The University of New Mexico

School of Engineering

Computer Science Department

CS 521 - Data Mining Techniques

Project report

(Predictive Analytics for Electric Car Motor

Temperature Management)

Fall 2024

Prof. Abdullah Mueen

1. Introduction

The motors of electric cars produce a lot of heat when running as any mechanical engineer would infer. The motor temperature plays an important role as you want to ensure optimized efficiency, safety and durability. There are three main issues that occur when a motor is overheated: decreased efficiency, component deterioration and system failure. Predictive analytics can guess future occurrences of overheating and flag suspicious activity in no time to tackle the problem.

Importance:

- Improves motor efficiency by preventing overheating.
- Enhances vehicle safety and reliability.
- Reduces maintenance costs and increases motor lifespan.

Impact of Solution:

Solving this problem enables the creation of smart motor temperature monitoring systems which can be incorporated into electric vehicles. Such systems would enhance efficiency, minimize risks inherit in operations and increase user convenience.

2. Data

Original Data Collection:

The dataset was acquired through sensors installed within electric motors, measuring parameters like stator winding temperature, coolant temperature, permanent magnet temperature, ambient temperature, etc. This data reflects real world operational conditions.

Data Acquisition:

The dataset was obtained from a repository from Kaggle.

Preprocessing Steps:

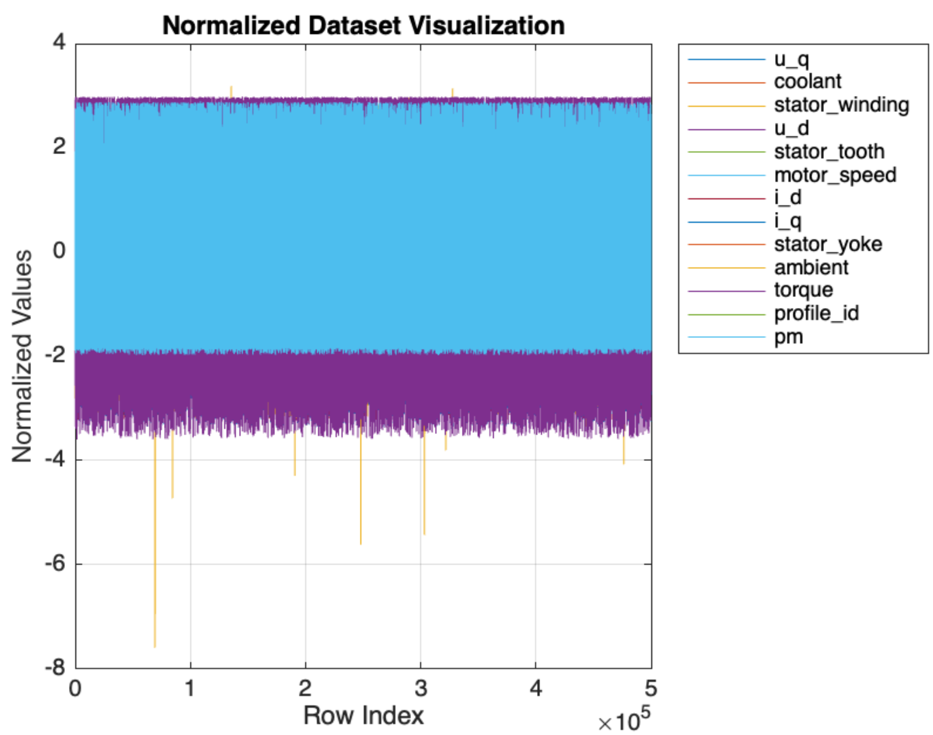
1. **Handling Missing Data:**
   Missing values are replaced with the mean of respective columns:
   $$x_{\text{new}} = \text{mean}(x_i)$$
2. **Shuffling Data:**
   All the rows are shuffled to eliminate bias during training and testing.
3. **Normalization:**
   Data was scaled to [0,1] to ensure equal contribution of features:
   $$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Sample Data:

