Artificial Intelligence based Thermal Management of the Electric Motor Drive for Off-Road Vehicles

M.Tech. Thesis

By
MAITREYA JAIN
2102106007

CENTER FOR ELECTRIC VEHICLE AND INTELLIGENT TRANSPORT SYSTEMS
INDIAN INSTITUTE OF TECHNOLOGY INDORE

MAY 2023

Artificial Intelligence based Thermal Management of the Electric Motor Drive for Off-Road Vehicles

A THESIS

Submitted in partial fulfillment of the
requirements for the award of the degree
of
Master of Technology

by
MAITREYA JAIN
2102106007

CENTER FOR ELECTRIC VEHICLE AND INTELLIGENT TRANSPORT SYSTEMS
INDIAN INSTITUTE OF TECHNOLOGY INDORE

MAY 2023

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **Artificial Intelligence based Thermal Management of the Electric Motor Drive for Off-Road Vehicles** in the partial fulfillment of the requirements for the award of the degree of **Master Of Technology** and submitted in the **Center for Electric Vehicle and Intelligent Transport Systems, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from August, 2021 to May, 2023 under the supervision of Prof. Trapti Jain, Professor at Indian Institute of Technology Indore, Indore, India and Mr. Vivek Nannajkar, Staff Architect at John Deere India Pvt. Ltd., Pune, India.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

Maitreya Jain 25-05-2023

Signature of the student with date
(MAITREYA JAIN)

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Jain 25/05/23

Signature of the Supervisor with date
(Prof. Trapti Jain)

01-06-2023

Signature of the Supervisor with date
(Mr. Vivek Nannajkar)

Maitreya Jain has successfully given his M.Tech. Oral Examination held on **18th May 2023**.

Jain 25/05/23

01-06-2023

(Prof. Trapti Jain) **(Mr. Vivek Nannajkar)**
Signature(s) of Supervisor(s) of M.Tech. thesis with date

01-06-2023
(Dr. Sunil Kumar)
Signature of Convener, DPGC with date

1/6/2023

1 June 2023

(Prof. Aruna Tiwari) **(Dr. Vivek Kanhangad)**
Signature(s) of PSPC Member(s) with date

01/06/2023
(Prof. Amod C. Umarikar)
Signature of Prof. In-Charge CEVITS with date

ACKNOWLEDGEMENTS

I am grateful to the Center for Electric Vehicle and Intelligent Transport Systems at Indian Institute of Technology (CEVITS), Indore and John Deere India Pvt. Ltd. for giving me the opportunity to pursue Master of Technology course. The inter-disciplinary course has instilled in me the capabilities to understand systems engineering as a whole.

My supervisors, Prof. Trapti Jain and Mr. Vivek Nannajkar, have been an invaluable source of guidance and support throughout my research work. Their expertise and feedback have been instrumental in shaping my thesis and improving its quality. They have shared their knowledge, and motivated me to pursue my research objectives.

I want to express my gratitude to Prof. Aruna Tiwari and Dr. Vivek Kanhangad, who were part of my research progress committee. They have provided me with valuable feedback and suggestions that helped me enhance my work at different stages.

I would like to express my sincere gratitude to the Head of the Department, Dr. Amod C. Umarikar and Dean, Dr. Devendra Deshmukh, for their continuous support and encouragement. Their leadership and vision have been critical in creating a culture of research and innovation in the department and providing the necessary resources for our academic pursuits.

I acknowledge the support of my managers at my current working organization. Their understanding and encouragement have allowed me to balance work and research and have provided me with the resources necessary to complete my thesis.

My colleagues and classmates have also played an important role in my academic journey. Their discussions and feedback have helped me broaden my perspective and enhance my research work.

I am also thankful to my family for their support and encouragement. Their love and support along with their follow ups regarding my study and thesis progress have been a constant source of motivation throughout my academic journey.

Lastly, I would like to acknowledge the contributions of my wife, Saloni Jain, for her patience and understanding during my research work. Her unconditional support and encouragement have been critical in helping me overcome challenges and complete my thesis.

MAITREYA JAIN

DEDICATION

I would like to dedicate this thesis to both my parents Dr. Raka Jain and Prof. Vijay Kumar Jain. My mother has always wished me to pursue my higher education and that zeal in her has always made me realize the value of education. My father on the other hand is the sole inspiration figure in my life to “Live Life Lead Life”.

With love and gratitude,

MAITREYA JAIN

Abstract

The advancement in the microcontroller technology with time, have improved its processing power along with its power efficiency, in addition to enhanced memory and communication capabilities. These capabilities, opens the pathway for the integration and usage of Artificial Intelligence (AI) into the embedded systems, which enables its usage into the real-time applications of automotive Electric Vehicles (EVs). Accordingly, this thesis work highlights the behavior of enlisted Machine Learning (ML) algorithms, which when applied to the target vehicle i.e., Off-Road Vehicles such as electric tractors, to achieve application requirement needs.

Automotive industry is taking efforts to migrate towards EV, taking a step towards sustainability. Electric motor drives play a key role in the architecture of EV. With this importance of electric motor drives, need arises in terms of its safe operation during the lifecycle of the vehicle. Different vehicle protection measures are to be employed to prevent its failure due to thermal stress i.e., motor temperature.

Detection of abnormalities in terms of rise in temperature above warning or critical motor temperature, shall allow the longetivity of the motor and lead to vehicle performance under stress conditions both physically and internally. To achieve the same, ML models were identified after doing literature study and trained on available bench mark data and then applied to actual target vehicle.

Two ML algorithms i.e., Extreme gradient boosting (XGBoost) and Random Forest Regressor (RFR) along with two Deep-Learning (DL) i.e., Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) algorithms are considered in this thesis work, to understand algorithms' behaviour and evaluate algorithms' performance in both ML and DL based models, when trained with real target vehicle datasets recorded from electric motor drives used in off-road vehicles.

XGBoost demonstrated the promising results when compared to targeted models and proved its feasibility in predicting the motor temperature when used for electric motor drives for off-road vehicles. This evaluation was done using the performance metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), R Squared (R2) and Root Mean Squared Error (RMSE).

Index Terms: Electric Vehicles (EVs), Thermal Management, Motor Temperature, Time Series Analysis, Supervised Learning, Machine Learning (ML), Temperature Prediction

LIST OF PUBLICATIONS

1. Maitreya Jain, Prof. Trapti Jain and Mr. Vivek Nannajkar, "Artificial Intelligence based Thermal Management of the Electric Motor Drive for Off-Road Vehicles", <Where, yet to be identified> (To be submitted).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ACRONYM NAME	DESCRIPTION
AI	**A**rtificial **I**ntelligence
CAN	**C**ontroller **A**rea **N**etwork
CEVITS	**C**enter for **E**lectric **V**ehicle and **I**ntelligent **T**ransport **S**ystems
CNN	**C**onvolutional **N**eural **N**etwork
DNN	**D**eep **N**eural **N**etwork
ECU	**E**lectronic **C**ontrol **U**nits
EDA	**E**xploratory **D**ata **A**nalysis
EM	**E**lectric **M**otor
EVs	**E**lectric **V**ehicles
IIoT	**I**ndustrial **I**nternet **o**f **T**hings
LSTM	**L**ong **S**hort**-T**erm **M**emory Neural Network
MAE	**M**ean **A**bsolute **E**rror
MSE	**M**ean **S**quared **E**rror
OBC	**O**n-**B**oard **C**harger
PMSM	**P**ermanent **M**agnet **S**ynchronous **M**otor
ReLU	**Re**ctified **L**inear **U**nits
RFR	**R**andom **F**orest **R**egressor
RMSE	**R**oot **M**ean **S**quare **E**rror
RNN	**R**ecurrent **N**eural **N**etwork
R2	**R** Squared
tanh	**Tan**gent **H**yperbolic function
XGBoost	E**X**treme **G**radient **Boost**ing
VCU	**V**ehicular **C**ontroller **U**nit

Chapter 1

1 Introduction

This thesis work aims to:

- Use available machine learning (ML) or deep-learning (DL) regression-based predictive algorithms to assist the applications in taking action based on available data from the mounted sensors and motors.
- Identify the key steps and different stages of the framework that enable the integration of machine learning (ML) concepts into real-world automotive applications.
- Initially, targeted machine learning (ML) algorithms are fine-tuned and trained with the available data sets to prepare the development environment. Later, training and testing are done with the target vehicle datasets.

1.1 Motivation

Technological advancements are one of the key driving factors in the adoption of Electric Vehicles (EVs). The use of EVs has not only reduced vehicle emissions but also increased efficiency compared to internal combustion engines [4]. Accordingly, induction motors are used for the development of EVs. The use of electric motors, such as PMSM (Permanent Magnet Synchronous Motor) [1], may become exhaustive with vehicle drive time and conditions. This leads to the motivation of our *first* and *second objectives*, i.e., to manage the heat of the electric motor used in EV powertrain design and to identify factors that affect the motor's performance.

Enhancements in the capabilities of the microcontroller, such as processing power, memory, and communications, have led to the adoption of artificial intelligence (AI) in real-time embedded applications in the automotive industry. These capabilities allow one to build a lightweight AI model that is compatible with available ECU resources or use *edge computing* along with IIoT and cloud services in machine learning (ML) applications [10]. The target vehicle aims to use the output from the lite model in decision-making as one of its features. This leads to the motivation for *the third objective*, i.e., to develop an AI-enabled feature that will predict the critical temperature and failure time of the motor. Based on this information, we reach our *fourth objective*, i.e., where the feature shall take corrective measures based on predicted behaviour.

1.2 Thesis Objectives In-Scope

Key research objectives were identified that are within the scope of this thesis work. The above-mentioned brief background description, with the aim in Section 1 and motivation in Section 1.1, helps to list in-specific objectives leading to the building of the feature requirements and their implementation.

The following list of objectives forms a high-level problem statement to build the required feature.

- To manage heat from the contributing sources in an electric motor for off-road vehicles.
- To identify determining factors on which temperature will affect (degrade) the performance of the motor.
- To predict the behaviour of the target motor based on the above-identified attributes based on data received from sensors and motors by applying target AI algorithms.
- To prevent thermal shutdown based on predicted behaviour, i.e., take corrective action when the warning temperature is reached.

1.3 High Level Feature Execution Layout

Brief objectives, as listed in Section 1.2, give us an idea about the high-level requirements. These objectives help to define the problem statement i.e., to estimate the motor temperature of an electric motor for continuous time-series data. The end-to-end development stages of the application are illustrated in figure 1-1.

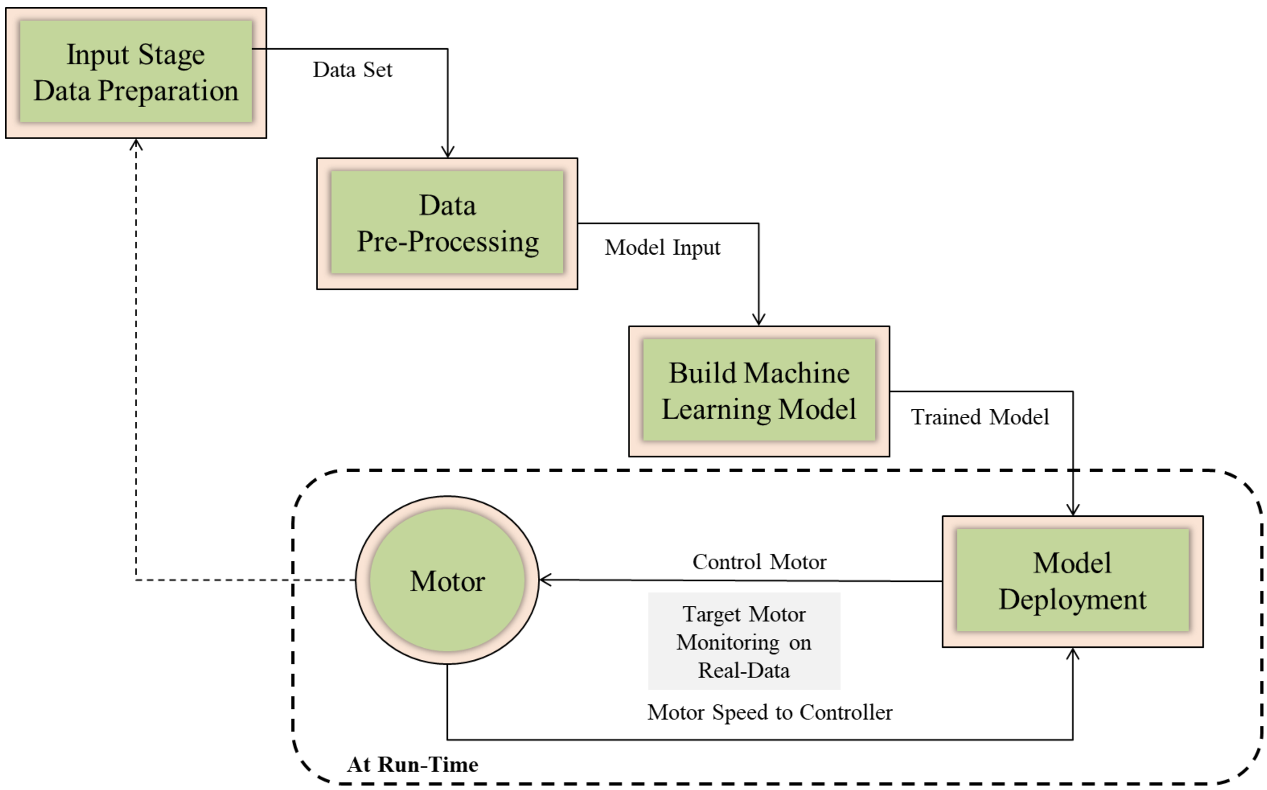

Figure 1-1: High-Level Block Diagram

In the *first stage* (section 3.3.1), the communication interface for the target vehicle is identified, i.e., over the CAN bus (Controller Area Network). The ECU of the target motor controller uses this CAN bus to send the motor information to the requester application. This way, the required information is captured from the target motor and sensors. How the captured data is inferred and other steps are detailed later in Section 3.3.1. Later, in the *second stage* (section 3.3.2), the data set goes through pre-processing and feature engineering to prepare the model input that is required to execute model training. In this stage, the input and target feature attributes are selected or dropped based on the strong correlation matrix of the feature attributes [8].

In the initial stage of thesis work, research papers based on their relevance to the topic were sorted, and a comparative study of the AI algorithms was done where models were trained on data sets obtained from PMSM or Electric Motor. The models were sorted based on the type of supervised learning, i.e., ML or

DL, and their performance results as specified in different study works. Evaluation results were observed quite efficiently when decision-tree-based regression models, i.e., RFR and XGBoost, were employed. Similarly, LSTM and 1-D CNN were shortlisted based on their applicability even in applications of regression-based prediction.

In the *third stage* (section 3.3.3), four models, i.e., RFR, XGBoost, LSTM, and 1-D CNN, are identified, and their hyperparameters are fine-tuned, trained, and evaluated to select an appropriate model that meets the objectives (section 1.2) efficiently. Their performance is evaluated based on four metrics, namely MAE, MSE, R2 score, and RMSE. These metrics were chosen to compare the results from available model applications in similar environments and with similar objectives.

In the *final stage* (section 3.3.4), the trained model is then used to predict the motor temperature and the failure time. The vehicle ECU shall take corrective measures based on the prediction information. It may control the motor speed or turn it off based on threshold conditions, which prevents motor damage and enhances the longevity of the motor. This way, the application of target models is meant to satisfy the thesis objectives.

1.4 Organization of the Thesis

The aim is to give the reader an idea of how the thesis is structured and its brief summary. The subsequent chapters of the thesis will contain a detailed and in-depth analysis of the research topic.

Chapter 2: Provides a comprehensive review of the relevant literature on the thesis topic. Identifies the four models that can be used to achieve the objectives.

