Electric Motor Temperature Prediction

Project Report

1. Introduction

Electric motors are widely used in industrial, automotive, and household applications. Monitoring their temperature is critical for ensuring performance, safety, and longevity. This project involves building a machine learning model to predict the temperature of an electric motor's components based on input parameters. The model is trained using historical sensor data and deployed via a web interface using Flask and HTML.

2. Objective

The main objective of this project is to develop and deploy a machine learning model capable of predicting the **Permanent Magnet (PM)** temperature of an electric motor using real-time operational parameters. The application is designed to provide a simple web interface for users to input motor data and receive temperature predictions instantly.

3. Dataset Description

Source: Kaggle
Dataset URL:https://www.kaggle.com/datasets/wkirgsn/electric-motor-temperature
File Used:`PMSM_temperature_data.csv`

Features Selected:

- **ambient**: Ambient air temperature
- **coolant**: Coolant temperature
- **u_d**: d-axis voltage
- **u_q**: q-axis voltage
- **motor_speed**: Motor rotational speed
- **i_d**: d-axis current
- **i_q**: q-axis current

Target Variable:

- **pm**: Permanent Magnet temperature

The dataset contains continuous measurements of these parameters and temperature values collected from a Permanent Magnet Synchronous Motor (PMSM).

4. Tools and Technologies Used

- **Python** – Programming Language
- **Pandas & NumPy** – Data manipulation and analysis
- **Scikit-learn** – Machine Learning model training and evaluation
- **Google Colab** – Environment for model development and training
- **Flask** – Web framework for backend
- **HTML** – User interface design
- **joblib** – Saving and loading models
- **Kaggle API** – Downloading datasets

5. Methodology

Step 1: Data Collection and Preparation

- The dataset was downloaded from Kaggle using the Kaggle API.
- Missing values were checked, and only relevant features were selected for training.
- The dataset was split into training and testing sets (80/20 split).
- StandardScaler was used to normalize the input features.

Step 2: Model Development

- A **Random Forest Regressor** was used due to its robustness and accuracy for regression tasks.
- The model was trained on the scaled data and evaluated using Mean Squared Error (MSE).

Step 3: Model Saving

- The trained model and the scaler were saved using the `joblib` library for later use in the Flask application.

Step 4: Web Deployment

- A simple Flask application was created with a route for input and prediction.
- An HTML form was used to capture input from users.
- The model and scaler were loaded, and predictions were made on form submission.

7. Web Application Workflow

1. User visits the home page (`/`).
2. User enters the input parameters in the HTML form.
3. On submission, the form sends data to the `/predict` route.
4. The Flask app processes the input, scales it, and passes it to the trained model.
5. The predicted temperature is displayed on the same page.

9. Evaluation Metric

Mean Squared Error (MSE) was used as the evaluation metric to measure the accuracy of predictions. The model achieved a low MSE on the test set, indicating reliable performance.

10. Future Scope

- Extend prediction to other temperature targets such as stator_yoke, stator_tooth, and stator_winding.
- Integrate real-time IoT sensor input using APIs.
- Enhance the user interface using Bootstrap or JavaScript.
- Deploy the application on cloud platforms like Heroku, Render, or AWS.
- Implement deep learning models for higher accuracy.

11. Conclusion

The project successfully demonstrates the use of machine learning in predicting motor temperatures using real-time input parameters. It integrates a complete pipeline from data preparation, model training, and evaluation to deployment via a user-friendly web interface. This approach can be extended to real-time systems for predictive maintenance in industrial motors.

12. References

- Electric Motor Temperature Dataset: https://www.kaggle.com/datasets/wkirgsn/electric-motor-temperature
- Scikit-learn Documentation: https://scikit-learn.org/
- Flask Documentation: https://flask.palletsprojects.com/