Electric Motor Temperature Prediction

Project Report

Submitted by: Harshada Gahininath Gujar
PRN: 202402100001 | Roll No: 77 | Batch: A4
MIT Academy of Engineering, Alandi

Acknowledgement

I would like to express my sincere gratitude to **SmartBridge** for providing me with the opportunity to work on this internship project titled *“Electric Motor Temperature Prediction.”*

I am thankful to my mentors and guides at SmartBridge for their valuable guidance, continuous support, and encouragement throughout the project. Their insights have been instrumental in enhancing my technical and analytical skills.

I also extend my heartfelt thanks to my faculty members at **MIT Academy of Engineering, Alandi**, for their constant motivation and academic support.

Finally, I am deeply grateful to my family and friends for their encouragement, patience, and unwavering support during this project.

Abstract

The Electric Motor Temperature Prediction project focuses on designing a machine learning model to estimate the rotor temperature of an electric motor. Different algorithms such as Linear Regression, Decision Tree, Random Forest, and Support Vector Machine (SVM) are implemented and evaluated. The best-performing model is selected and deployed using Flask to create a simple web-based interface. This project helps in predictive maintenance, reducing motor failures, and increasing operational efficiency.

1. Introduction

1.1 Background

Electric motors form the backbone of modern industry and household appliances. From manufacturing plants and transportation systems to fans, refrigerators, and washing machines, electric motors are indispensable. However, one of the major challenges in maintaining motor efficiency and longevity is **temperature management**.

Excessive heating can cause serious issues such as insulation degradation, reduced efficiency, and even catastrophic motor failures. Traditionally, motors are equipped with temperature sensors that provide real-time monitoring. However, these methods are **reactive** in nature and fail to provide foresight into potential overheating conditions.

Machine learning offers a **predictive approach**. By analyzing motor operating parameters such as current, voltage, ambient temperature, and speed, we can develop models that predict rotor temperature before dangerous thresholds are reached. This enables **preventive maintenance**, reduces downtime, and enhances both performance and safety.

1.2 Problem Statement

Conventional sensor-based monitoring systems provide only the current motor temperature but do not predict future trends. This can lead to **unexpected failures** when sudden load changes or harsh environmental conditions cause rapid heating. Moreover, adding additional physical sensors increases hardware costs and complexity.

Therefore, there is a need for a **data-driven predictive model** that uses sensor data (voltage, current, ambient conditions, speed, etc.) to estimate the **rotor temperature** accurately. Such a model should not only perform well in controlled datasets but also generalize to unseen operating conditions.

1.3 Objectives

The main objectives of this project are:

- To preprocess and analyze electric motor sensor data for better understanding.
- To apply **feature engineering techniques** to select and transform relevant input variables.
- To train and compare multiple **machine learning regression models** such as Linear Regression, Decision Tree, Random Forest, and Support Vector Machine.

- To evaluate models using performance metrics like **Mean Squared Error (MSE)** and **R^2 score**.
- To select the best-performing algorithm, save it in **.pkl format**, and integrate it into a **Flask web application**.
- To create a user-friendly interface that allows users to enter motor operating parameters and instantly receive predicted rotor temperature.

---

2. Literature Review

Recent research has shown that predictive maintenance using machine learning is a powerful tool for industrial systems.

- Studies demonstrate that **Random Forest and Gradient Boosting models** perform exceptionally well for regression problems in fault diagnosis.
- Researchers have also applied **Support Vector Machines (SVM)** for predictive modeling of motor faults, reporting good generalization on noisy data.
- Some works integrate **deep learning approaches** like LSTMs for sequential sensor data, but these models require significantly more computation and training time.
- In industrial applications, balancing **accuracy, interpretability, and deployment feasibility** is critical. Hence, ensemble models like Random Forest are often preferred.

This project builds on these insights by implementing and comparing traditional regression techniques with ensemble methods to determine the best fit for electric motor temperature prediction.

---

3. Methodology

3.1 Dataset Description

The dataset includes measurements of various operating conditions of an electric motor:

- **ambient_temp**: surrounding temperature (°C)
- **coolant_temp**: cooling fluid temperature (°C)
- **voltage_d, voltage_q**: d-axis and q-axis voltages (V)
- **current_d, current_q**: d-axis and q-axis currents (A)
- **motor_speed**: rotational speed (rpm)
- **pm_temp (target variable)**: rotor permanent magnet temperature (°C)

The target is continuous, making this a **regression problem**.

3.2 Feature Engineering

Feature engineering plays a crucial role in improving prediction performance:

- Handling missing values (if any)
- Normalizing/standardizing sensor readings for consistent scaling
- Removing highly correlated or redundant features
- Creating polynomial/interaction terms if necessary

In this project, primary raw features such as **ambient temperature, coolant temperature, voltages, currents, and speed** were used, as they directly influence rotor heating.

3.3 Algorithms Used

1. **Linear Regression** – A simple baseline model to check linear relationships between input features and rotor temperature.
2. **Decision Tree Regressor** – A non-linear model capable of handling complex relationships without feature scaling.
3. **Random Forest Regressor** – An ensemble of decision trees that reduces variance and improves accuracy.
4. **Support Vector Machine (SVM)** – Uses kernel tricks to model non-linear relationships with high accuracy.

3.4 Model Training and Testing

- Data split into **80% training** and **20% testing** subsets.
- Each model trained on training data and evaluated on test data.
- Evaluation metrics:
    - **Mean Squared Error (MSE)** – measures average squared error between predicted and actual temperature.
    - **R^2 Score** – indicates how well the model explains the variance in the target variable.

3.5 Model Selection

After comparing performance, **Random Forest Regressor** provided the best accuracy with the lowest MSE and highest R^2.