Chapter 3: Industrially applied powertrain architecture is explained. Its detailed system overview is given and emphasizes the in-scope features to be considered for the application of the selected ML model. The deployment of the model in a real-time application is discussed with an end-to-end system architectural diagram.

Chapter 4: This chapter selects the target models filtered from available research work with similar problem scopes. It explains, in brief, the selected four models.

Chapter 5: This chapter details the design and how experiments are conducted. It first highlights in a procedural manner how analysis is done for the PMSM and target vehicle datasets. It then further discusses the results of the different models applied and their performance.

Chapter 6: This chapter interprets the results of the implementation and discusses their implications for the research field. It also provides a critical reflection on the research process and identifies potential limitations and areas for future research.

Chapter 7: Summarizes the main findings of the research and restates the contributions of the research work. It also discusses the significance of the research and its potential impact on the upcoming research benchmarking the empirical results obtained from the target vehicle. In the end, it highlights the areas where it can be further worked upon as a future scope.

Overall, this section provides a roadmap for the reader, highlighting the key content of each chapter and how they fit together to satisfy the objectives of the project.

Chapter 2

2 Literature Survey

In the first part of this chapter, a review of research papers is done, and appropriately relevant models are identified based on their applicability to a similar problem statement as ours. Finally, we discuss several metrics used to evaluate the performance of our model, along with normalization techniques.

2.1 Review of the Past Work

In reviewing past work, our problem statement focuses on developing a system warning that is notified in case of any abnormality in the temperature of the electric motor used for off-road vehicles, i.e., electric tractors. Hence, our problem statement narrows down based on where it is applied, the application requirement, and the type of data. The data collected from our target vehicle is from different sensors internally and is continuous time-series data. The application is for the prediction task and is applied to embedded systems with Vehicular Controller Units (VCU).

Similarly, the problem statement is addressed by Kirchgassner *et al.* for the prediction of abnormalities. Also, it highlights the challenges of applying deep learning to the prediction of temperature for monitoring purposes [6]. The research paper also benchmarked the state-of-the-art datasets captured from the Permanent Magnet Synchronous Motor (PMSM) for different test runs. This availability of datasets has allowed us to perform experiments on applying AI models using these datasets. Their performance with real-time applications gives us the confidence to utilize the AI capabilities in our embedded systems with limited resources.

Li *et al.*, in their research study, proposed using deep learning-based LSTM models provided the input-output feature relationship is known [1]. Here, they used the average absolute correlation values to select attributes. Torque is dropped from the input features because of its low *AvgAbsCorrCoef* value of 0.089. MSE is used as a critical evaluation metric. Their proposed LSTM-based models' MSE and MAE results are compared (section 6.1) with our experiment results to prepare our baseline project, which is applied to the target vehicle at a later stage.

Hosseini *et al.*, in their research, have proposed LSTM and 1-D CNN based models applied to the same PMSM datasets [5]. The CNN model accurately predicted the desired target values with high precision and an average MSE of 2.64$°C^2$, as per their experimental results.

Al-Gabalawy *et al.*, in their research, have used SVM and XGBoost models for temperature prediction [4] and MSE and RMSE as the evaluation metrics. The test RMSE value of 0.589 for SVM was the lowest among the other applied models in their work. The XGBoost test and train RMSE values were found to be 0.829 and 1.226, respectively.

Kim *et al.*, in their research, have focused on developing an optimized predictive maintenance model based on LSTM for machinery's bearing components [8]. They have specifically worked on the tuning of LSTM design hyperparameters.

Sampaio *et al.*, in their research, have applied the Random Forest Regression model [2]. They have worked on the estimation of failure time. RMSE is used as an evaluation metric to examine performance.

The study by Wallscheid *et al*. involved exploring the potential of recurrent neural networks (RNNs) to accurately predict the temperature of PMSMs. Particle swarm optimization was utilized in their work to determine appropriate hyper-parameters, such as the number of hidden layers and neurons.

Savant *et al.*in their research, have applied SVM, Polynomial regression and RFR and evaluated the results using R-Squared metric for Stator Winding Temperature and Rotor temperature and Torque. Out of the three RFR performed better, R-Squared values of stator winding for the three were 0.936, 0.993, and 0.932 respectively.

2.2 Normalization Techniques

2.2.1 Standard Scaler

Standard Scaler, also known as Z-score normalization, is a commonly used normalization technique in data preprocessing. It rescales the features in such a way that their standard deviation is one and their mean is zero, resulting in a distribution that is centered around zero. The mathematical expression [4] for the standard scaler is:

$$x_{scaled} = \frac{x_i - \mu}{\sigma} \tag{1}$$

where mean (μ) is given by:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} (x_i) \tag{2}$$

where standard deviation (σ) is given by:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2} \tag{3}$$

Here N, x_i, and i denote the number of samples, the original value, and the sample index. It helps reduce the effects of outliers and improves the performance of an optimization algorithm.

2.2.2 Min-Max Scaler

Min-Max Scaler is a popular normalization technique used in data preprocessing to rescale the values of a feature into a fixed range between 0 and 1. It transforms the data such that the minimum and maximum values are 0 and 1, respectively, with all other values scaled proportionally between these two values. The mathematical expression [8] is:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4}$$

Where x_{min} and x_{max} are the values of the attribute to be normalized, i.e., the original value (x). It can help improve the convergence of some optimization algorithms and reduce the effects of outliers. The Min-Max Scaler may not be suitable for datasets with extreme outliers or a non-normal distribution. Z-score (standard scaler) normalization may be used instead. It is recommended that it be applied separately to training and testing datasets to avoid data leakage and overfitting.

2.3 Evaluation Metrics

MAE, MSE, RMSE, and R-squared are commonly used evaluation metrics in regression analysis to measure the performance of predictive models. Each metric provides unique insights into the strengths and weaknesses of a model.

2.3.1 Mean Absolute Error (MAE)

MAE is a metric used to measure the size of errors in a group of predictions. It's computed by finding the average absolute difference between the predicted and actual values. The formula for calculating MAE is:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{5}$$

Here N, $\hat{y}$, y_i, and i denote the number of samples, the predicted value, the actual value, and the index of the sample. MAE is beneficial to evaluate a model's performance in the presence of outliers, as it is less affected by them than RMSE. A lower value of MAE indicates that the model is more accurate and better at predicting the target variable.

2.3.2 Mean Squared Error (MSE)

The real-value predictions are often evaluated by using an evaluation metric such as Mean Squared Error (MSE). The MSE is used to calculate the average of the squared difference between predicted and true values, and it is a standard metric for regression tasks. The mathematical expression for MSE is:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{6}$$

Here N, $\hat{y}$, y_i, and i denote the number of samples, the predicted value, the actual value, and the index of the sample. A lower value of MSE indicates that the model is more accurate and better at predicting the target variable. MSE is particularly useful when the data does not contain outliers, as it is more sensitive to outliers than MAE. MSE is a widely used metric in machine learning (ML) algorithms, as it can be used as a loss function to optimize the parameters of a model.

2.3.3 R Squared (R^2)

R-squared is a statistical measure used to evaluate the goodness-of-fit of a regression model to the data. It helps to understand variance proportionality in the target and independent variables. It can take values between zero and one, where one indicates a perfect fit of the model to the data. The mathematical expression [4] for R-squared is:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} \tag{7}$$

here N, $\hat{y}$, y_i, $\bar{y}_i$ and i denote the number of samples, the predicted value, the actual value, the mean value of the dependent variable, and the index of the sample. It must be noted that R-squared can be influenced by outliers and may not be appropriate in all situations, such as when the data is not normally distributed or when there are nonlinear relationships between the independent and dependent variables.

2.3.4 Root Mean Squared Error (RMSE)

A statistical method called Root Mean Squared Error (RMSE) is employed to evaluate the accuracy of a predictive model by measuring the average distance between the actual and predicted values. It calculates the square root of the average squared differences between them. The mathematical expression [2] for RMSE is:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{8}$$

Here N, $\hat{y}$, y_i, and i denote the number of samples, the predicted value, the actual value, and the index of the sample. RMSE helps evaluate the performance of a predictive model because it measures the average magnitude of the errors in the model's predictions. It helps understand the spread of the errors and can be compared to the range of the target variable to determine the relative size of the errors. A lower value of RMSE indicates that the model is more accurate and better at predicting the target variable i.e., motor temperature. Similar to R-Squared, RMSE can also be influenced by outliers.

2.4 Summary

In this chapter, we have done a survey of the methods for normalization techniques and evaluation metrics that are to be used for estimating performance of the machine learning (ML) algorithms. Research papers with similar problem statement are studied to understand the different types of algorithms which have been used for estimation of motor temperature for continuous time-series data. Evaluation results from these research papers are to be further used in our thesis work to validate our experimental results from applied machine learning algorithms. PMSM datasets [10] are used for the initial experimentation because they will help evaluate the applied techniques and prevent any rework in the later stage for validation when compared proven results.

Among the above-mentioned study work, except for Sampaio *et al.*, all research papers have used the PMSM dataset for conducting experiments and have proposed different strategies at different stages of the development cycle. Their research studies clearly tell us that the estimation of the thermal temperature of an electric motor can be achieved with precision and accuracy. Keeping this in mind, we consider these studies in the later stage to identify our target models, which are to be applied to target vehicles for off-road vehicles, i.e., electric tractors.

Chapter 3

3 System Description

This chapter follows the top-down approach in highlighting the requirement of the system. Aim is to highlight and explain where the AI algorithms are to be applied in the system and give reader clear insights about the target vehicle and its in-scope components.

3.1 Powertrain Layout of Target Vehicle

The two-motor variant powertrain design of the target electric vehicle is illustrated in figure 3-1. In a general study, it is observed that energy consumption for a dual motor is found to be better than that of a single motor, as in [12].

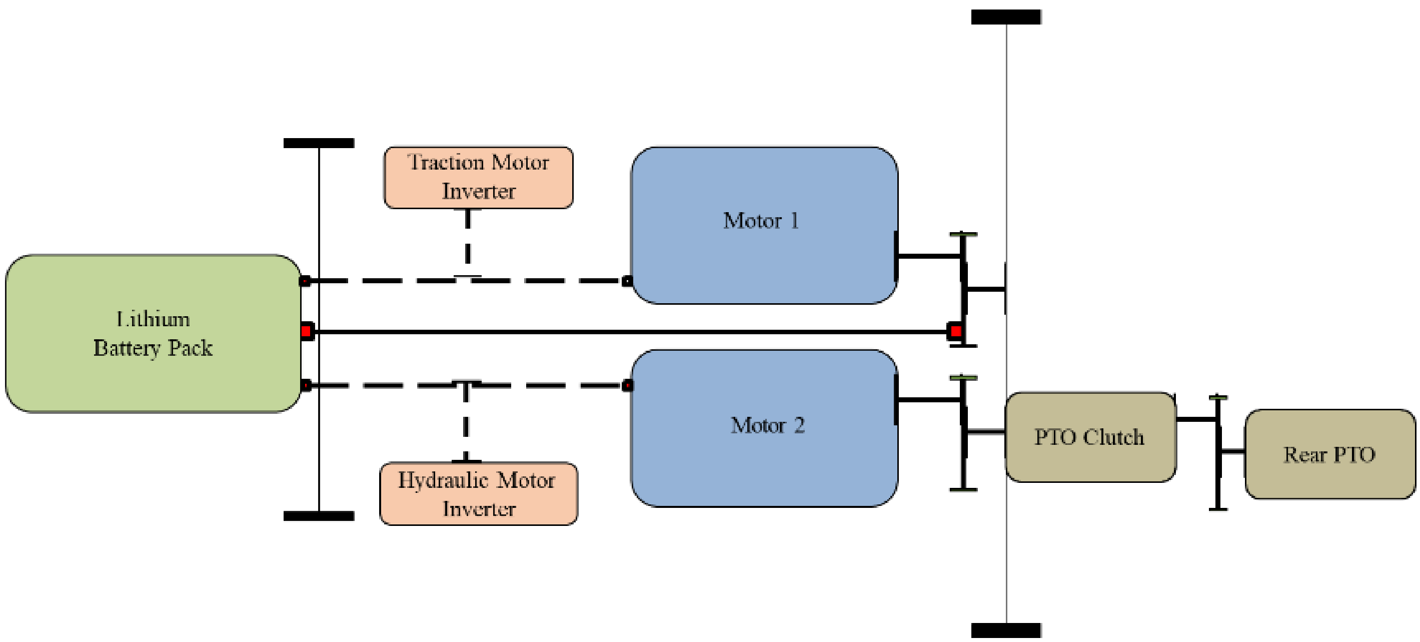

Figure 3-1: Powertrain layout of 2 Motor Variant

- Traction is equipped with a high-capacity lithium-ion battery that provides the necessary power to the motors.
- Motor 1, i.e., the traction motor, is responsible for providing traction to the wheels, allowing vehicle movement.
- Motor 2, i.e., the Power Take-Off (PTO) motor, is responsible for driving the PTO system, which allows the target vehicle, i.e., an electric tractor, to power various agricultural farm implements such as mowers, balers, and plows.
- Two inverters are used to control the speed and torque of their respective motors. Their primary role is to convert direct current (DC) power from the battery to alternating current (AC) power used to drive the motor.

3.2 System Overview

The system design of the target electric vehicle is illustrated in figure 3-2. The system has five main parts that work together to power two inverters.

- There is an onboard charger (OBC) that charges the battery pack using electricity from an external power source.
- The battery pack consists of two batteries connected in parallel and provides the DC power needed for the system.
- The battery management system (BMS) manages and monitors the health and charge level of the batteries to keep them working safely.
- The power distribution unit (PDU) distributes the DC power from the battery pack to different parts of the system, including the two inverters.
- The battery thermal management system (BTMS) keeps the temperature of the batteries at safe levels and can even actively control the cooling or heating if necessary.

All the specified parts work together to ensure the system can power the two inverters safely and efficiently. Combining these parts ensures the system operates without any problems or damage.

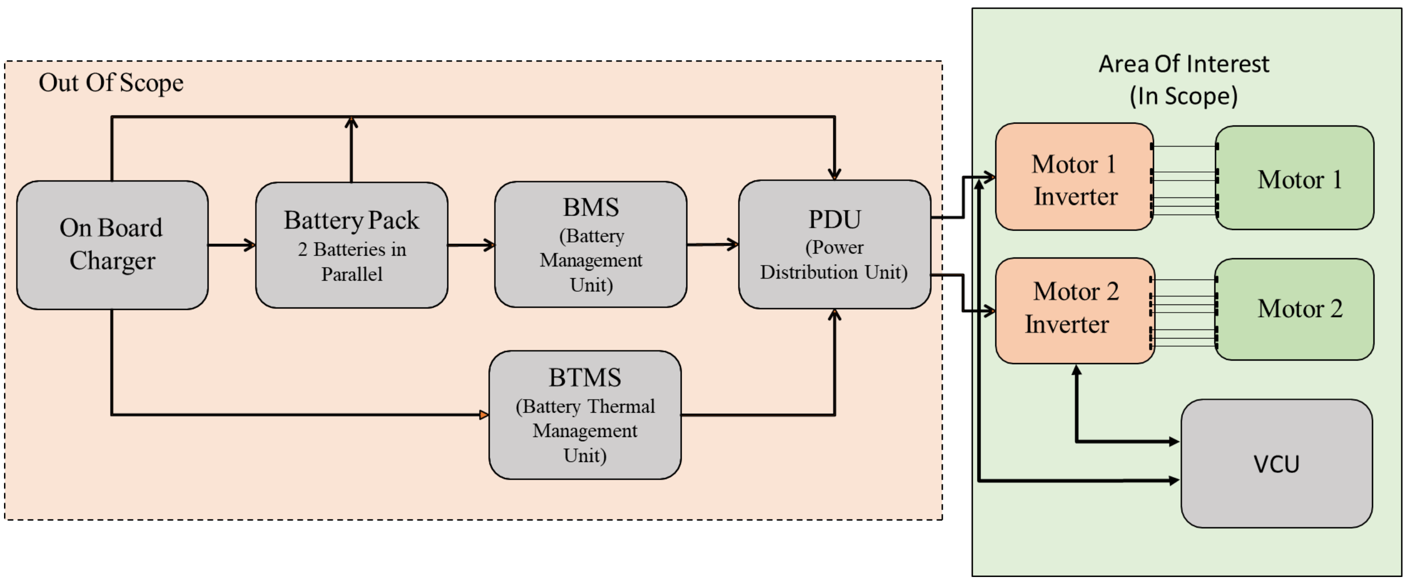

Figure 3-2: System Overview of the target vehicle

3.2.1 Inverter and Motor Interface

As shown in figure 3-1, this section narrows down the system to highlight the specific in-scope area of interest, i.e., inverter-motor interfacing, as shown in figure 3-3. When an inverter powers a motor, there are different ways they communicate with each other, as listed below.