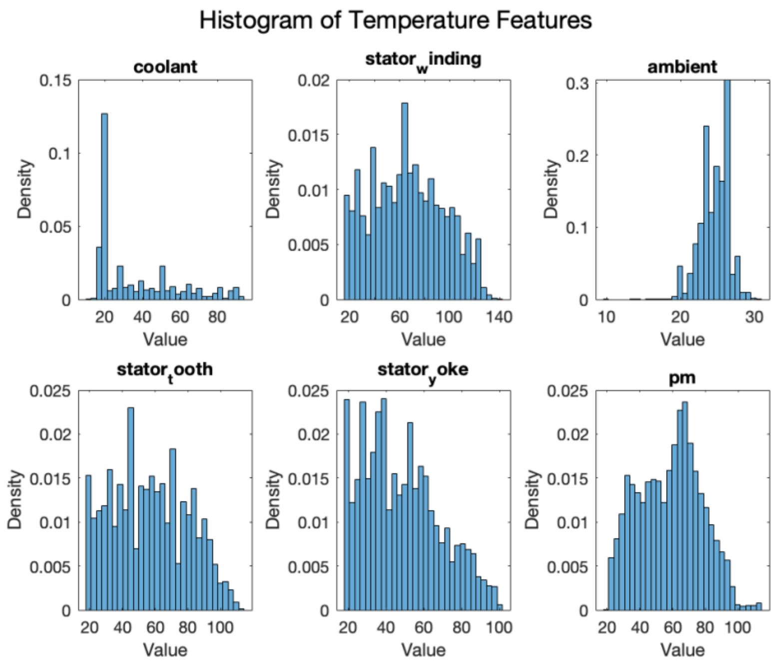
u_q	coolant	Stator winding	u_d	Stator tooth	Motor speed	i_d	i_q	Stator yoke	ambient	torque	id	Permanent magnet
86.46	65.03	104.79	-98.28	91.93	4999.95	-132.62	54.13	78.84	25.71	47.5	6	90.43

Preliminary Statistics:

- **Feature Distribution:**
  - Mean Coolant Temperature: 36.25
  - Mean Stator Winding Temperature: 66.35
  - Mean Permanent Magnet Temperature: 58.56
- **Feature Visualization:**
  - **Normalized Dataset Visualization:**



  - **Histogram of Temperature Features:**



3. Algorithm

Task 1: Predictive Modeling

- **Algorithm:** Liner Regression.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n +$$

- **Parameters:**
  - Training set size: 80% of data.
  - Testing set size: 20% of data.
- **Toy Example:**
  - **Feature:** Coolant temperature (30℃), Stator Winding Temperature (55℃).
  - **Prediction:** Linear Regression predicts the temperature of Permanent Magnet as 65℃ based on learnt coefficients.

Task 2: Anomaly Detection

- **Algorithm:** DBSCAN (Density-Based Spatial Clustering of Applications with Noise).
  - Identifies clusters in feature space and flags points outside clusters as anomalies.
  - **Core Points:**
    - A point ‘p’ is a core point if at least minPts points (including itself) are within a radius ε of it.
  - **Directly Density-Reachable:**

- A point ‘q’ is directly density-reachable from p if p is a core point and q is within a distance ε from p.

- **Density-Reachable:**
  - A point q is density-reachable from p if there is a chain of points p_1, p_2,…., p_n where p_1 = p and p_n = q, such that each p_{i+1} is directly density-reachable from p_i.

- **Noise:**
  - Points not reachable from any core point are classified as noise or outliers.

