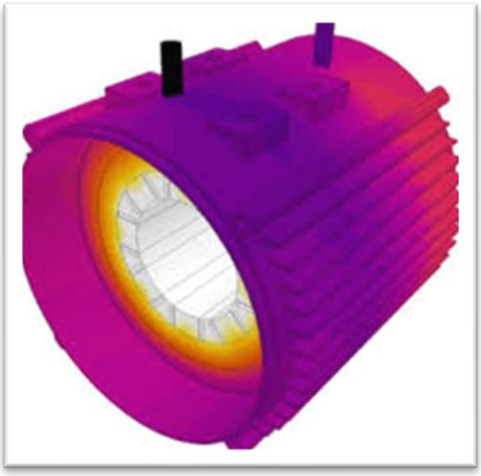
Electric Motor Temperature Prediction using Machine Learning



Team ID : LTVIP2025TMID49141

Team Size : 4

Team Leader : Nadimidoddi Venkata Murali Krishna

Team member : Sri Venkata Naga Sai Mukesh Chintapalli

Team member : Nagisetti Suresh Babu

Team member : Kommisetty Gopi

Abstract

The Electric Motor Temperature Prediction using Machine Learning project aims to forecast the temperature of electric motors before they overheat. Using historical sensor data such as voltage, current, motor speed, and ambient conditions, a machine learning model is trained to predict the permanent magnet (PM) temperature. This predictive approach helps in early detection of overheating, preventing motor damage and ensuring operational efficiency. The system leverages Python, Flask, and Scikit-learn to deliver accurate and real-time predictions through a web interface.

Table of Contents

1. Introduction

Electric motors play a crucial role in industrial and automotive systems. Continuous operation leads to heating, which, if not monitored, can cause motor failure. Traditional maintenance methods are reactive and inefficient. This project introduces a predictive maintenance approach using machine learning to forecast motor temperature based on operating parameters.

2. Objectives

- To predict electric motor temperature before it reaches unsafe limits.
- To analyze real-time sensor data for preventive maintenance.

- To enhance operational efficiency and reduce motor downtime.
- To visualize temperature predictions through a user-friendly web interface.

3. Problem Statement & Proposed Solution

Problem Statement

Frequent overheating of electric motors results in reduced lifespan, unscheduled maintenance, and potential system failure. Manual monitoring is time-consuming and lacks accuracy.

Proposed Solution

A machine learning-based predictive system is proposed to estimate the PM temperature using multiple sensor inputs such as current, voltage, motor speed, and ambient temperature. The trained model predicts temperature values in real-time via a Flask-based web application.

4. System Scenarios & Use Cases

Scenario 1: Preventive Maintenance

The system helps detect when a motor might overheat in advance. This allows maintenance teams to fix problems early and prevent motor failure.

Scenario 2: Energy Efficiency

By knowing the motor temperature, industries can keep motors running efficiently, save energy, and reduce electricity costs.

Scenario 3: Equipment Reliability

Predicting temperature ensures motors work within safe limits, improving their lifespan and avoiding unexpected breakdowns.