- The inverter sends electricity to the motor in a way that can control its speed and power.
- The inverter sends control signals to the motor to turn it on or off, change its direction, speed, or power, and detect if there is a problem.
- The information (such as phase voltage and current, position, and temperature sensor signals) is exchanged between the inverter and motor using different protocols like controller area network (CAN). Motor controllers receive the data from sensors and motors in the form of analog inputs. The motor controller transmits such signal information over the CAN bus.
- Both the inverter and motor have protection signals that detect problems like overheating or overloading and can shut down the system to prevent damage.

This way, the inverter, and motor communicate with each other to work together effectively and safely. Electric motors for such applications are often paired with controllers, such as *the Curtis 1239E* model *data sheet* [13], for reference purposes only. Refer to the user manual of the example controller to have a clearer understanding of the similar interfaces talked about in this section. Refer controller user manual for details [14] on transmitted and received signals from the motor controller.

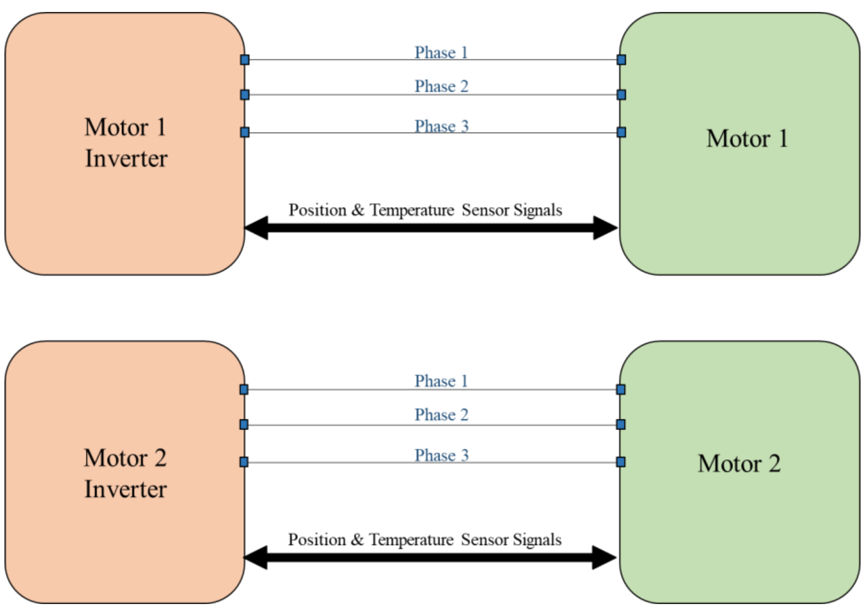

Figure 3-3: Interface diagram of the Motor and Inverter

3.2.2 Target Electric Motor Specifications

The values of induction motors used in industrial applications, such as our target vehicle, with different parameter values in the range are specified in table 3-1. This information further helps in making design decisions with respect to controlling strategies of the motor when the abnormality is estimated.

Table 3-1: Three Phase Induction Motor Specifications

PARAMETER	**VALUE**
Operating Voltage	75-110 V
Operating Speed	3000-8000 rpm
Operating Torque	15-90 Nm
Operating Power	5-15 kw

3.3 Step Flow Diagram

This section elaborates on the high-level feature block diagram, as discussed in Section 1.3. An effort is made to provide detailed steps from data capture to model deployment. The first stage is to capture data, followed by its processing and extraction of features with which applied AI models (section 4) are trained. The trained models are then used for the critical motor temperature estimation with real-time data, and corrective actions are taken by the ECU application, as discussed in detail in Section 3.3.4.

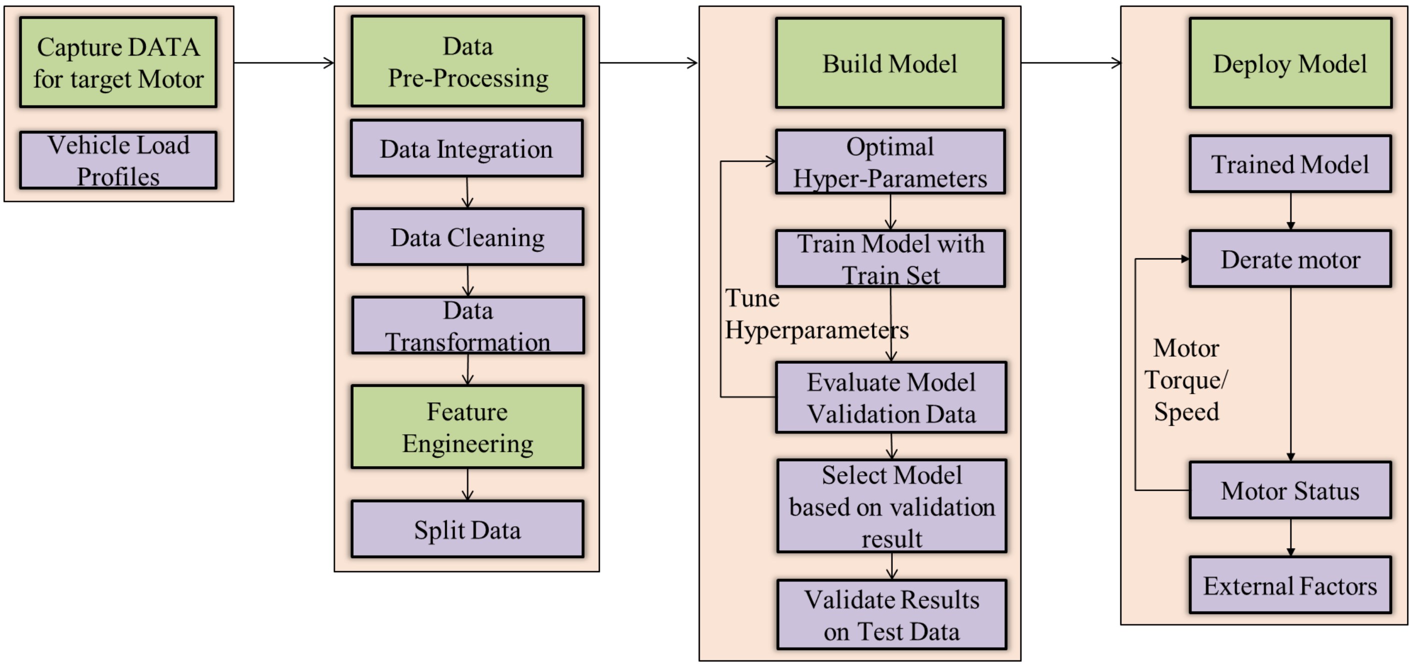

Figure 3-4: Sequential Block Diagram

3.3.1 Capture Data

The process followed is to record and capture information and validate it for different load profiles. The following steps can be followed to capture data in a structural and systematic manner:

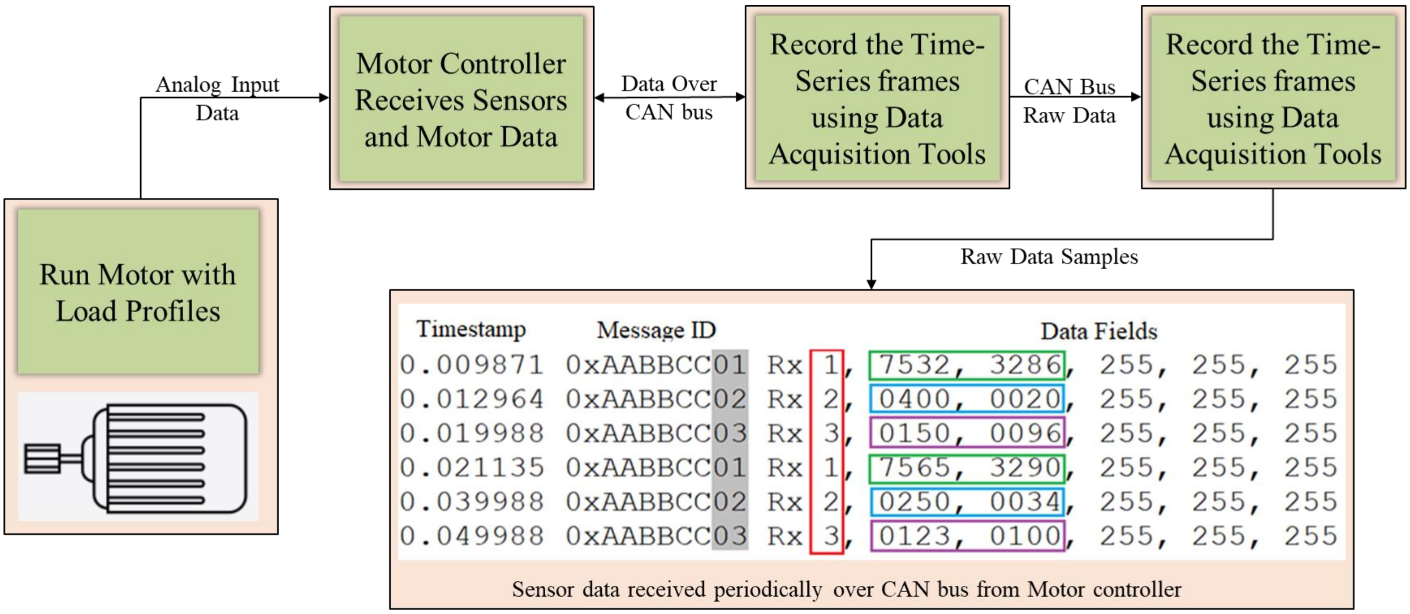

Figure 3-5: Detailed Illustration for Raw Data Capturing

- The scope is to run a motor with different loads, and its purpose is to capture the signal data as detailed in Section 3.2.1, which is relevant to the defined objectives (Section 1.3).
- These signals (phases and temperature) are provided by sensors available in the motor.
- Information is recorded using data acquisition tools when interfaced with the controller area network (CAN) bus. Signal data is distributed into different messages as per the specification of the target motor used.
- Recorded CAN logs contain 8 bytes of informative data each, distributed in different message identifier frames. The ECU requests this signal data through control commands and receives the signal data over the bus. The periodicity of the received message frames can differ according to the requirements. It helps manage the CAN bus and central processing unit (CPU) utilization of the ECU.
- Currently, available data for the target vehicle Motor 2, as detailed in Section 6.1.2, from the lab consists of four load profiles (based on the implements moving speeds in rpm, i.e., 350, 450, 540, and 650).

- The data received is time-series data; hence, this should be taken into account while selecting the AI models to be applied to the given problem statement.
- Since the received signal data is distributed into different CAN messages, the received signal shall be recorded sequentially, and its occurrence shall depend on the defined periodicity of its control signal command. Samples of the data set must be prepared from these logs so that all feature attributes information is available to be used as input.
- Samples obtained are required to be filtered based on message identifiers to group similar signal data together in sequential order with respect to time.
- The raw data set is prepared using these recorded messages from the identified attributes (Table 5-4).
    - Frames are filtered based on message identifiers for the respective signal information present in the respective messages.
    - Data frame format details are provided by *motor controllers. Refer* to an example motor controller user manual [14].
    - This frame format (with unique message identifiers) information, along with data length and resolution, is used to extract all signal information, which constitutes a data set comprising data attributes.

This way, raw data is clubbed together from recorded CAN bus logs and ready to be fed as input to the data processing stage. This is the first stage in the application of AI models, and the quality of the data does play an important role in the training of applied models.

3.3.2 Data Pre-Processing and Feature Engineering

The raw data made available from the data collection stage needs to be processed to prepare a final data set that has all required attributes and derived attributes. To yield an efficient result from data training, data must go through cleaning, smoothing, scaling, transformation, feature engineering, and splitting operations. These stages make it possible for data to be free from any outliers, overfitting, or underfitting.

3.3.2.1 Integrate Raw Data

Raw data from multiple load profiles and sources is combined into a single dataset with the desired formats and resolutions. The main aim here is to filter out inconsistencies and redundancies.

3.3.2.2 Data Cleaning

The integrated data set needs to be processed in order to be ready for model training. For that purpose, the following steps are followed while performing the pre-processing data cleaning operation:

- Duplicates are removed.
- The missing data handling is done. Heat map visualization allows us to identify any pattern in missing values and determine the appropriate method for handling them, such as imputation or deletion.
- Outliers' detection is done by applying various techniques, as mentioned later in Section 5.4.1.3. Box plots are used for visualization purposes.
- Detected attributes with outliers have to be cautiously analyzed based on the available information. They could be genuine extreme values, which can be kept as it is for consideration in modeling, while if they are incorrect values based on observation, then they must be removed or imputed.
- Data smoothing is done using the rolling mean average method for the data [5]. The window size for each attribute varies based on the visualized data.
- Data is imputed for nan values if present after smoothing.
- There are various ways data is imputed based on its adjoining values in that attribute. Methods such as Mean, Median, KNN, and Bayesian Ridge are applied, and their performance is evaluated based on metrics as specified in Section 2.3.

3.3.2.3 Data Transformation

For data transformation, various methods, such as normalization and feature scaling, are used based on the nature of the data. The aim is to have a high-quality data set after transformation is applied without changing its original meaning, which helps in accurate analysis and trains quite well with applied models to predict motor temperature.

Normalization of data is often required when attributes of a dataset have values with different units or ranges and, hence, are likely to vary in their distribution. Such different scales of data during model fitting may introduce bias. High bias during model fitting leads to underfitting. That means the model will not be able to capture relevant relationships between input and target features [5].

Scaling the data means scaling the values of attributes to a specific range or the common range (often used in the range 0 to 1) so that they can be compared and work better with applied algorithms (Section 4). The need to perform scaling operations solely depends on the nature of the data and the models to which they are applied. A tree-based model may not require scaling.

3.3.2.4 Feature Engineering

The intent of doing feature engineering is to identify meaningful data concerning the domain it is applied to, i.e., in our case, data attributes are identified (Table 5-4) based on our main objective (Section 1.2). The aim is to estimate the motor temperature of the target vehicle, Motor 2.

- *The feature selection* process is done to meet the problem statement, as mentioned above. Input features and target features are to be identified.
- The target feature among all attributes is "*Motor Temperature"*, which is received from the motor sensor as analog data and is requested by the Vehicular Control Unit (VCU).
- *'Motor Controller Temperature'* is intentionally chosen as an input feature. The idea is to use its information to predict the target motor temperature.
- Inputs other than the identified target feature are considered input features for training purposes.
- More input derived features can be created based on requirements. For now, in our case, the currently available feature attributes meet the objectives; hence, no such inputs were created or derived based on any domain-specific formula.
- The major goal of feature engineering is to prepare the input required for model training by applying the models identified in Section 4.
- The identified four models are RFR, XGBoost, LSTM, and 1-D CNN.
- An input data set in a numerical matrix shape is required for RFR and XGBoost model fitting.
- Three-dimensional input is required to be fed to the CNN model. Required data must be of 3-D shape with attributes such as the number of samples, time steps, and number of channels.
- While for LSTM, the first input and output data of shape 2-D (samples, dimension) is prepared. Then input data is reshaped to 3-D (samples, time-steps, number of channels) for LSTM model needs.