The trained model was serialized into **`model.pkl`** using Python's `pickle` library for deployment in Flask.

---

4. System Design

4.1 Architecture

The system consists of two layers:

1. **Machine Learning Layer** – Trains and saves the predictive model.
2. **Web Application Layer (Flask)** – Provides a simple form for user input and displays predictions in real time.

4.2 Workflow

1. Data Collection
2. Preprocessing and Feature Engineering
3. Model Training
4. Model Evaluation
5. Saving Best Model (`model.pkl`)
6. Flask Integration
7. User Input (via Web Form)
8. Predicted Rotor Temperature Display

---

5. Implementation

- **train_model.py** – Handles dataset loading, preprocessing, model training, evaluation, and saving the best model.
- **app.py** – Flask backend that loads the trained model and serves predictions.
- **templates/index.html** – Frontend form for user input and displaying results.

Libraries used:

- **pandas, numpy** – Data handling
- **scikit-learn** – Model training & evaluation
- **pickle** – Model serialization
- **Flask** – Web deployment

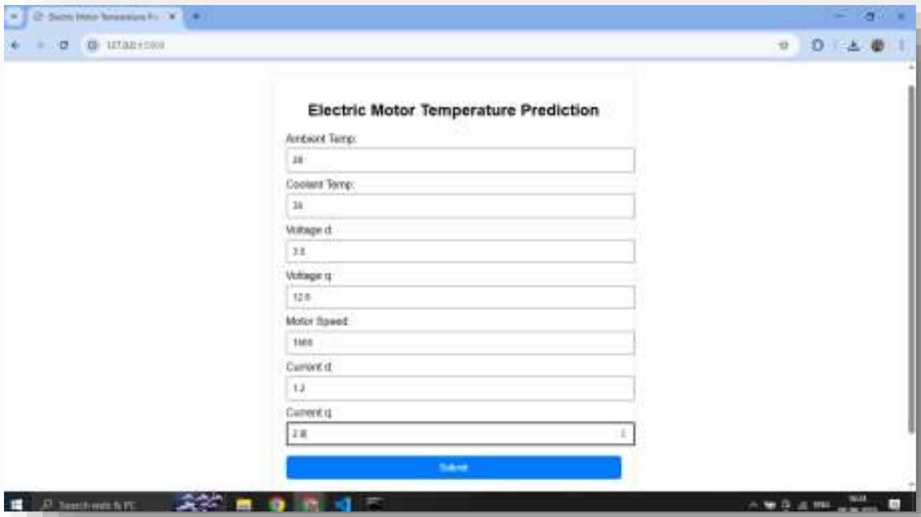
---

6. Results and Discussion

- Linear Regression provided a simple baseline but underfit the data.
- Decision Tree captured non-linear patterns but tended to overfit.
- Random Forest gave the **highest accuracy** and balanced performance.
- SVM performed well but required longer training time.

The Flask web app allowed users to input values (ambient temperature, coolant temperature, motor speed, etc.) and receive an instant rotor temperature prediction.

ScreenShots:



Electric Motor Temperature Prediction

Ambient Temp: 28

Coolant Temp: 36

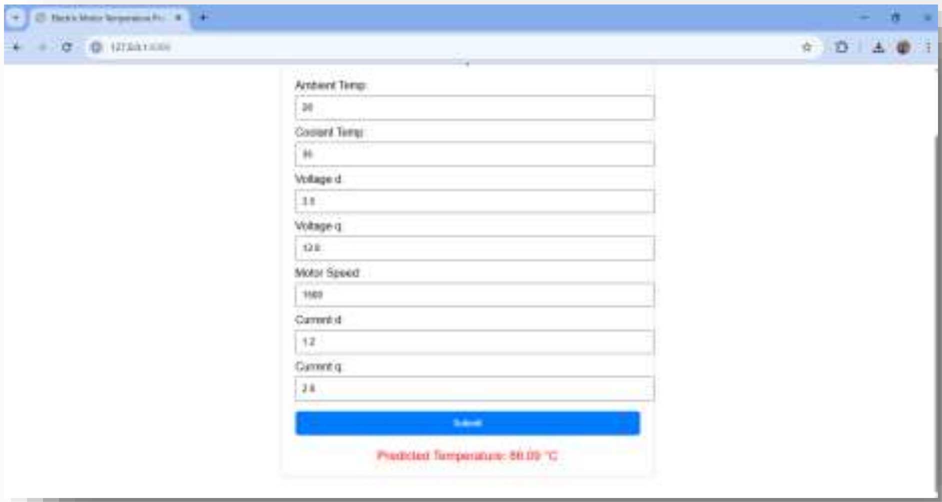
Voltage d: 3.8

Voltage q: 12.6

Motor Speed: 1500

Current d: 1.2

Current q: 2.8



Ambient Temp: 28

Coolant Temp: 36

Voltage d: 3.8

Voltage q: 12.6

Motor Speed: 1500

Current d: 1.2

Current q: 2.8

Predicted Temperature: 58.09 °C

7. Conclusion and Future Scope

This project demonstrates how machine learning can replace traditional sensor-only monitoring with predictive intelligence. The developed system provides:

- **Accurate rotor temperature prediction**
- **Preventive maintenance** capabilities
- **Low-cost deployment** using Flask

Future improvements:

- Use of **deep learning models** (e.g., LSTMs) for time-series sensor data.
- Integration with **IoT devices** for real-time monitoring in industries.
- Expanding dataset size and diversity for improved generalization.

8. References

- Pedregosa et al., *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research, 2011
- Research papers on predictive maintenance and electric motor monitoring
- Flask Documentation