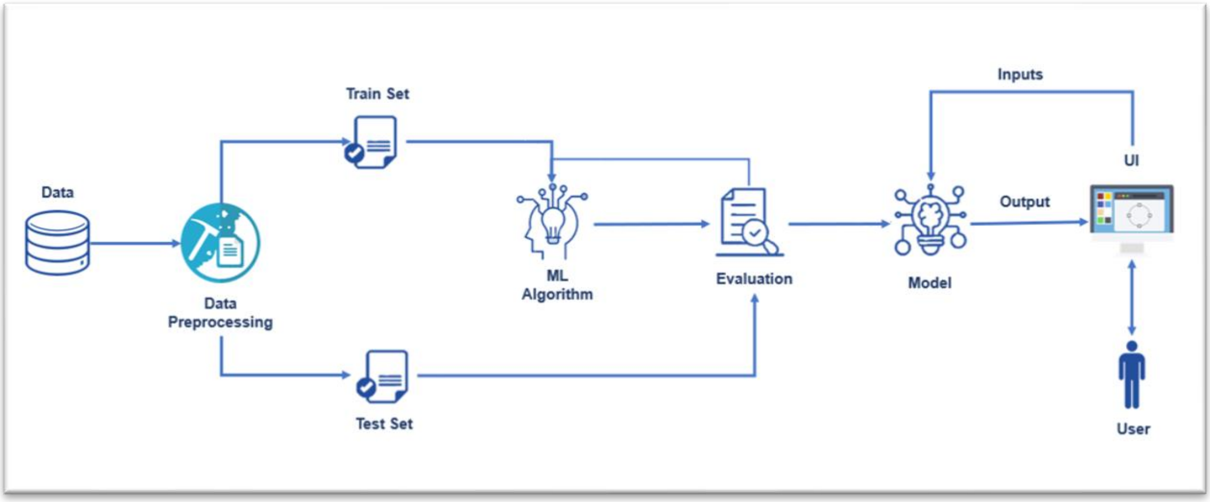
5. Technical Architecture

Components:

- **Frontend:** HTML, CSS, Bootstrap
- **Backend:** Python (Flask Framework)
- **Machine Learning:** Scikit-learn (Decision Tree Regressor)
- **Dataset:** Measures_v2.csv
- **Model Storage:** Joblib
- **Visualization Tools:** Matplotlib, Seaborn

Architecture Flow:

1. User inputs motor parameters through the web form.
2. Data is scaled using the MinMaxScaler.
3. The pre-trained Decision Tree model predicts PM temperature.
4. Results are displayed on the web page.



6. Methodology

1. **Data Collection:**
   Dataset containing features like voltage, current, torque, and ambient temperature.
2. **Data Preprocessing:**
   - Dropping unnecessary columns.
   - Handling missing values.
   - Scaling using MinMaxScaler.
3. **Model Training:**
   - Models tested: Linear Regression, Decision Tree, Random Forest, SVR.
   - Best performing model: **Decision Tree Regressor** ($R^2 \approx 0.92$).
4. **Model Evaluation:**
   - Metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), and R^2 Score.
5. **Model Deployment:**
   - Flask used to serve model predictions via a web interface.

7. Implementation

Programming Language: Python
Framework: Flask
Libraries Used:

- Numpy, Pandas – Data processing
- Matplotlib, Seaborn – Visualization
- Scikit-learn – Model building
- Joblib – Model saving/loading

Steps:

1. Train and test the model.
2. Save the model (model.save) and scaler (transform.save).
3. Develop Flask app with routes for input and prediction.
4. Deploy locally or on cloud platforms.

8. Results

The trained Decision Tree Regressor achieved:

- **R^2 Score:** 0.92
- **MSE:** 0.0018
- **MAE:** 0.025

These results indicate that the model can accurately predict PM temperature based on input sensor values.

9. Conclusion

The project successfully predicts electric motor temperature using machine learning. By integrating data analytics with predictive algorithms, it enhances safety, reduces downtime, and supports efficient maintenance. The web interface makes the system accessible and user-friendly.

10. Future Scope

- Integration of real-time IoT sensor data.
- Deployment on cloud platforms like AWS or Azure.
- Mobile application support for remote monitoring.
- Model optimization using deep learning (LSTM, CNN).

11. Appendix

Terminology

Parameter	Description	Unit
Ambient Temperature	Temperature of the environment	°C
Coolant Temperature	Temperature of cooling fluid	°C
Motor Speed	Rotational speed of motor shaft	RPM
Voltage (u_d, u_q)	Motor supply voltage components	V
Current (i_d, i_q)	Motor current components	A
PM Temperature	Predicted Permanent Magnet temperature	°C

12. References

- Flask documentation – https://flask.palletsprojects.com/
- Scikit-learn – https://scikit-learn.org/
- Python official documentation – https://docs.python.org/3/