3.3.2.5 Data Splitting

Engineered datasets are required to be divided based on features into two or more sub-sets. Splitting the data allows high performance in real-time application. Various points are to be taken into consideration while deciding the ratio with which train, test or validation data set. The sub-splitting helps in evaluation the performance and accuracy of the model.

- Data set must be randomly separated so that sub data sets represent entire data set.
- Ratio of the split depends on the size of the data set. Generally, 70-80% is used for training and remaining 30-20% for testing.

Data pre-processing prepared data set is generalized and reliable because different analytical points such as missing values, outliers, smoothing, normalization and scaling along with feature selection and data splitting gives confidence that correct data-set is prepared which is ready for its application.

3.3.3 Build Model

Aim is to build the model which has high accuracy in predicting the motor temperature when real time data is received. After literature study and research comparison of similar problem statement four models are targeted for training and evaluation. Identified four models are namely RFR, XGBoost, LSTM and CNN (Section 4). Based on application needs and performance requirement appropriate model shall be selected for deployment purpose.

- Hyper parameters tuning is done based on experimentation and empirical results of the fine-tuned parameters which produce high accuracy both in training and test data set.
- All four models are trained with their respective hyper-parameters and this operation is iterated until optimal results are obtained while experimentation.
- Hyper-parameters and their optimal values are detailed in section 6.4.
- Model evaluation is done on test dataset and cross-validated to see the accuracy over the complete data sub set.
- Evaluation metrics as identified in section 2.4, namely MAE, MSE, R2 score and RMSE are used to validate the results and performance.

Model is selected based on its accuracy over test and training data set and performance as per evaluation metrics. The selected model after performance evaluation is used for the estimation of motor temperature, when new real time signal data is received for input features.

3.3.4 Deploy Model

Trained models are evaluated based on their performance and selected to be used to meet the objectives of this thesis work in this stage. At run time, the sensors signal data is received, this data is fed as input after transforming with resolution factor for respective signals, to the trained model and motor temperature is estimated along with the failure time. This estimated temperature is checked against the desired warning and critical temperature thresholds as per design. When warning temperature is predicted and is not critical than configured action is executed, for example, control the motor speed of the target motor. Whereas, if critical temperature is estimated which is hazardous for motor and vehicle, then ecu shut-down sequence is initiated.

Before turning off the motor, all necessary information which may be required n next power on cycle is saved in non-volatile memory.

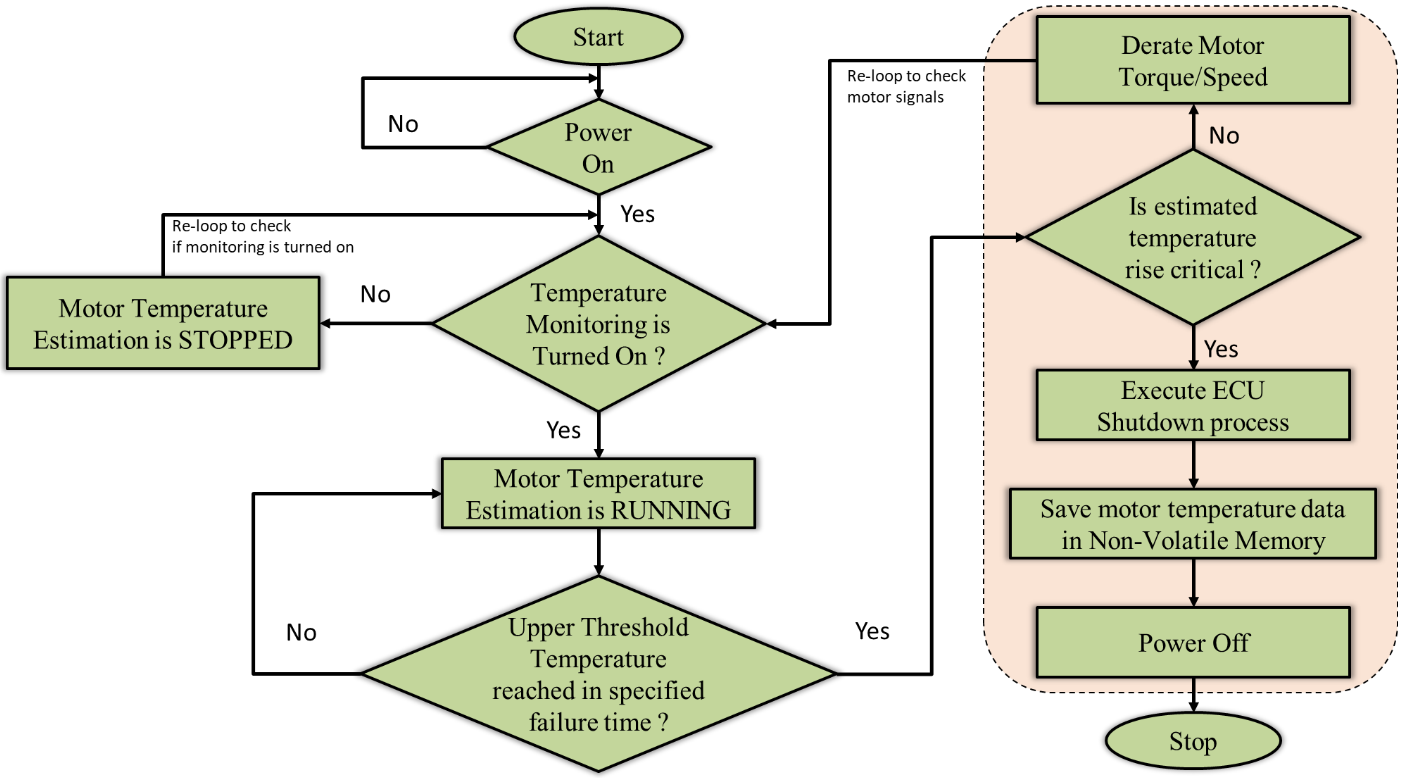

Figure 3-6:Flow Chart Representation of Trained Model Deployment

3.4 Summary

Details discussed earlier in the section gives user a clear idea like what is the target vehicle, where feature is to be utilized. Illustrations are made to explain the system overview and what is the area of interest i.e., inverter-motor and the process followed from capture of data to the deployment. Until now, reader can understand the system as a whole and similar framework can be applied to any embedded vehicular applications. Thermal management strategy is discussed through illustrated diagram (figure 3-6) in section 3.3.4.

Chapter 4

4 Target Algorithms

During our literature review, we examined various research papers (section 2.1) that had similar objectives to ours. In Section 2.1, we discussed these papers and used the information gathered to decide which AI based models would be most suitable for our regression-based prediction task problem. Research has shown that Deep Neural Networks (DNNs) are effective in addressing industrial problems related to regression, as demonstrated in previous studies [1]. This helps us in selection of DNN techniques such as 1-D convolution and LSTM for their application in prediction of PMSM temperature.

A data-driven approach such as a Neural Networks (NN) or black-box model which does not rely on motor data-sheet information [13] [14]. It is based solely on empirical measurements and can avoid estimation of errors even when physical model assumptions are not met during operation [10].

We have considered various factors, including the performance of the models on continuous time-series data. To gain a better understanding of the behavior of these models when applied to real-time applications, we selected both ML and DL models. We ultimately chose four models as our target models, namely Random Forest Regression (RFR), Extreme Gradient Boosting (XGBoost), Long Short-Term Memory (LSTM), and 1-D Convolutional Neural Network (CNN).

4.1 Random Forest Regressor

Random Forest Regressor (RFR) is a supervised learning algorithm used for solving regression problems. It is a type of ensemble learning method that constructs a multitude of decision trees at training time and outputs the mean prediction of the individual trees as the final prediction. Basic steps of how RFR model works are as follows:

1. Create a random sample of the original dataset using bootstrap sampling.
2. Build a decision tree using the bootstrap sample, recursively splitting the data based on the feature that results in the largest reduction in variance.
3. Repeat steps 1 and 2 for a specified number of trees, each trained on a different bootstrap sample and using a different subset of the available features.
4. Output of the final prediction of the random forest is obtained by averaging the predictions of all individual trees.

RFR is a popular algorithm due to its ability to handle high-dimensional data with complex relationships between the features and the target variable. It is also resistant to overfitting, as each tree is constructed on a random subset of the data, and the final prediction is obtained by averaging the predictions of multiple trees.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \tag{9}$$

Where N denotes number of data points, i is data point, $\hat{y}_i$ is the predicted value at each step for given input y_i. The mean squared error (MSE) is utilized to measure the deviation of the data from each node. Mentioned formula (Eq. 9), calculates the distance of each node from the predicted and actual value, which helps to determine the better decision branch for the forest.

Hyperparameters of the RFR, such as the number of trees, the maximum depth of the trees, and the number of features to consider at each split, can be tuned to improve the performance of the algorithm on a given dataset. Cross-validation techniques are used to find the optimal values for these hyperparameters. Overall, RFR is a versatile and powerful algorithm that can be used for a wide range of regression tasks.

4.2 Extreme Gradient Boosting

XGBoost is an ensemble learning algorithm that utilizes gradient boosting with trees. Is widely recognized for its speedy execution time [18] and high-performing algorithm for supervised learning applications [4]. It is commonly utilized in regression prediction and has demonstrated exceptional performance in numerous ML evaluations.

XGBoost is to be applied in the context of regression based supervised learning task, where the objective is to predict a target variable (y_i) based on input training data (x_i) that consists of multiple features. [16]. Objective function can be defined as:

$$obj(\theta) = L(\theta) + \Omega(\theta) \tag{10}$$

where train loss function is denoted by L and regularization term is Ω. The measure of how well our model predicts the training data is referred to as the training loss. The mean squared error (MSE) is a commonly used metric (detailed in section 2.3.2) to calculate this loss. Whereas, the regularization component (Ω) of the model regulates its complexity and helps to prevent overfitting. With MSE as loss function, objective function in eq. 10 can be written as:

$$obj^{(t)} = \sum_{i=1}^{N} \left(y^i - \left(\hat{y}_i^{(t-1)} + f_t(x_i)\right)\right)^2 + \sum_{i=1}^{t} \omega(fi) \tag{11}$$

Where i, t, N, y^i, $\hat{y}$, f_t, x_i and ω are number of trees, iteration number, number of samples, actual output, predicted output, input data, and i_{th}tree respectively. This way with the objective function we can determine how good the particular tree is. Taylor expansion of the loss function up to the second order for eq. 11 can be written as:

$$obj^{(t)} = \sum_{i=1}^{N} [l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \omega(f_t) + \text{constant} \tag{12}$$

Where g_i, and h_i are equated as:

$$g_i = \partial_{\hat{y}_i} l\left(y_i, \hat{y}_i^{(t-1)}\right) \tag{13}$$

$$h_i = \partial_{\hat{y}_i}^2 l\left(y_i, \hat{y}_i^{(t-1)}\right) \tag{14}$$

First and second order gradients can be written as:

$$G_i = \Sigma_{i \in I_j} g_i \tag{15}$$

$$H_i = \Sigma_{i \in I_j} g h_i \tag{16}$$

Let us see in the tree structure perspective how gain score of a leaf when they are split using eq. 15 and 16, is determined.

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \lambda \tag{17}$$

Where G_L, G_R, H_L, and H_R are first order and second order gradient statistics (eq. 15 and 16) on the loss function, and λ are additional leaf regularizations. Each component in eq. 17 determines score of new left leaf, score of new right leaf, original leaf and additional leaf regularization. It is worth noting (figure 4-1) that if the gain is less than γ, it would be more beneficial not to include that branch, which corresponds to the pruning methods used in tree-based models.

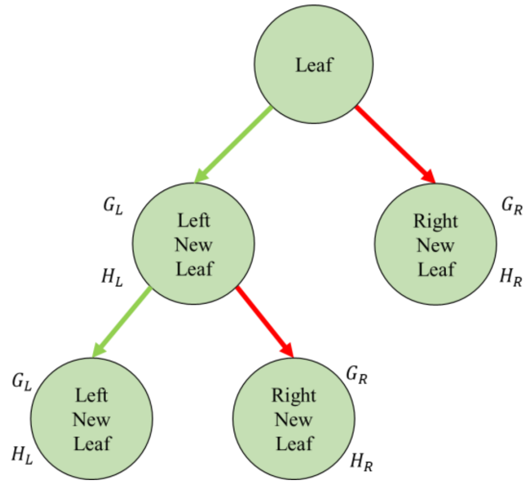

Figure 4-1: Tree Leaf Split

Basic steps of how XGBoost model works are as follows:

1. Initialize the model hyperparameters, such as the maximum tree depth, learning rate, regularization parameters, and the number of trees to be trained (detailed in section 5.5.2).
2. Calculate the initial predictions for the target variable by computing the mean value of the target variable over the training dataset.
3. Build a new decision tree to correct the errors of the previous trees, using a greedy strategy that chooses the split to maximize a gain function.
4. Apply regularization techniques such as L1 and L2 regularization and early stopping to prevent overfitting.
5. Update the predictions for the target variable by adding the predictions from the new tree to the previous predictions.
6. Evaluate the performance of the model on a holdout validation dataset, using a loss function such as mean squared error (MSE) or log loss.
7. Re-iterate through the steps 3-6 until a stopping criterion is met.
8. Return the final model as the sum of the initial predictions and the predictions from each tree in the ensemble.

The fundamental concept of XGBoost for regression prediction involves integrating numerous weak learners to generate a robust learner. Decision trees are implemented as the base learners and trained sequentially, with each tree added to the model one at a time. The weights of the training instances are adjusted so that the subsequent trees focus on the regions where the previous trees exhibited weaknesses.

XGBoost can learn complex nonlinear relationships between the input features and the target variable. XGBoost can be used with limited resources and it requires less computational time [4].

4.3 Long Short-Term Memory

A Long Short-Term Memory (LSTM) cell is a type of Recurrent Neural Network (RNN) architecture that is designed to handle long-term dependencies in sequential data. It has a memory cell (figure 4-2) that can store information for an extended period and three gates: input gate, forget gate, and output gate, which control the flow of information into and out of the memory cell.

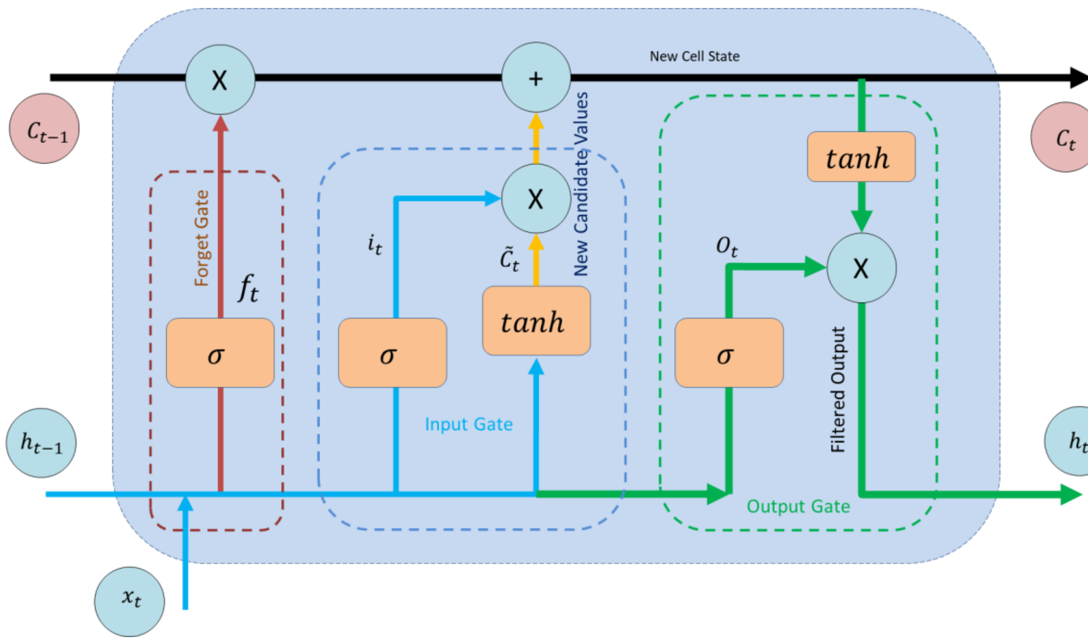

Figure 4-2: Single Standard LSTM Cell Diagram

The input gate (i_t) controls the amount of new information that is added to the memory cell, while the forget gate (f_t) determines the information that should be discarded from the memory cell. Finally, the output gate (o_t) controls the amount of information that is outputted from the memory cell to the next time step or the output layer.

