Project Design Phase
Solution Architecture

Date	19 February 2026
Team ID	LTVIP2026TMIDS61980
Project Name	Electric Motor Temperature Prediction System
Maximum Marks	4 Marks

Solution Architecture:

Overview

The solution architecture defines how the electric motor temperature prediction system is designed to transform raw sensor data into accurate temperature predictions and deliver them through a web interface.

The architecture bridges the gap between the industrial problem (motor overheating) and the technological solution (Machine Learning + Flask deployment).

The system consists of:

- Data Layer
- Processing Layer
- Machine Learning Layer
- Deployment Layer
- User Interface Layer

High-Level Architecture Description

- The proposed solution works in the following stages:
- Historical motor sensor data is collected.
- Data preprocessing is performed.
- Machine learning models are trained.
- Best model is selected (Decision Tree Regressor).
- Model is saved using pickle.
- Flask application loads the model.
- User inputs motor parameters.
- Scaler transforms input.
- Model predicts rotor temperature.
- Result is displayed on UI.

Example - Solution Architecture Diagram:

Solution Architecture Diagram:

Electric Motor Temperature Prediction using Machine Learning

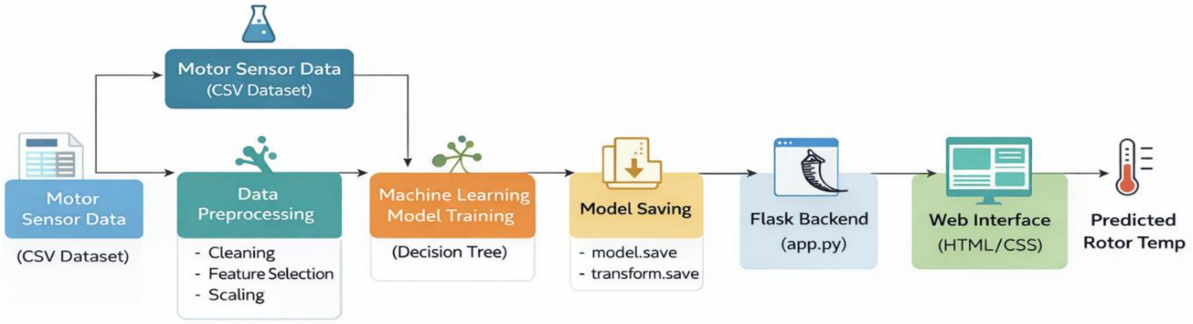

Figure 1: Architecture and data flow of the Electric Motor Temperature Prediction System

Data Layer

The dataset contains motor operational parameters including voltage, current, motor speed, ambient temperature, and stator temperatures

Preprocessing Layer

The preprocessing stage includes:

- Removing unnecessary columns
- Handling missing values
- Feature scaling using MinMaxScaler
- Splitting data into training and testing sets

Machine Learning Layer

Multiple regression models are trained:

- Linear Regression
- Decision Tree
- Random Forest
- SVR

The Decision Tree Regressor is selected based on highest R^2 score (~96%).

Presentation Layer

The frontend is built using HTML and CSS, providing:

- Manual prediction interface
- Sensor-based prediction module (future scope)
- Result display section

Data Flow Explanation

1. User enters motor parameters.
2. Flask receives input.
3. Input is converted into numeric format.
4. MinMaxScaler normalizes the data.
5. Decision Tree model predicts rotor temperature.
6. Prediction is displayed on the web page.

Security & Reliability

- Model file stored locally
- Input validation implemented
- Error handling mechanism included
- Scalable architecture for IoT integration