web dashboard for real-time visualization
5*	Node-RED*	Digital twin concept visualization
6	Serial Monitor (Arduino)	Debugging and real-time data monitoring
7	Arduino IDE	Data corting with light Pyleon Inc

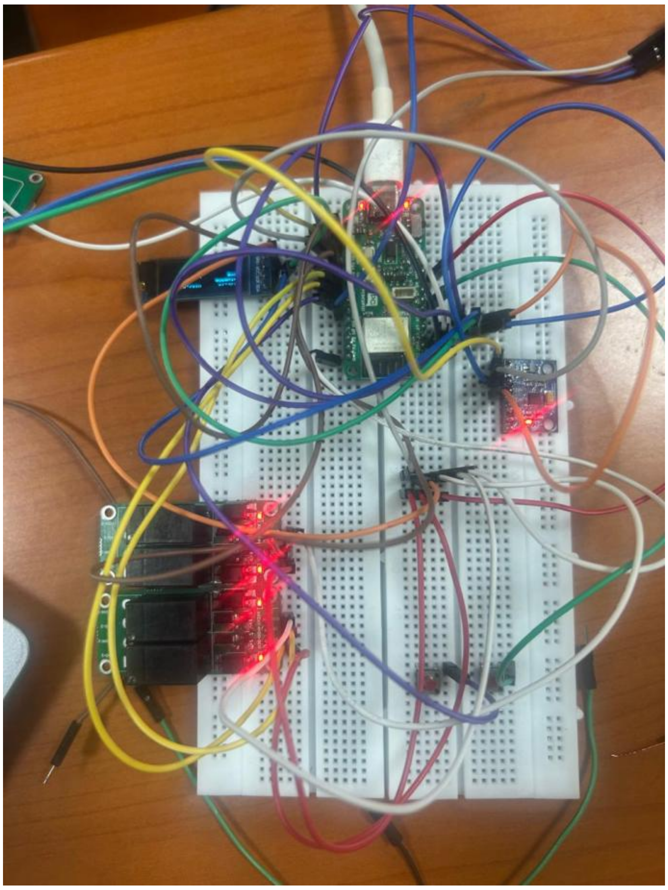
Methodology:



Work flow:


Digital Twin - Normal Operating Conditions Flow
Digital Twin (Reference Model – Normal Operating Conditions)
Reference model for normal operating conditions.
Sensor Data Acquisition
Temperature Sensor + Vibration Sensor
Feature Processing
Temperature Values, Vibration Data, Audio Signals
Machine Learning (KNN Model) + Threshold Logic
Apply KNN model and threshold logic.
Decision-Making System
Normal / Fault Classification
Fault
Normal
Control & Response Mechanism
Relay Control, Buzzer, LED Indication
WARP Web Dashboard
Real-Time Data Visualization
Made with Napkin

Pics of the prototype:



Video link :
https://drive.google.com/file/d/1NhE08ebV1vL7MSxwdK8YewmullEZ2v1P/view?usp=sharing