The key operations of an LSTM model along with relevant mathematical equations can be summarized as below. Where x_t, C_t, and H_t, is the time-series data input quantity, cell state, and hidden state, whereas i_t, f_t, and o_t are the LSTM cell gates all with timestamp t. Convolution kernels related to internal states and gates and convolution operator are denoted by W, '*' and $'o'$ respectively.

- The *forget gate* (eq. 18) takes the previous hidden state and decides which parts of it should be discarded from the memory cell. It computes a forget vector that determines the information to be removed from the cell.

$$f_t = \sigma\big((W_{xf} * x_t) + (W_{hf} * H_{t-1}) + b_f\big) \tag{18}$$

- The *input gate* (eq. 19) takes the current input and decides which parts of it are important and should be added to the memory cell. It computes a candidate activation vector using the current input and the previous hidden state.

$$i_t = \sigma((W_{xi} * x_t) + (W_{hi} * H_{t-1}) + b_i) \tag{19}$$

- The *memory cell* (eq. 20) stores the relevant information from the previous input and the current input based on the input and forget gates (refer figure 4-2).

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} \circ x_t + W_{hc} \circ H_{t-1} + b_c) \tag{20}$$

- The *output gate* (eq. 21) determines how much of the data present in the memory cell will be utilised to produce the output. Based on the current input, the prior hidden state, and the information in the memory cell, it computes an output vector.

$$o_t = \sigma((W_{xo} * C_{t-1}) + (W_{ho} * H_{t-1}) + b_o) \tag{21}$$

- The *hidden state* (eq. 22) is the output of the LSTM cell that is passed to the next cell in the sequence. It is calculated using the output gate and the memory cell.

$$H_t = \tanh(C_t) \circ (o_t) \tag{22}$$

- *Backpropagation* is used to train the LSTM model, which entails computing the gradient of the loss function relative to the model's parameters and changing those parameters as necessary.
- The performance of the LSTM model is highly dependent on the choice of hyperparameters such as the number of LSTM cells, the learning rate, and the activation function. These hyperparameters are tuned using techniques such as grid search or random search to find the optimal combination that results in the best performance on the validation set.

There are several hyperparameters that determine the proper training of an LSTM cell. Some of the critical hyperparameters are:

- Define number of LSTM layers in the network. A network with more layers can capture more complex dependencies, but it may be more prone to overfitting.
- Number of LSTM units in each layer. A network with more units can capture more fine-grained details in the data, but it may also increase the computational cost and the risk of overfitting.
- Utilizing the activation function, nonlinearity is added to the network. LSTM cells frequently activate through sigmoid, hyperbolic tangent, and ReLU.
- Learning rate parameter determines the step size of the optimization algorithm during training. A high learning rate can cause the model to converge quickly but may lead to overshooting the optimal solution, while a low learning rate can lead to slow convergence and getting stuck in local minima.

- During training, units in the network are randomly removed using the regularization process known as Dropout. By lowering unit dependency, it can avoid overfitting.
- Batch size hyperparameter controls how many samples are handled during each iteration. While a large batch size can result in high memory usage and quicker training times, it can also produce noisy gradients and slow convergence.

Proper training of an LSTM cell involves selecting the appropriate hyperparameters and optimizing them through techniques such as grid search or random search. It also involves carefully initializing the weights and biases of the network, selecting an appropriate loss function, and monitoring the training process for signs of overfitting or underfitting.

4.4 Convolutional Neural Networks

One dimensional Convolutional Neural Networks (CNN) is a type of Deep Neural Network (DNN) that can also be used for time-series data analysis and prediction tasks, where the goal is to predict a continuous value instead of a discrete label. In a 1-D CNN layers, the final layer is a fully connected layer that outputs a single continuous value. One of the most significant advantages of CNNs is their spatial-local connectivity, which allows layers to share parameters, making them efficient learning models. It has been observed that CNNs not only provide superior performance but also exhibit dominant performance in sequential data analysis problems. The convolution layer in CNNs plays a crucial role in feature extraction, where data passing through this layer convolves with respective kernels in each layer. The convolution operation, which is essentially a dot product between the input data and kernels, generates a volume of feature maps [5].

In our case, the data is sequential and one-dimensional, each convolution layer receives a one-dimensional input data, denoted as x(n). Then, a one-dimensional kernel w(n) convolves with the input data, producing a feature map, z(n), as given by eq. 23, where l represents the size of the kernel.

$$z_n = x(n) * w(n) = \sum_{m=-l}^{l} x(m) \cdot w(n-m) \tag{23}$$

The applied 1-D CNN-based layer architecture comprises of ten layers (Section 5.5.4). It is important to understand how input data is prepared to feed into CNN model. Figure 4-3 illustrates how CNN kernel strides over the time steps to capture features information. This input data of desired time-steps and number of features are made by creating batches from segregated input dataset of defined sequence length.

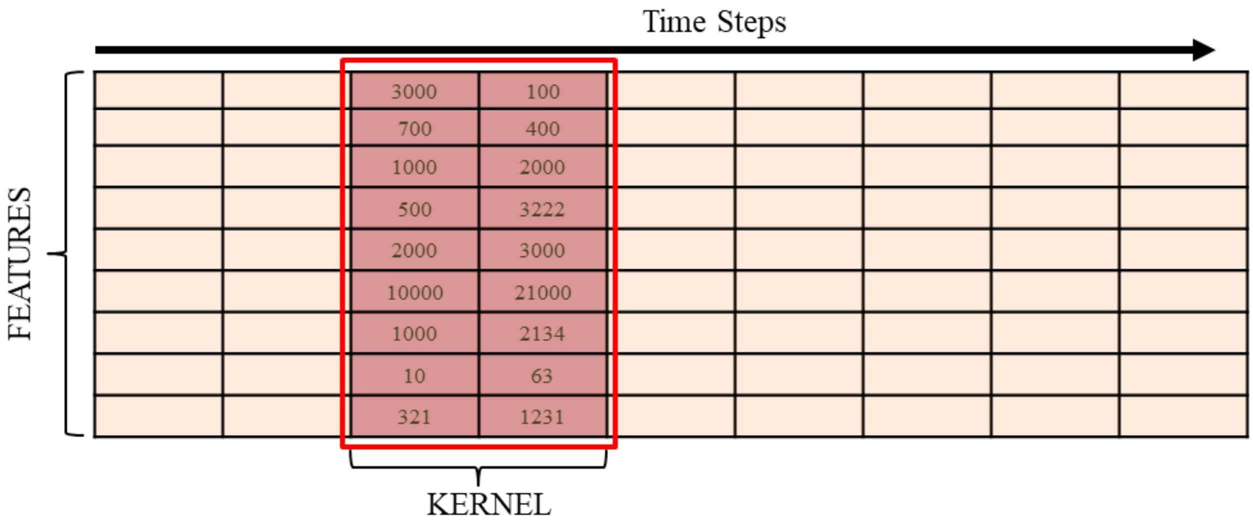

Figure 4-3: Kernel Stride over dataset features

The key hyperparameters that can affect the performance of a 1-D CNN model are:

- The *number of filters* determines the number of features that the model can learn from the input data. A higher number of filters can result in more features being learned, which can lead to better model performance. However, a high number of filters can also lead to overfitting if the model is too complex for the given dataset.
- The *kernel size* determines the width of the filter that is applied to the input data. A larger kernel size can result in a more global view of the input data, while a smaller kernel size can result in more local features being learned.
- The *stride size* determines the step size of the filter as it is applied to the input data. A larger stride size can result in faster processing, while a smaller stride size can result in more detailed information being captured.
- *Padding* is used to ensure that the output size of the convolutional layer matches the input size. There are two types of padding, 'Valid', and 'Same'. 'Valid' padding means that no padding is added whereas, 'Same' padding means that padding is added to the input data to ensure that the output size is the same as the input size.
- *Dropout rate* is a regularization technique that randomly drops out a certain percentage of nodes in the model during training to prevent overfitting. A higher dropout rate can result in better generalization performance, while a lower dropout rate can result in better training performance.
- The *learning rate* determines the step size of the gradient descent algorithm during training. A higher learning rate can result in faster convergence, while a lower learning rate can result in better

convergence. However, if the learning rate is too high, the algorithm may overshoot the optimal solution and fail to converge.

It is important to note that the appropriate hyperparameters can depend on the specific characteristics of the input data and the model architecture. Therefore, it is often necessary to perform hyperparameter tuning to find the optimal hyperparameters for a given task (section 5.5.4).

Chapter 5

5 Design of Experiments

The way the project objectives (Section 1.4) are to be achieved, depends on approach followed for conducting experiments, and how feasible its design is to accommodate the change in settings of the hyper-parameters required by respective applied AI models. To incorporate all above factors rigorously and systematically, methodological strategy is followed such as design of experiments (DOE).

5.1 Experiment Design Flow

Intent is to highlight the design flow with which experiments are to be performed. As per figure 5-1, this section shall focus on experiments conducted on training and testing model with PMSM dataset [6]. While after experimentation with PMSM datasets, tests shall be performed using target vehicle datasets using the baselined optimal hyper-parameters as detailed in section 5.4.

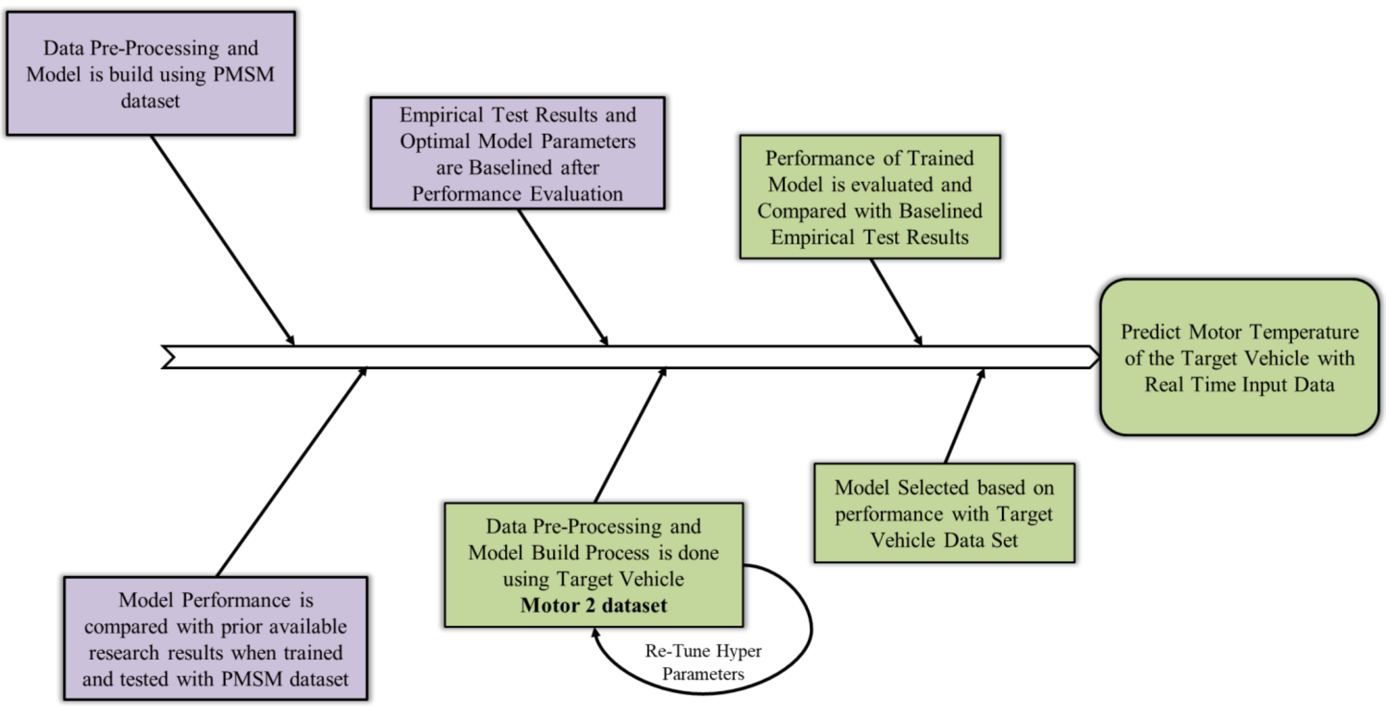

Figure 5-1:Experiment Design Flow with Data Sets

5.2 Problem Definition and Goal

Detailed system design and its execution steps at each stage are discussed in section 3. Considering the system design, end result is to build an AI model to be capable of estimating the motor temperature at real time. To build such model, initially in literature study (Section 2.1) different research papers [3], [4], [1], [5], whose goal is similar to our objectives of this thesis work are studied and their evaluation results are taken into account to validate the end result for comparison purpose.

- Target vehicle datasets were not available from the lab, during the early stages of project.
- *Design decision* is taken to run experimental tests on data sets [10], [6] as specified in section 6.1.1, which were used in research papers [3], [4], [1], [5].
- The data set contains information from sensors placed on a type of motor called a permanent magnet synchronous motor (PMSM). The motor was tested on a machine, and the data was recorded by the LEA department at Paderborn University [6].
- PMSM data set is used for experimentation purpose and model hyper parameters are tuned using the same.
- Final goal of this experimentation is to build a model with tuned hyper-parameters.
- After which when target vehicle data sets are applied, close to similar results should be obtained by using the same system AI pipeline framework.
- Estimation of temperature for target Motor 2 is done using dataset prepared from input sensor data from target vehicle (As detailed in section 5.2.2).
- Details of experimentation flow is illustrated in figure 5-1, which clearly gives information to the reader that baselining of hyper parameters and results were done using PMSM data. This baseline after performing test experiments helps the target vehicle to have reliable evaluation method.

5.3 Experiment Setup

Target vehicle is attached with an implement, and that load is attached to the PTO as seen in figure 3.1. The PTO is powered by our target vehicle Motor 2. This load attached to the PTO is controlled by Vehicular Control Unit (VCU) which drives the implement or load at different speed i.e., 350/450/540/650 rpm. Vehicle logs for these implement controls are recorded separately for a period of time to observe its behavior. Process followed to capture these data logs are explained in detail in section 3.3.1.

The PMSM motor is connected to the test bench setup where tests carried out by the LEA department at Paderborn University in the context of collecting sensors data from PMSM setup [10]. Datasets are described in further depth in section 5.3.2.

5.3.1 Computation Platform

Scripts of this thesis work to execute the different models were executed on local laptop machine. Online platform (Google Colab [19]) was used for the computation of Deep Neural Network (DNN) based algorithms and various occasions wherever execution time was high.

Table 5-1: Computing Machine Specifications

Specifics	Configuration
Processor	12th Gen Intel(R) Core (TM) i7-12700H, 2300 Mhz, 14 Core(s), 20 Logical Processor(s)
Google Colab	GPU resources

5.3.2 Dataset Used in Experiments

This thesis work uses the dataset which is readily available and is used to meet similar objectives (Section 1.2) i.e., had the data attributes which could be used for the estimation of motor temperature.