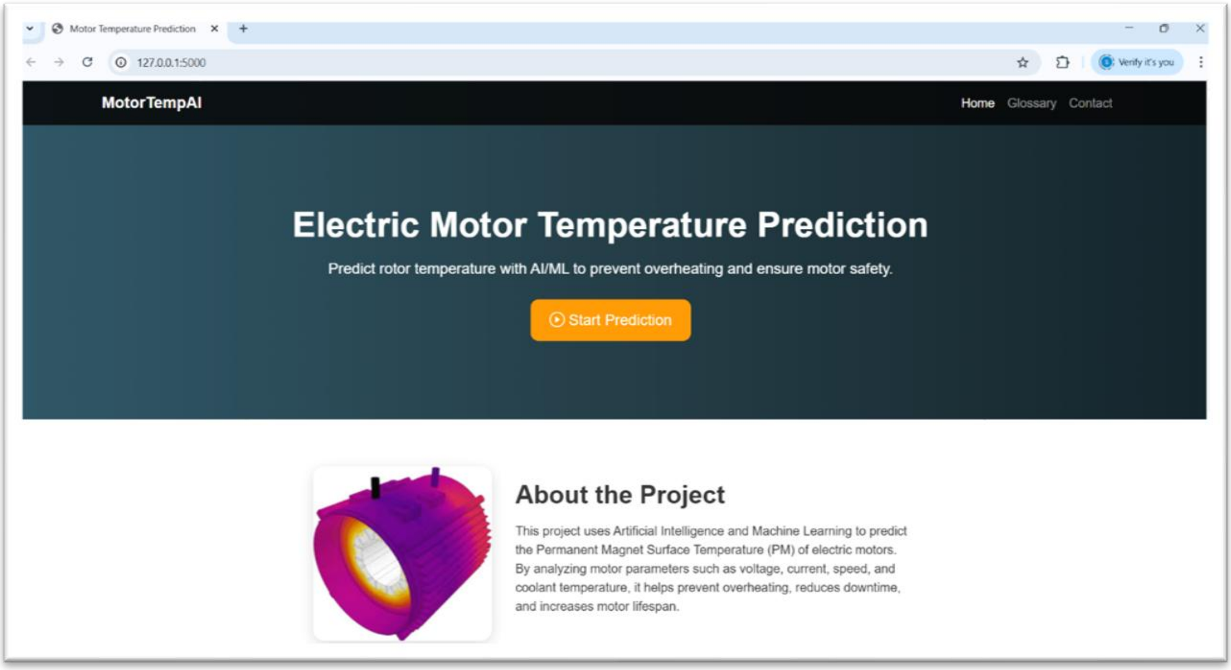
8.1 Demonstration

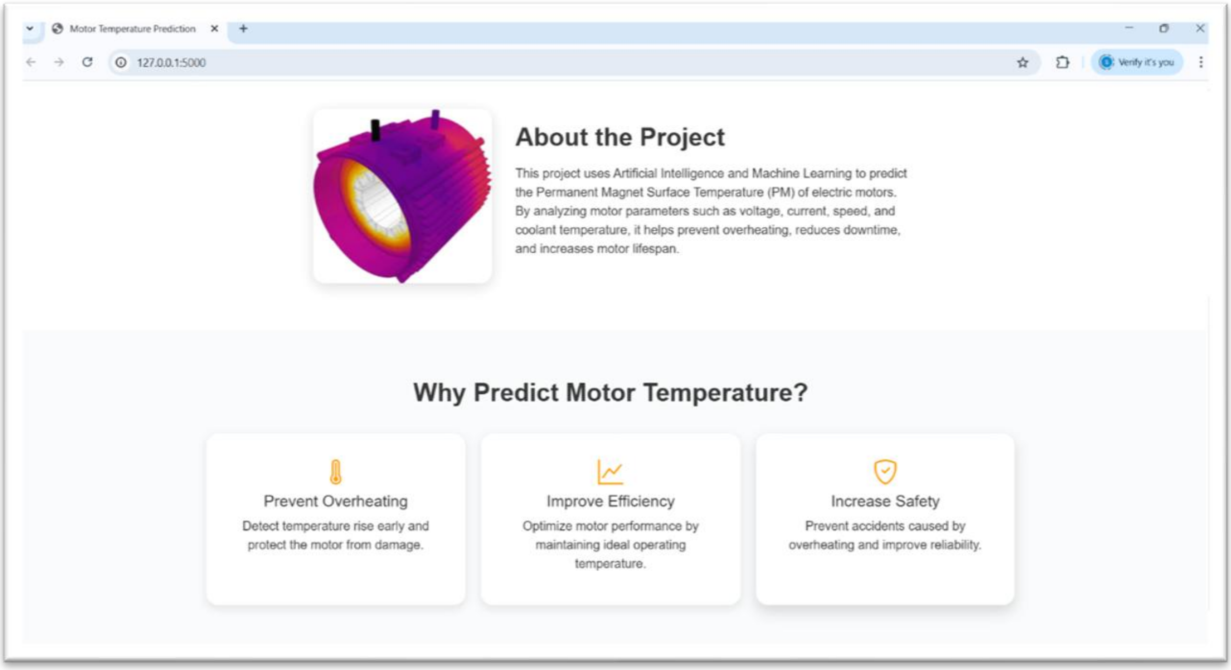
This section provides visual evidence of the developed project, including:

