Technology Stack:

Technology Stack (Architecture & Stack)

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

Technical Architecture:

The Electric Motor Temperature Prediction System is developed using Machine Learning techniques to predict rotor temperature based on motor sensor data. The system analyzes operational parameters such as voltage, current, motor speed, and stator temperatures to forecast overheating conditions before failure occurs.

The architecture integrates data preprocessing, model training, prediction logic, and web deployment to provide an intelligent predictive maintenance solution.

Key Functionalities include:

- Sensor data preprocessing and feature scaling
- Machine Learning–based temperature prediction
- Regression model comparison and evaluation
- Automated prediction using trained model
- Web-based interface for user input and prediction results
- Visualization of prediction outputs
- Support for preventive maintenance decision-making

Table-1: Components & Technologies

S.No	Component	Description	Technology
1	User Interface	Web interface for entering motor parameters and viewing predictions	HTML, CSS, Flask Templates
2	Application Logic-1	Data preprocessing and feature scaling	Python, Pandas, NumPy, Scikit-learn
3	Application Logic-2	Machine learning model training and prediction	Decision Tree Regressor (Scikit-learn)
4	Application Logic-3	Model evaluation and performance measurement	RMSE, R^2 Score (Scikit-learn Metrics)
5	Database	Dataset storage and processing	CSV Dataset / Pandas DataFrame
6	Model Storage	Save trained ML model	Pickle (.pkl file)
7	Deployment Server	Backend application hosting prediction service	Flask (Python Web Framework)
8	Visualization	Data analysis and graphs	Matplotlib, Seaborn

Table-2: Application Characteristics

S.No	Characteristics	Description / Technology
1	Open-Source Frameworks	Python ecosystem, Scikit-learn, Flask
2	Security Implementations	Input validation and controlled backend processing
3	Scalable Architecture	Modular ML pipeline allowing future IoT integration
4	Availability	Local or cloud deployment capability
5	Performance	Fast prediction using trained Decision Tree model

Milestones Implemented:

- Collected and analyzed electric motor sensor dataset
- Performed data cleaning and preprocessing
- Applied feature scaling using MinMaxScaler
- Split dataset into training and testing sets
- Implemented multiple regression models
- Compared models using RMSE and R^2 metrics
- Selected Decision Tree Regressor as final model
- Saved trained model using Pickle
- Developed Flask backend for prediction service
- Designed web interface for user interaction
- Integrated model prediction with frontend inputs
- Tested system performance and validated prediction accuracy

References:

- https://scikit-learn.org/stable/
- https://pandas.pydata.org/docs/
- https://flask.palletsprojects.com/
- https://numpy.org/doc/
- https://matplotlib.org/stable/