Table 5-2: Dataset Samples Size

Data Set	Data Set Length
PMSM	998070
Target Motor 2	30276

5.3.2.1 Dataset from PMSM

The dataset contains sensor data gathered from a test bench where a PMSM was deployed. The measurements from the test bench were obtained by the LEA department at Paderborn University. The dataset is partially anonymized [10].

PMSM data sets were used for the study and to create baseline after training a model and fine-tunning its hyper parameters.

Table 5-3:Attributes of the PMSM Data Set

ATTRIBUTES	DESCRIPTION
profile_id	Measurement session id. Each distinct measurement session can be identified through this integer id.
ambient	ambient temperature (°C)
motor_speed	Motor speed (rpm)
torque	Motor torque (Nm)
stator_winding	Stator winding temperature (°C) measured with thermocouples
stator_yoke	Stator yoke temperature (°C) measured with thermocouples
stator_tooth	Stator tooth temperature (°C) measured with thermocouples
i_q	Current q-component measurement in dq-coordinates
u_q	Voltage q-component measurement in dq-coordinates (V)
coolant	Coolant temperature (°C)
pm	Permanent magnet temperature (°C) measured with thermocouples and transmitted wirelessly via a thermography unit.
u_d	Voltage d-component measurement in dq-coordinates
i_d	Current d-component

5.3.2.2 Dataset from Target Vehicle

The dataset contains the sensor data from the target motor controller ECU. Raw data of the target Motor 2 is received over CAN bus and recorded.

Table 5-4: Identified attributes of the target vehicle

ATTRIBUTES	DESCRIPTION
Motor Temperature	Implement motor temperature read from motor temperature signal (°C)
Motor Speed	Implement Motor speed in rpm
Motor Torque	Implement Motor Torque in Nm
Phase current	Phase current (Amps) from the motor controller
DC Bus Voltage	Voltage recorded from the DC bus (Volts)
DC Bus Current	Current recorded from the DC bus (Amps)
Max Drive Torque	Maximum drive torque of the motor (Nm)
Max Braking Torque	Maximum drive braking of the motor (Nm)
Motor Ctrl Temp	Hydraulic Inverter Temperature (°C)

5.4 Experimentation Analysis

The main aim of this portion is to inspect the outcomes of experiments using approaches for data pre-processing (section 3.3.2) and model building (section 3.3.3) using datasets that are currently accessible. Perform experiments as highlighted in section 5.1. Detail the discussion points of the results of model and its explanation w.r.t target parameters. Idea is to understand the data and its features to identify the appropriate techniques to extract the correct information from it, while ignoring the faulty data or incorrect information. Exploratory data analysis (EDA) is an ideal approach for illustrating the unique and distinctive features of the data and the valuable insights that are included within it [4]. Datasets are processed and inspected based on different stages specified in figure 3-4. The points from data cleaning and transformation (section 3.3.2.2 and 3.3.2.3) which are to be discussed in subsequent sections for accessible datasets (section 5.3.2) are listed below.

5.4.1 Analysis Methodology Overview

Methodological approach followed for experimentation in the sequential manner is briefly described in the following subsections. Exploratory data analysis is done followed by feature selection. Detailed subsections of data pre-processing are broken down to specific methods and are explained in sequential order.

5.4.1.1 Data Samples Distribution

Various ways are used to start the analysis of dataset. The histogram plot is used for the visualization to understand different aspects of the dataset. For instance, it can give the brief idea about the distribution of entire dataset and each variable. We can sort the test run with maximum samples to further analyze and filter the target variable.

Analyzing the first n number of rows of samples in the dataset, can help to visually identify patterns or trends in the data, particularly if there are many columns or rows to consider.

Statistical summary of the attributes of the dataset can help to identify key features of the dataset and for making decisions about its preprocessing and its analysis approach. The summary is sorted based on the standard deviation of the columns. It shows the descriptive statistics of each attribute, including count, mean, standard deviation, minimum, 25th percentile, 50th percentile (median), 75th percentile, and maximum.

Overall, above approaches aim to capture key features information along with getting clear idea of the dataset, we are dealing with. These methods shall help in taking further decisions.

5.4.1.2 Redundant & Missing Data Handling

When there are duplicate or null values in a dataset, it can affect the accuracy and reliability of any analysis or model built on it. Duplicate values can cause problems as they may result in incorrect counts, statistical analysis, and visualizations. On the other hand, null values (also known as missing values) can cause errors in calculations, as well as skew results, if not handled properly.

To deal with duplicate values, one can remove them from the dataset, or combine them if necessary. However, it's important to make sure that removing or combining the duplicates doesn't result in a loss of important information or data. In some cases, duplicates might be intentional and represent multiple occurrences of the same event or entity. There are various methods to handle missing values in a dataset, such as:

- Deleting rows with missing values: This method is useful when there are only a few missing values in the dataset, and removing them doesn't significantly affect the analysis or model.
- Imputing missing values: This involves filling in the missing values with estimated values based on the available data. The estimated values could be the forward fill, backward fill, mean, median, or mode of the available data, or they could be predicted using machine learning (ML) models.
- Treating missing values as a separate category: In some cases, missing values might represent a separate category or feature in the dataset. This method can be useful when the missing values are significant and cannot be ignored.

Experiment is conducted to check whether there are any null or duplicate values in the dataset. Number of null values are calculated in each column of the dataset. Which provides information on which columns have the most missing data. Heatmap plot is used to visualize the null values in the dataset. The plot is used to display missing values as yellow cells, and non-missing values as purple cells. This plot helps to identify which columns have the most missing data and the distribution of the missing values throughout the dataset.

5.4.1.3 Outliers' Detection

Outlier detection in a dataset is an important step in data preprocessing and analysis. Outliers are data points that deviate significantly from other observations in the dataset, and their presence can have a significant impact on statistical analysis, modeling, and decision making. Outliers can arise due to various reasons such as measurement errors, data entry errors, and true anomalies in the data. Outliers do-not affect median values, which makes them useful [4].

Identifying and handling outliers is important because they can skew the results of statistical analysis, leading to incorrect conclusions or biased models. Outliers can also affect the accuracy of machine learning (ML) models, leading to overfitting or underfitting. Therefore, it is important to detect and handle outliers in a dataset before performing any analysis or modeling.

From the boxplot, we can infer several pieces of information about the distribution of each numerical column. The box in the plot represents the interquartile range (IQR), which constitutes of middle 50% of the data in the range between the 25th (Lower Quartile) and 75th (Upper Quartile) percentile. The line inside the box represents the median of the data. The whiskers of the plot represent the range of the data outside the IQR, up to a maximum of 1.5 times the IQR. Any points outside this range are considered outliers and are plotted as individual points.

Using the boxplot, we can see the range of values for each column, as well as any potential outliers. We can also compare the distributions of different columns and identify any columns with similar distributions. This can help us identify potential relationships between variables which in turn helps in the data cleaning and modeling processes.

5.4.1.4 Variable Distribution

Distribution of dataset can be graphically represented through usage of density plots. They show the density of data points along the y-axis, with the x-axis representing the range of values in the data set. The density plot can help in identifying the shape of the distribution, such as whether it is unimodal or bimodal, as well as the location and spread of the data. Density plots can provide valuable insights into the distribution and structure of the data, which can help in understanding the data and making informed decisions.

5.4.1.5 Variable Correlation

Attributes data analysis is done using different visualizations methods such as heat map and attributes pair plot matrix (figure 5-6 and 5-7) of the given dataset. It helps in identifying the pairs of variables that have high correlation coefficients. Pearson correlation coefficient is used to measure the linear relationship between the variables. Lists of variables are identified by filtering the indices of correlation matrix with threshold. By this method highly correlated variables are identified.

The average absolute correlation coefficient (*AvgAbsCorrCoef*) between all pairs of variables in the dataset helps to understand the degree of correlation among the variables. Whereas the *AvgAbsCorrCoef* for each attributes helps to understand which attributes are highly correlated with other variables attributes. This way such correlation information can be useful in feature selection (section 3.3.2.4). It also helps in finding multicollinearity in regression models.

Scatter Plots (section 5.6.2.6) is another pictorial method used to deep dive into getting insights of the relationships between variables. These plots show how one variable changes in relation to another. Positive correlations are indicated by a trend upward from left to right. While negative correlations are indicated by a downward trend. No correlation is indicated by a lack of trend or scattered dots. Pairwise relationships between all the variables in the dataset is plotted.

To infer from the scatter plot, we can examine the patterns and trends that emerge from the plot, including any linear or nonlinear relationships, clusters or outliers. These insights can help identify potential predictors for the dependent variable and highlight areas where further exploration or analysis may be needed.

5.4.1.6 Data Scaling Techniques

Normalization techniques are used to transform the values of numerical features to a common scale, which helps to improve the performance and accuracy of the applied models. There are several benefits of applying normalization technique:

- It improves accuracy bringing all the features to the same scale, which avoids giving undue importance to any particular feature. This leads to more accurate predictions.
- It helps in faster convergence of the gradient descent algorithm, which is used to optimize the model parameters. This is because it allows the algorithm to take larger steps towards the global minimum, which leads to faster convergence.
- Normalization techniques such as Min-Max scaling and Z-score normalization help in reducing the impact of outliers on the model performance. This is because they bring the values of the features within a smaller range, which makes them less sensitive to outliers.
- It also helps in better interpretability of the model parameters. This is because the coefficients associated with normalized features represent the relative importance of each feature in predicting the target variable.

5.4.1.7 Input and Target Data Preparation

For model to train with datasets, it must be prepared such that it is ready to be fed to build model stage. Here, mainly data is split into desired training and test or validation set (as discussed in section 3.3.2.5), and processed into desired shape as per the given model.

5.4.2 Analysis using PMSM Dataset

In this section experiments analysis is done using PMSM datasets (dataset info detailed in section 5.5.1). This dataset is analyzed and transformed to prepare the data, ready to be fed for model training. Observations of the experimental tests are described below, which are followed based on the sequential methodological approach detailed in section 5.4.1.

5.4.2.1 Data Samples Analysis

The histogram plot for the entire dataset shows the distribution of each variable across all test runs identified by the 'profile_id' column. Dataset is visualized to understand the sample size of different test runs in the entire dataset.

- Histogram as illustrated in the figure 5-2 is the visual representation of distribution of data size.
- It highlights the size limit of each test run.
- Test run with 'profile_id', 20 has the maximum number of samples recorded. Distribution of data for this test run in histogram (figure 5-2) shall help to identify and choose one of the predictor dependent target variables or variable out of multiple redundant variables as observed later in section 5.4.2.4.
- We can filter specific samples based on size, which could be later used for the validation of the model.

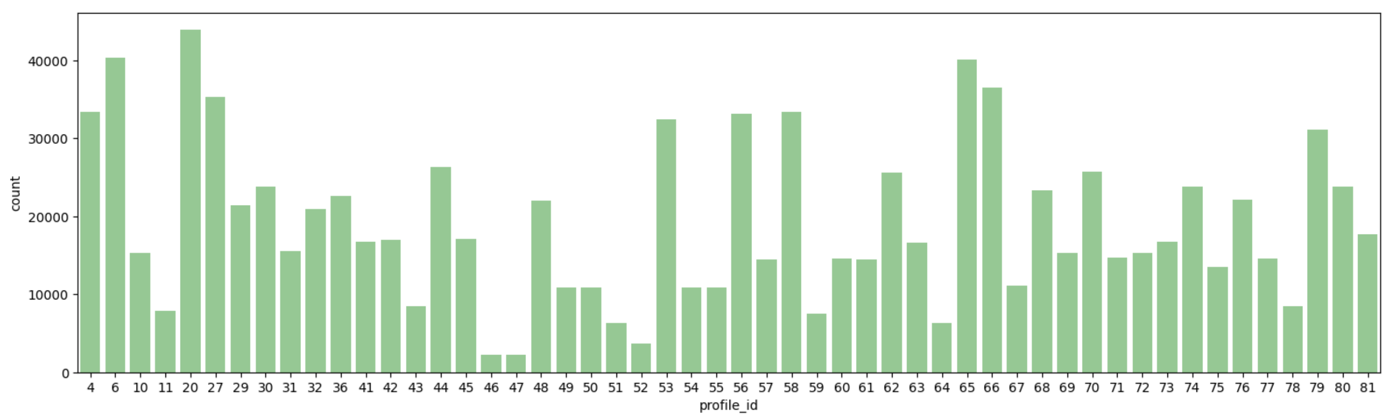

Figure 5-2: Dataset Histogram

- The first n number of rows of samples in the dataset are as listed in table 5-5. It helps us to observe the dataset for any patterns or trends in the data, particularly when there are many columns or rows to consider.
- Visualize and understand the range of values and any anomalies are present in the given list of the dataset for all attributes.

- We can see that in this dataset, most of the values lie between -2 and 2, with only a few outliers with values below -2 or above 2.
- The lack of references to the units used for each of the samples in the dataset description makes it more difficult to understand the values measured.

Table 5-5: Dataset Samples List

	ambient	coolant	u_d	u_q	motor_speed	torque	i_d	i_q	pm	stator_yoke	stator_tooth	stator_winding	profile_id
0	-0.752143	-1.118446	0.327935	-1.297858	-1.222428	-0.250182	1.029572	-0.245860	-2.522071	-1.831422	-2.066143	-2.018033	4
1	-0.771263	-1.117021	0.329665	-1.297686	-1.222429	-0.249133	1.029509	-0.245832	-2.522418	-1.830969	-2.064859	-2.017631	4
2	-0.782892	-1.116681	0.332771	-1.301822	-1.222428	-0.249431	1.029448	-0.245818	-2.522673	-1.830400	-2.064073	-2.017343	4
3	-0.780935	-1.116764	0.333700	-1.301852	-1.222430	-0.248636	1.032845	-0.246955	-2.521639	-1.830333	-2.063137	-2.017632	4
4	-0.774043	-1.116775	0.335206	-1.303118	-1.222429	-0.248701	1.031807	-0.246610	-2.521900	-1.830498	-2.062795	-2.018145	4

The sorted statistical summary based on the standard deviation of each attribute in the dataset can be seen in the table 5-6. This table can be useful for quickly identifying key features of the dataset and for making decisions about how to preprocess and analyze the data.

- The descriptive statistics features including count, mean, standard deviation, minimum, 25th percentile, 50th percentile (median), 75th percentile, and maximum, of each attribute is shown in the table 5-6.
- The attributes ‘ambient’ and ‘torque’ have the largest standard deviations of 0.99 and 1.00 respectively, indicating that their values are widely dispersed from the mean.
- The ‘ambient’ attribute has the lowest minimum value of -8.57, which is considerably lower than the other attributes. This suggests that there are outliers or errors in the data for this attribute.
- Similarly, the ‘torque’ attribute has the highest maximum value of 3.02, which is significantly higher than the other attributes. This means that there are outliers or errors in the data for this attribute.
- We can see that the mean of each variable is close to zero, indicating that the data is centered around zero. The standard deviation of each variable is close to one, indicating that the data is spread out. Overall, it is certain that the variables are normalized or standardized in the dataset.
- The minimum and maximum values for each variable vary across a wide range, which suggests that the variables have different scales and ranges.

Therefore, to address the abnormalities or outliers in the ‘ambient’ and ‘torque’ attributes, it may be necessary to investigate the data further and remove any erroneous or outlier data.