- **Formulas:**
  - **Distance Calculation:** Euclidean distance is commonly used to calculate the distance between two points p = $(p_1, p_2, \ldots\ldots, p_n)$ and q = $(q_1, q_2, \ldots\ldots, q_d)$:

$$\boldsymbol{d(p,q)} = \sqrt{\sum_{\boldsymbol{i=1}}^{\boldsymbol{d}} (\boldsymbol{p_i} - \boldsymbol{q_i})^{\boldsymbol{2}}}$$

  - **Core Point Check:** For a point P , check if:

$$|\{q \in D: d(p,q) \leq \epsilon\}| \geq \text{minPts}$$

Where, D is the dataset

  - **Clustering:**
    - Assign clusters based on density reachability.
    - Points with no cluster assignment and not density-reachable are labeled as noise (-1).

- **Parameters:**
  - Epsilon: 0.1
  - MinPts: 5
- **Pseudocode:**

```
Input: Normalized data, epsilon, minPts
Output: Cluster labels, anomalies
For each point in dataset:
   Identify neighbors within radius epsilon.
   If neighbors >= minPts, assign to a cluster.
   Else, mark as anomaly.
End
```

4. Experimental Results:

Execution Time:

Task	Execution Time (seconds)
Predictive Modeling	0.70654
Anomaly Detection	3769.0394
Total	3776.2439

Accuracy (Predictive Modeling):

- Mean Squared Error (MSE):

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

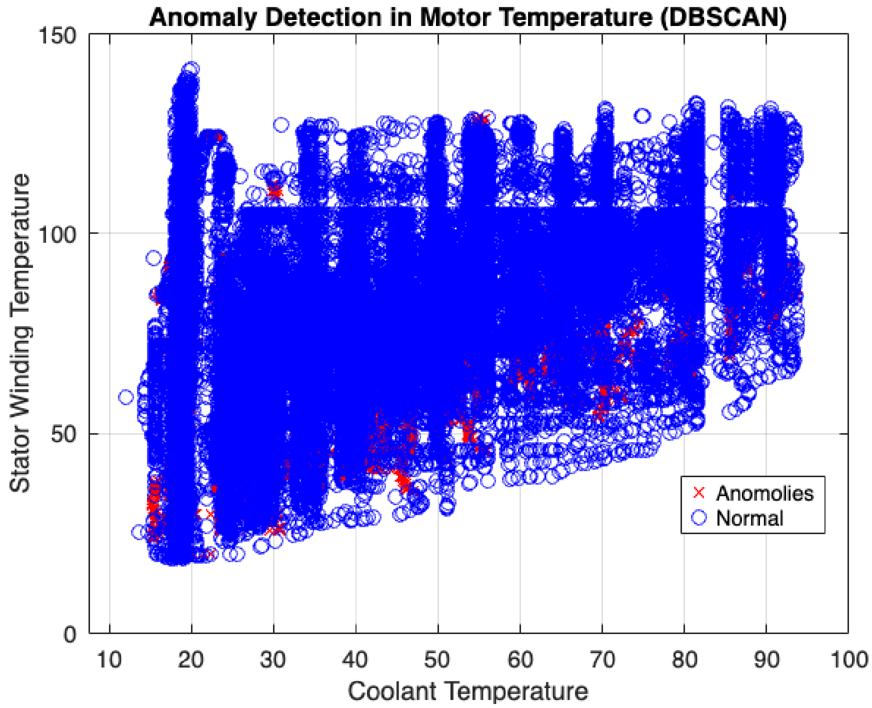
Achieved MSC: 0.51151

Anomaly Detection Metrics:

Metric	Value
Precision	0.53032
Recall	0.53032
F1-Score	0.53032

Visualization:

- Scatter plot highlighting anomalies (Task 2)



5. Discussion

Key Learnings:

- Linear Regression is a fast and interpretable technique for predictive modelling.
- DBSCAN achieves effective results in anomaly detection in multidimensional spaces, but it is sensitive to parameter tuning.

Improvements for Next Iteration:

- Incorporate advanced models like neural networks for predictive modeling.
- Use grid search to optimize DBSCAN parameters for better anomaly detection.