Source Code & Web Application Interface:

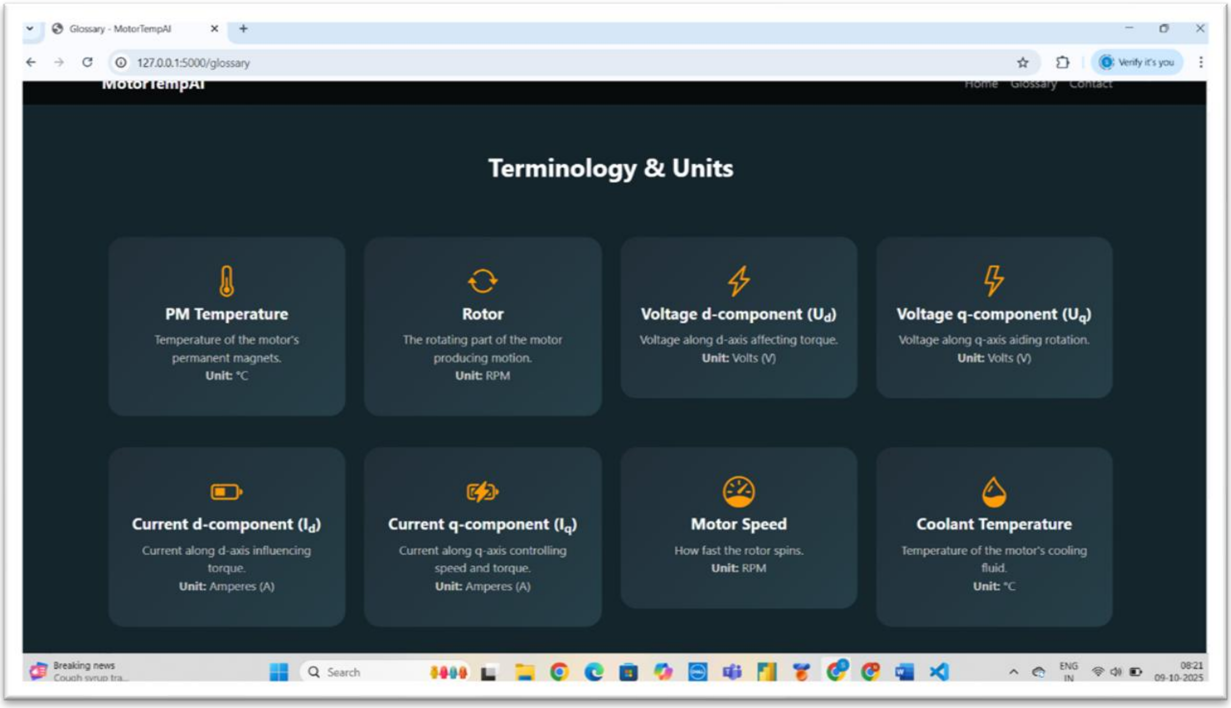
- **Home Page:** Overview of the project.
- **Manual Prediction Page:** Allows users to input sensor values manually.
- **Glossary Page:** Explains motor-related terminology.
- **Prediction Output:** Displays PM temperature in °C.