Table 5-6: Statistical Analysis of Data Attributes

	count	mean	std	min	25%	50%	75%	max
coolant	998070.00	0.00	1.00	-1.43	-1.04	-0.18	0.65	2.65
u_d	998070.00	0.00	1.00	-1.66	-0.83	0.27	0.36	2.27
u_q	998070.00	-0.01	1.00	-1.86	-0.93	-0.10	0.85	1.79
motor_speed	998070.00	-0.01	1.00	-1.37	-0.95	-0.14	0.85	2.02
torque	998070.00	-0.00	1.00	-3.35	-0.27	-0.19	0.55	3.02
i_d	998070.00	0.01	1.00	-3.25	-0.76	0.21	1.01	1.06
i_q	998070.00	-0.00	1.00	-3.34	-0.26	-0.19	0.50	2.91
pm	998070.00	-0.00	1.00	-2.63	-0.67	0.09	0.68	2.92
stator_yoke	998070.00	0.00	1.00	-1.83	-0.75	-0.06	0.70	2.45
stator_tooth	998070.00	-0.00	1.00	-2.07	-0.76	0.01	0.77	2.33
stator_winding	998070.00	-0.00	1.00	-2.02	-0.73	0.01	0.73	2.65
ambient	998070.00	-0.00	0.99	-8.57	-0.60	0.27	0.69	2.97

5.4.2.2 Redundant & Missing Data Samples

PMSM Dataset is checked for any duplicate samples and null values. Strategies discussed in section 5.4.1.2 can be used to fix the issues related to redundancy and missing data samples. Incorporating proper strategy based on dataset prevents the presence of outliers in the data samples.

- PMSM dataset does not contain any duplicate data samples across the entire dataset.
- The heatmap plot (figure 5-3) displays that no yellow samples were observed. It denotes that it is a clean plot with no missing values or null values.

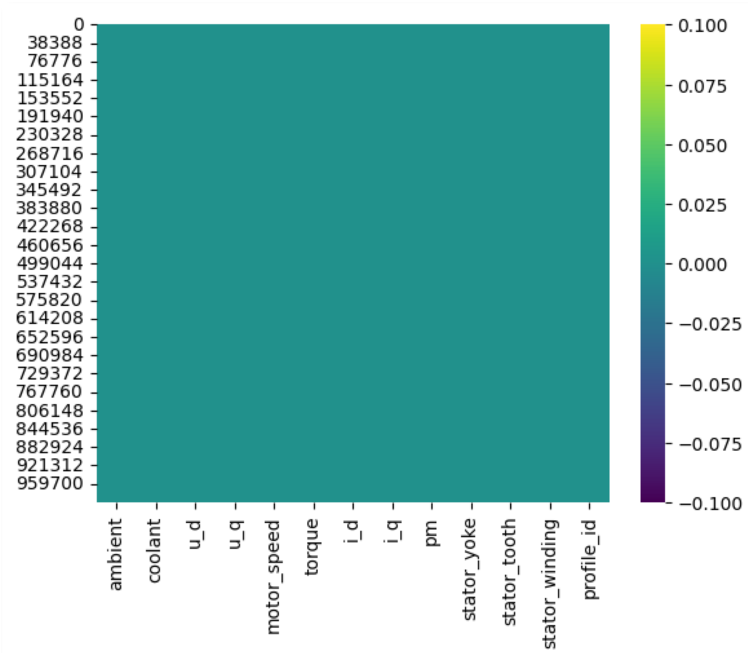

Figure 5-3: Missing Values Visualization through Heat Map

5.4.2.3 Outliers' Detection Analysis

Boxplot is a graphical representation (as represented in figure 5-4) of the distribution of a dataset that visually displays the range, median, quartiles, and outliers of the data. Outliers in the data are detected using boxplots.

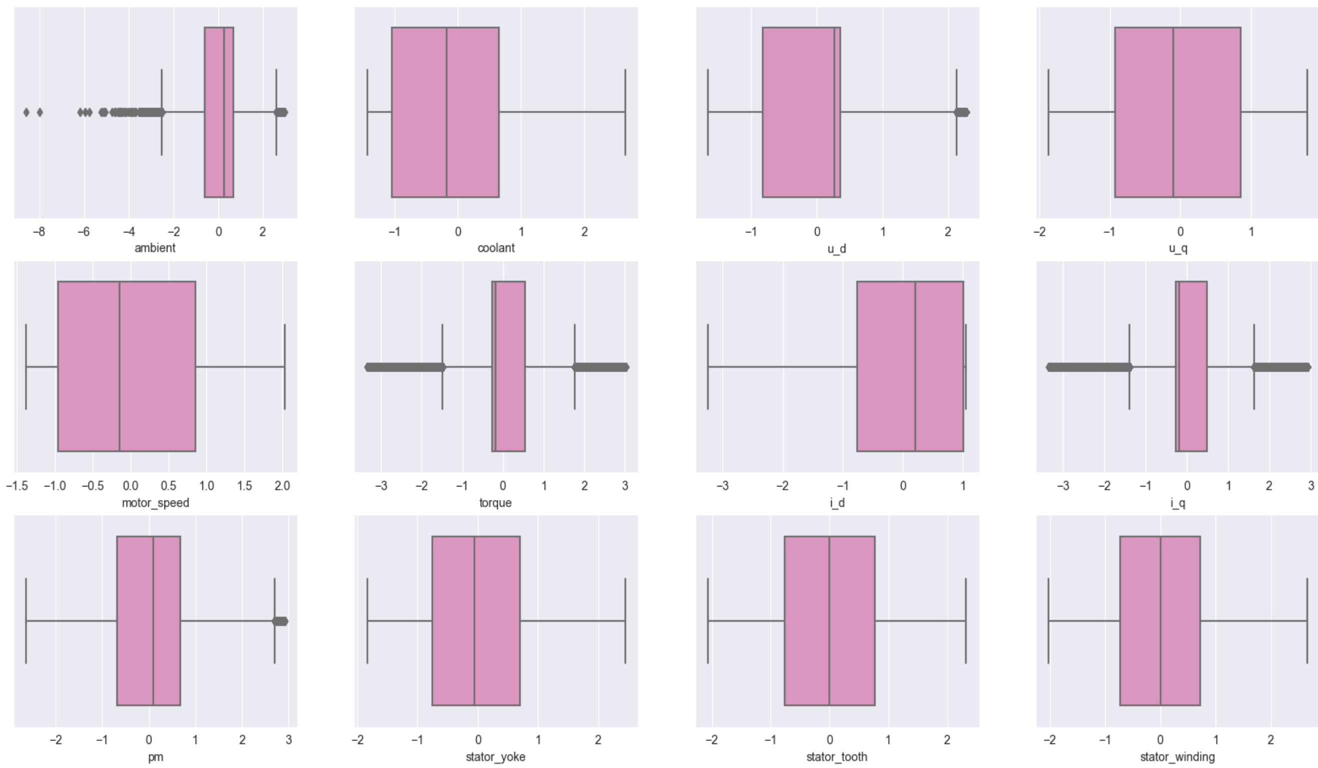

Figure 5-4: Box-Plots Representation

- We can see whiskers for the attributes 'ambient', 'pm', 'u_d', 'i_q' and 'torque'.
- Most of the attributes show equal variance along the median line such as 'coolant', 'u_q', 'motor_speed', 'i_d', 'stator_winding', 'stator_tooth' and 'stator_yoke'.

5.4.2.4 Variable Distribution

Density plots shows the distribution of variables as displayed in figure 5-5 for all attributes of dataset.

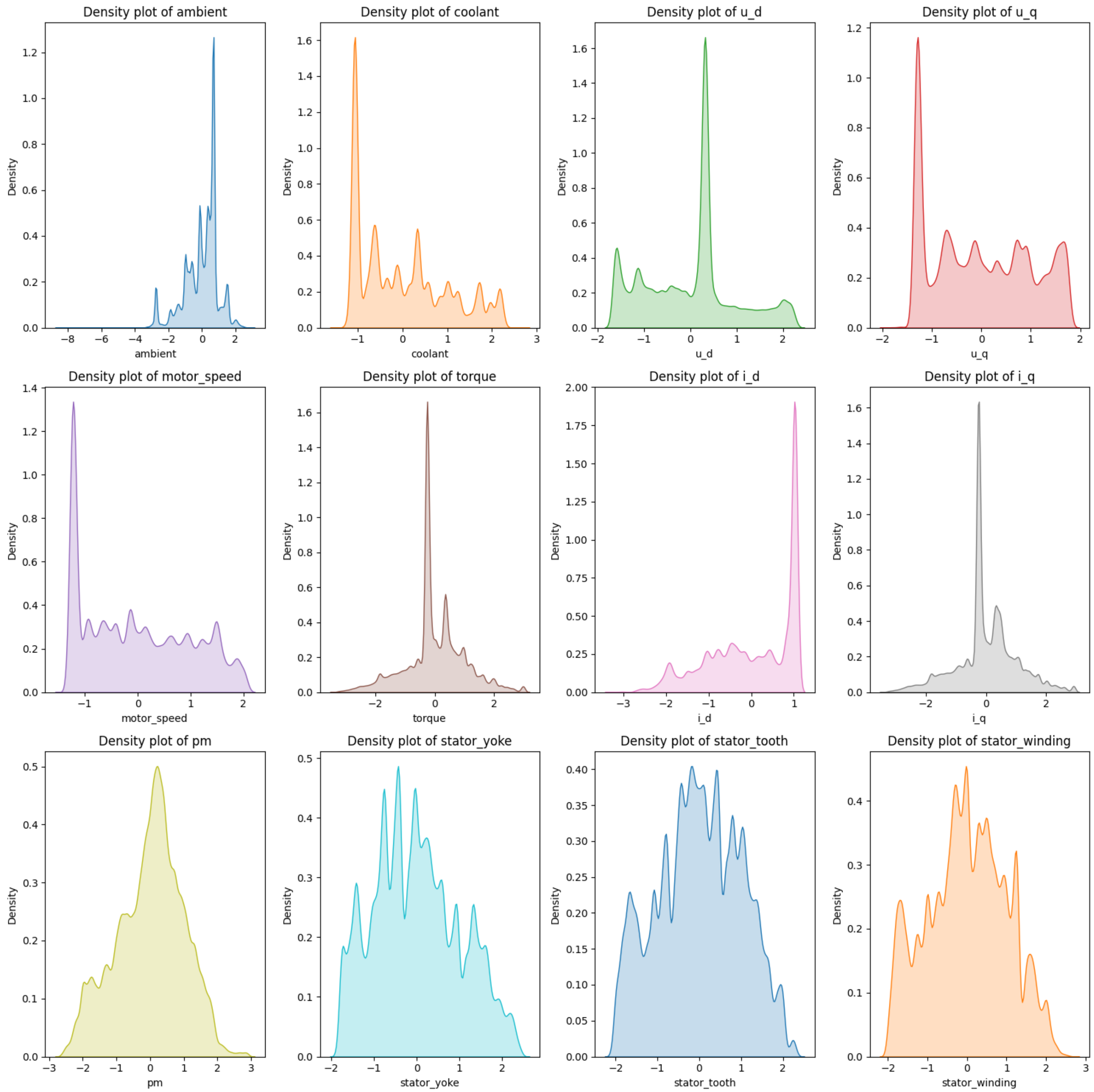

Figure 5-5: Data Distribution Density Plots

Looking at the histograms following observations are found:

- We can see that some variables, such as ‘coolant’, ‘motor_speed’, and ‘u_q’ skewed on the positive side because long tail is in the positive direction. In case of positive skews mean is larger than the median. There will be some large values that pull the mean towards the right causing the skewedness.
- Whereas ‘ambient’ and ‘i_d’ attributes are negatively skewed. That means skewed on the negative side, i.e., long tail is in negative direction. Here median is larger than the mean. There will be some large values that pull the mean towards the left causing the skewedness.
- Attributes ‘stator_yoke’, ‘stator_tooth’, and ‘stator_winding’ have a relatively similar and normal distribution, with most of the data clustered around the mean.
- The histogram plot also shows the distribution of each variable for each test run, which are identified by the "profile_id" variable. We can see that the distribution of some variables, such as ‘coolant’, ‘u_q’ and "motor_speed," varies depending on the test run, indicating that these variables may be affected by the conditions of the test.

Overall, the density plot gives us a visual representation of the distribution of each variable and how they vary across the different test runs. It gives us an idea of how the variables may be related to each other.

5.4.2.5 Variable Correlation Analysis

Attributes correlation coefficients are highlighted in the heatmap illustrated as in the figure 5-6.

- The correlation coefficients are calculated between all pairs of variables in the dataset and it identifies the pairs of variables that have a correlation coefficient greater than 0.8 in absolute value.
  [('coolant', 'stator_yoke'), ('u_d', 'torque'), ('torque', 'i_q'), ('stator_yoke', 'stator_tooth'), ('stator_yoke', 'stator_winding'), ('stator_tooth', 'stator_winding')]
- The absolute average correlation coefficient (*AbsAvgCorrCoef*) between *all pairs of variables* in the dataset is 0.311, which suggests that there is a moderate degree of correlation among the variables.
- The *AbsAvgCorrCoef* for each attribute of the dataset are given below respectively:
  [(ambient: 0.331), (coolant: 0.388), (u_d: 0.329), (u_q: 0.216), (motor_speed: 0.338), (torque: 0.319), (i_d: 0.352), (i_q: 0.312), (pm: 0.425), (stator_yoke: 0.460), (stator_tooth: 0.479), (stator_winding: 0.475)]
- For instance, ‘stator_yoke’, ‘stator_tooth’, and ‘stator_winding’ have high *AbsAvgCorrCoef*, which implies that they are highly correlated with other variables in the dataset. Two or more variables with high *AbsAvgCorrCoef* values, may indicate that they are measuring the same underlying concept and could potentially be combined or one of them may need to be removed. On this basis, we consider ‘stator_winding’ temperature as our target variable and ignore others since ‘stator_yoke’ and ‘stator_tooth’ attributes are redundant.
- On the other hand, ‘u_q’ has a low absolute average correlation coefficient, indicating that it is weakly correlated with other attributes in the dataset. Low value may indicate that it is not related to the other variables and may not be useful for analysis.

These findings are valuable for understanding the relationships between variables and take actions based on this information. We are using this inference from *AbsAvgCorrCoef* values, in the selection of the attributes.

