Low Level Design (LLD)

Smart Grid Management: AI-Powered Predictive Maintenance and Demand Forecasting

Content

1. Introduction

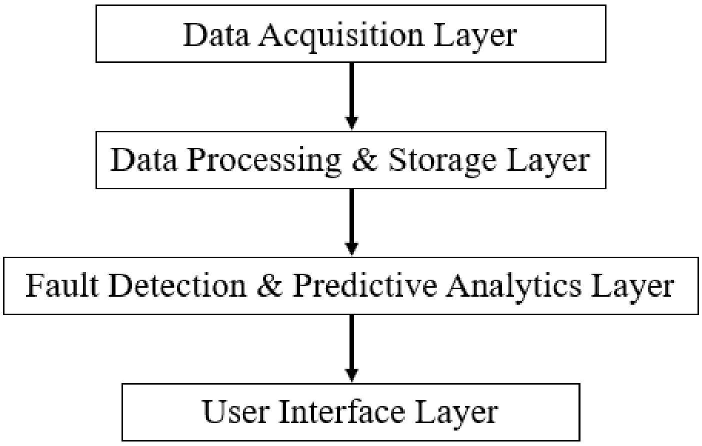
1.1. Purpose

The purpose of this Low-Level Design (LLD) document is to provide a detailed blueprint of the Smart Grid Management: AI-Powered Predictive Maintenance and Demand Forecasting system. It elaborates on the implementation details of components, modules, and data flow, ensuring seamless development and deployment.

1.2. Scope

The system integrates IoT-based real-time monitoring, AI-driven predictive maintenance, and demand forecasting. It processes data from temperature sensors (MLX90614), voltage sensors, and current sensors, leveraging machine learning models for fault detection and early warning.

2. Architecture



2.1. Architecture Components

The system consists of the following key components:

1. Data Acquisition System (DAQ)
    - Hardware: ESP32, MLX90614, Voltage Sensor, Current Sensor, GPS Module
    - Data Collection: Reads real-time sensor data
    - Communication: WiFi-enabled data transmission
2. Cloud Infrastructure
    - Storage: Firebase / AWS for logging sensor data
    - Processing: Google Apps Script for real-time analytics
3. AI-Based Predictive Maintenance
    - Fault Classification: Uses Random Forest, Gradient Boosting, and SVM
    - Dynamic Thresholding: Defines normal, warning, and critical levels
    - Federated Learning: Distributed AI training across grid nodes
4. User Dashboard
    - Visualization: Streamlit-based dashboard
    - Alerts: SMS/Email notifications for fault detection

3. Architecture Description

The Smart Grid Management: AI-Powered Predictive Maintenance and Demand Forecasting system is designed as an IoT-driven, AI-powered framework that integrates real-time data collection, AI-based fault detection, and cloud-based predictive maintenance for enhanced grid reliability.

3.1. Architectural Overview

The architecture follows a modular design with distinct components for data acquisition, processing, analytics, and visualization. The system leverages edge computing for real-time monitoring and cloud integration for large-scale predictive analysis.

3.2. System Architecture Layers

The architecture consists of the following primary layers:

3.2.1. Data Acquisition Layer

- Components:
    - ESP32 Microcontroller: Central controller for IoT-based sensor readings.
    - MLX90614 Infrared Temperature Sensor: Monitors overheating in transformers.
    - Voltage Sensor: Tracks fluctuations in voltage levels.
    - Current Sensor: Detects anomalies in current flow.
    - GPS Module (NEO-6M): Enables geolocation tracking of grid components.
- Functionality:
    - Continuously collects temperature, voltage, and current readings.
    - Transmits data using WiFi (HTTP API) to cloud servers.
    - Provides edge-based early warnings using LED indicators & buzzers.

3.2.2. Data Processing & Storage Layer

- Cloud Integration:
    - Google Firebase / AWS for real-time data logging.
    - Google Apps Script for initial processing.
- AI-based Data Processing:
    - Uses Ensemble Learning (Random Forest, Gradient Boosting, SVM).
    - Implements Dynamic Thresholding to categorize grid parameters into:
        - Normal (Safe Operating Range)

    - Warning (Anomaly Detected)
    - Critical (Immediate Action Required)
  - Federated Learning (FedAvg, PySyft) for decentralized training of predictive models across multiple substations.

3.2.3. Fault Detection & Predictive Analytics Layer

- Machine Learning Model:
  - Input: Sensor logs from the cloud.
  - Process: AI-based fault classification.
  - Output: Predicted fault status (Normal/Warning/Critical).
  - Technology: Python (Scikit-learn, TensorFlow), PySyft for Federated Learning.
- Health Score Computation:
  - Weighted parameters: Temperature (40%), Voltage (30%), Fault Frequency (30%).
  - Provides a risk assessment index for each transformer or substation.

3.2.4. Early Warning & Visualization Layer

- Local Alerting Mechanism:
  - LED/Buzzer System: Alerts based on AI-predicted faults.
  - Mobile & Web Notifications: Sends real-time alerts via email/SMS.
- User Interface (Dashboard):
  - Streamlit Web Dashboard: Displays grid health, fault trends, and predictive maintenance insights.
  - Real-time Visualization: Graphs, Heatmaps, and Fault Reports.

4. Unit Test Cases

Test Case ID	Test Description	Expected Outcome
TC01	Sensor data acquisition	Successfully logs real-time sensor data from MLX90614, voltage, and current sensors
TC02	Fault classification model	Correctly classifies faults into Normal (0), Warning (0.5), and Critical (1) based on AI model output
TC03	Early Warning System (EWS)	Triggers notifications (LED, buzzer, cloud alerts) when threshold levels are exceeded
TC04	Data transmission to the cloud	Sensor readings are successfully sent to the cloud storage (Firebase/AWS)
TC05	Federated Learning Model	Updates model parameters without sharing raw data
TC06	Health Score Calculation	Generates accurate health scores based on weighted sensor inputs
TC07	Dynamic Thresholding Algorithm	Adjusts fault thresholds dynamically based on historical data trends
TC08	Real-time Dashboard Visualization	Displays live grid parameters and fault conditions accurately
TC09	User-defined Input Predictions	Allows users to input custom sensor values and receive AI predictions
TC10	Power Outage Simulation	System accurately detects anomalies leading to simulated grid failure
TC11	GPS Location Tagging	Correctly assigns geographic coordinates to each grid component
TC12	WiFi Connectivity & Network Failure Handling	System reconnects and resumes data logging after network failure
TC13	Power Failure Recovery	System restarts and retains previous sensor readings and alerts
TC14	Load Demand Forecasting	Predicts power consumption trends accurately based on past data
TC15	Sensor Calibration	Automatically adjusts sensor readings to eliminate noise and drifts