1.Home page

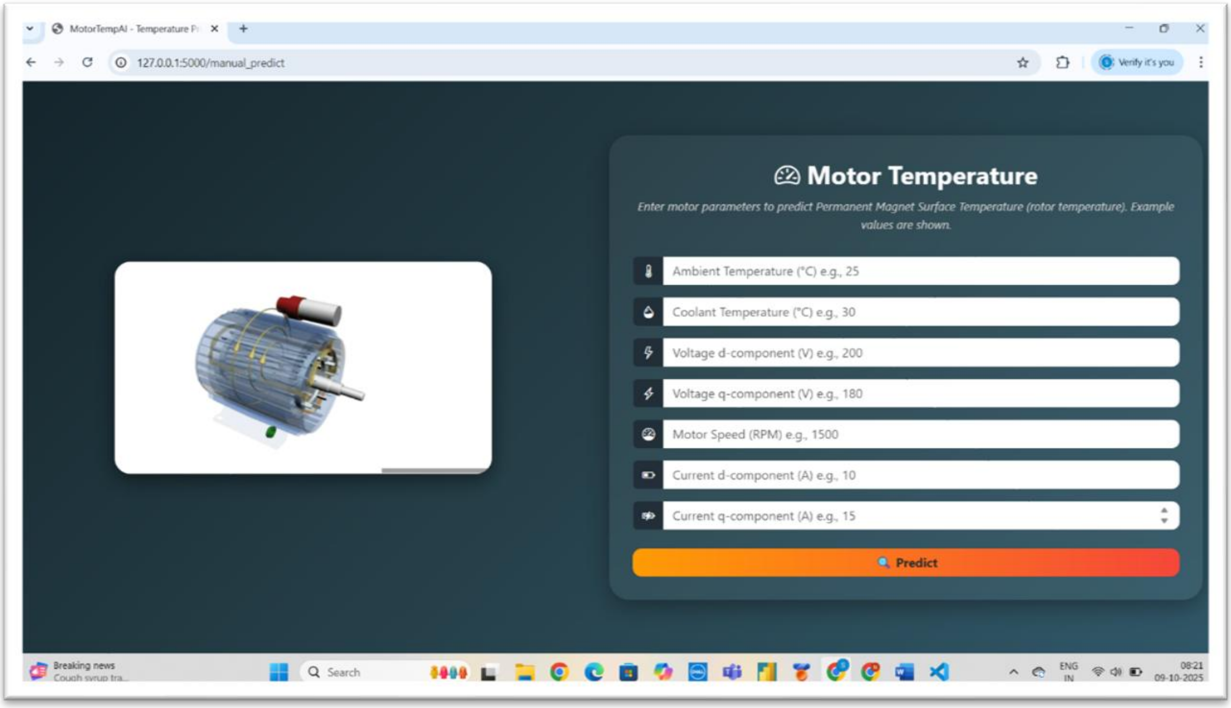

Motor Temperature Prediction
127.0.0.1:5000
Verify it's you
MotorTempAI
Home Glossary Contact
Electric Motor Temperature Prediction
Predict rotor temperature with AI/ML to prevent overheating and ensure motor safety.
Start Prediction
About the Project
This project uses Artificial Intelligence and Machine Learning to predict the Permanent Magnet Surface Temperature (PM) of electric motors. By analyzing motor parameters such as voltage, current, speed, and coolant temperature, it helps prevent overheating, reduces downtime, and increases motor lifespan.


Motor Temperature Prediction
127.0.0.1:5000
Verify it's you
About the Project
This project uses Artificial Intelligence and Machine Learning to predict the Permanent Magnet Surface Temperature (PM) of electric motors. By analyzing motor parameters such as voltage, current, speed, and coolant temperature, it helps prevent overheating, reduces downtime, and increases motor lifespan.
Why Predict Motor Temperature?
Prevent Overheating
Detect temperature rise early and protect the motor from damage.
Improve Efficiency
Optimize motor performance by maintaining ideal operating temperature.
Increase Safety
Prevent accidents caused by overheating and improve reliability.

2.Glossary Page



3.Manual Prediction Page



4.Prediction Output Page

GitHub Repository

🔗 GitHub Repository – https://github.com/muralikrishna4526/Electric-Motor-Temperature-Prediction-Using-Machine-Learning.git

Demo Video

🎥 [Demo Video – https://drive.google.com/file/d/1Wio6vez0JJEsjJNsqaY8WEKwcijJY35T/view?usp=sharing

References

- Flask documentation – https://flask.palletsprojects.com/
- Scikit-learn – https://scikit-learn.org/
- Python official documentation – https://docs.python.org/3/
- GitHub Repository – https://github.com/muralikrishna4526/Electric-Motor-Temperature-Prediction-Using-Machine-Learning.git
- Demo Video – https://drive.google.com/file/d/1Wio6vez0JJEsjJNsqaY8WEKwcijJY35T/view?usp=sharing