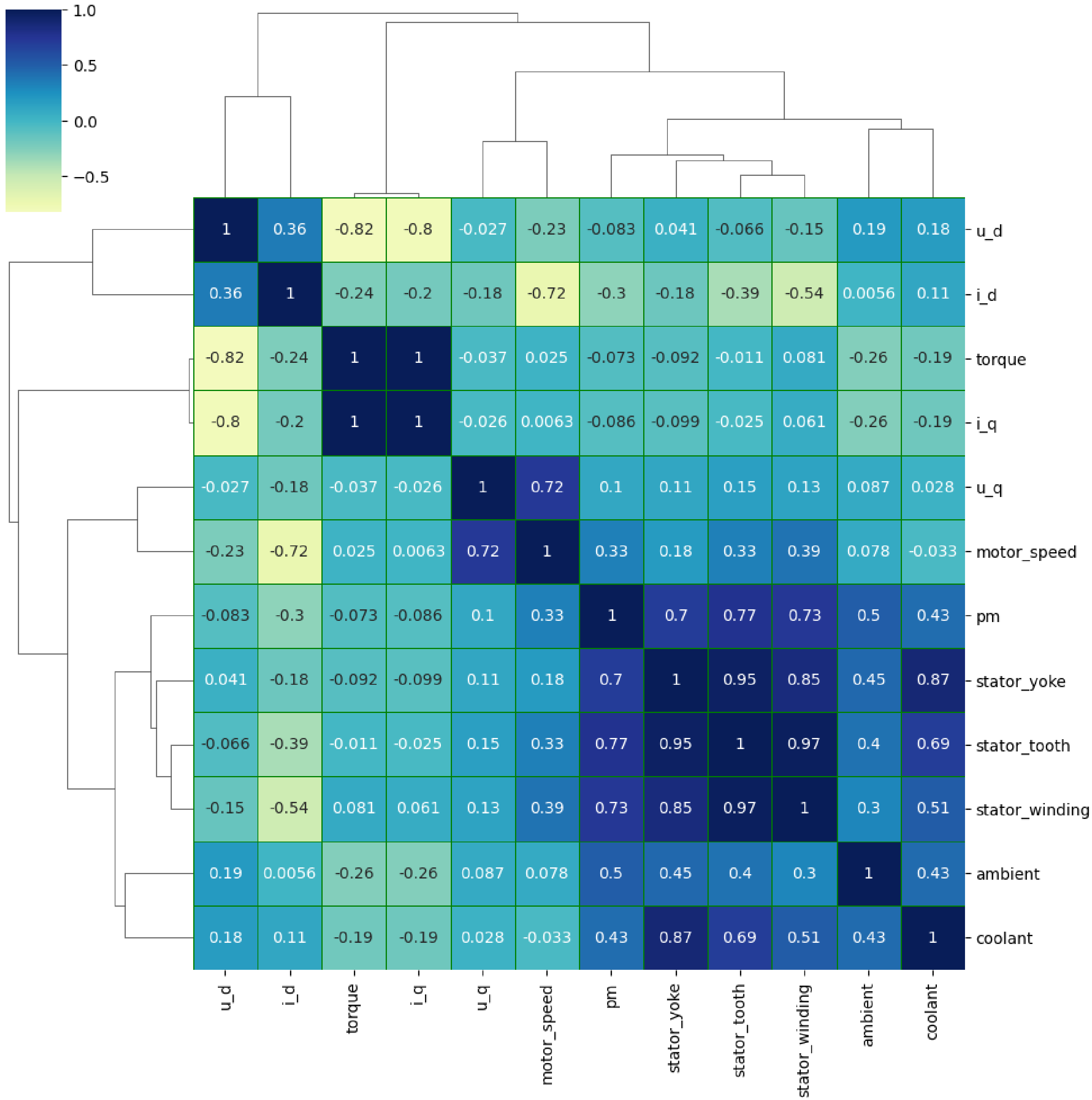

Figure 5-6: Data Correlation Matrix

Pairwise relationships between all the variables in the dataset is plotted using scatterplots. The diagonal plots show distribution of each variable.

- Variables 'stator_winding', 'stator_yoke' and 'stator_tooth', can be seen to have the direct strong linear relationship with each other. Hence, Predictor variables for anyone of them shall help to get the output for another variable. That’s the reason, ‘stator_winding’ is further analyzed, assuming based on above inferences, that they are linearly related.

- Similarly, ‘i_q’ and ‘torque’ are linearly related.
- Curvilinear relationship can be observed between ‘i_d’ and ‘i_q’.

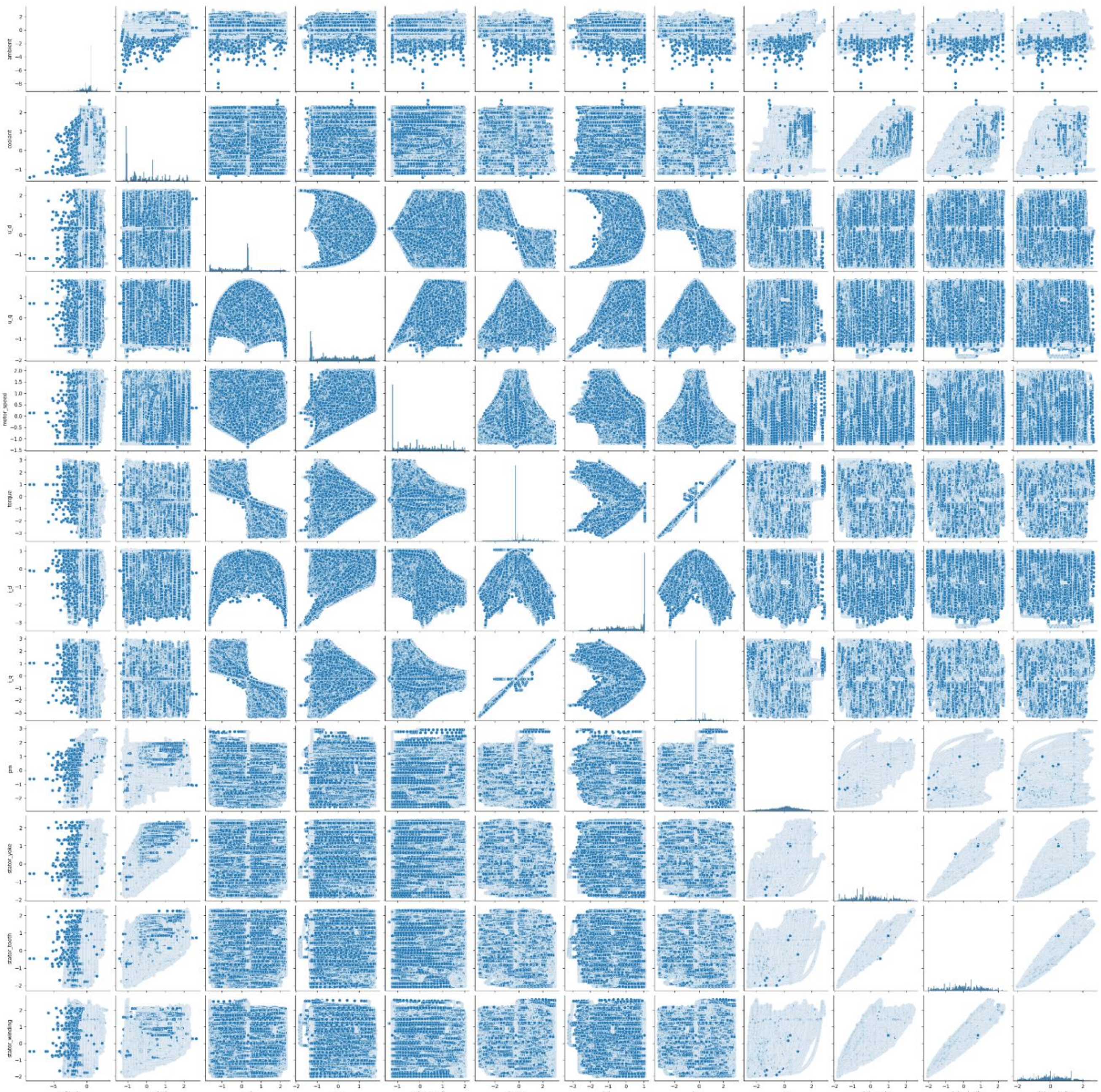

Figure 5-7: Attribute Pairs Plot Diagram

5.4.2.6 Feature Selection

After performing data pre-processing, analysis is done based on data distribution, outliers and variable correlation of attributes on the dataset.

- Inferences from boxplots, density plots and heatmap correlation matrix suggest the strong linear relationship between the stator temperatures i.e., ‘stator_winding’, ‘stator_tooth’ and ‘stator_yoke’ attributes.
- In order to gain a deeper understanding of the correlation among the three stator temperatures, we evaluate the plots (figure 5-8) representing the feature values for a selection of randomly chosen test runs.
- The subplots validate that the three temperature features have a similar pattern. Amongst the three, the stator winding temperature exhibits the highest variability, followed by the stator tooth and stator yoke temperatures. This disparity becomes prominent when there is a considerable fluctuation in the stator winding temperature.

Figure 5-8: Stator Temperatures Analysis across test runs

- It can be inferred from the observed pattern of stator temperatures, that they are influenced in the same manner based on behavior of the predictor variables.

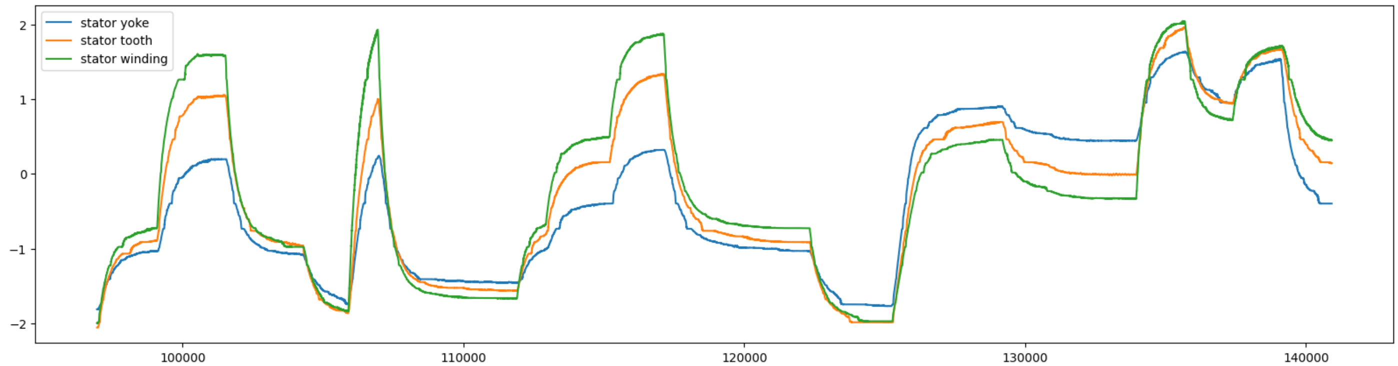

Figure 5-9: Stator Temperatures Spread for a test run

- Hence, lets recheck the selected target feature i.e., 'stator_winding' distribution over entire dataset (as in figure 5-9). Data is symmetrically distributed.

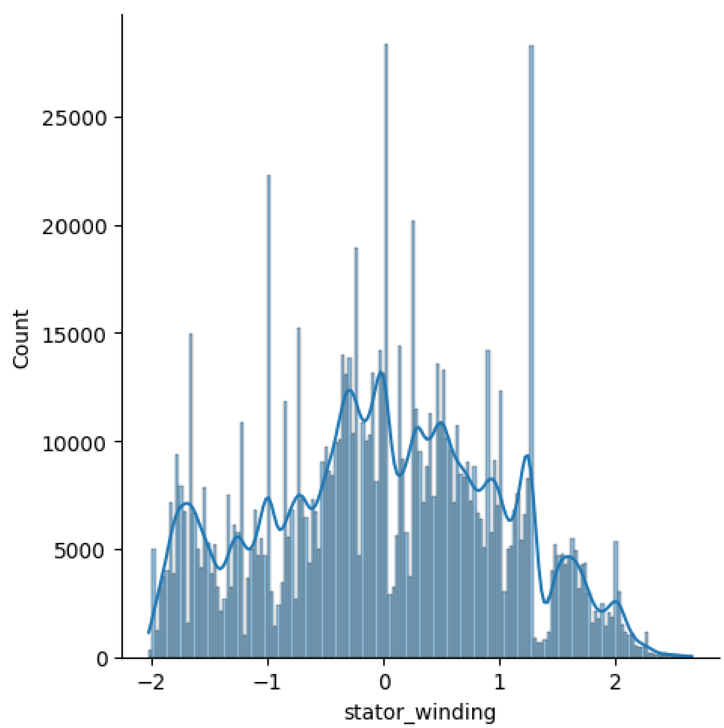

Figure 5-10: Stator Winding Data Distribution Plot

- We analyze the target feature i.e., stator winding vs input features such as torque and motor speed (as in figure 5-11).

- Relation can be observed between the stator winding, torque and motor speed. When motor speed or torque is varied, the change in stator winding temperature is observed.

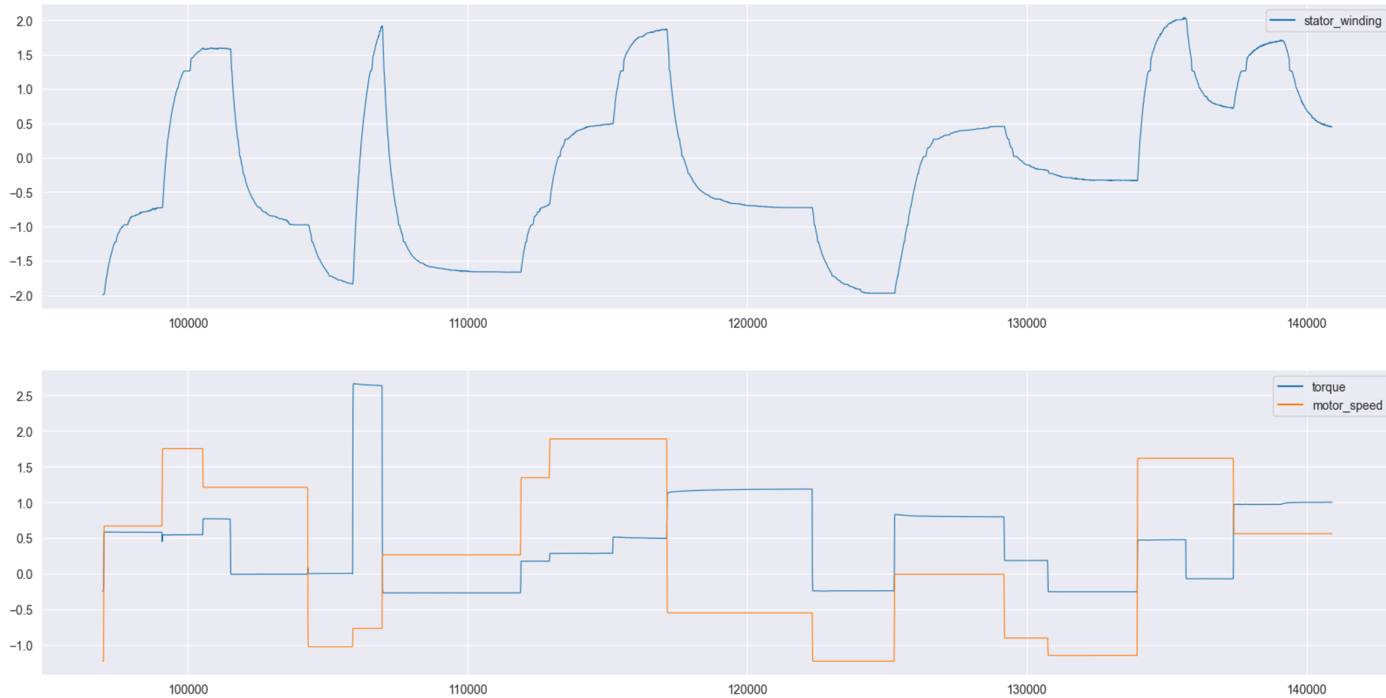

Figure 5-11: Stator Winding vs Motor Speed or Torque Plot

Once, input and target features are identified, dataset is required to be split into train and test data for the model training and validation purpose. Input data shape for model training is different for different model.

5.4.3 Analysis using Target Vehicle Dataset

In the prior stage the experiments were conducted to perform data preprocessing and feature selection using PMSM dataset (section 5.3.2.1). Inferences, results and model performances are baselined after they are compared with available research study. Baselined hyper-parameters (section 5.5) and target models (section 4) are then again pre-processed and trained with target vehicle i.e., off-road vehicle dataset.

In this section experiments analysis is done using target vehicle i.e., Off-Road vehicle datasets (dataset info detailed in section 5.3.2.2). This dataset is analyzed and transformed to prepare the data, ready to be fed for model training. Observations of the experimental tests are described below, which are followed based on the sequential methodological approach detailed in section 5.4.1.

5.4.3.1 Data Samples Analysis

The histogram plot for the complete dataset displays the distribution of each variable over all test runs specified by the load profile column. The full dataset is visualized to help comprehend the sample size of the various test runs.

- Histogram as illustrated in the figure 5-12 is the visual representation of distribution of data size.
- It highlights the size limit of each load profile run.
- Test run with load profile, 450 has the maximum number of samples recorded. Distribution of data for this load profile in histogram shall help to identify and choose one of the predictor dependent target variables or variable out of multiple redundant variables as observed later in section 5.6.3.6.
- We can filter specific samples based on size, which could be later used for the validation of the model.

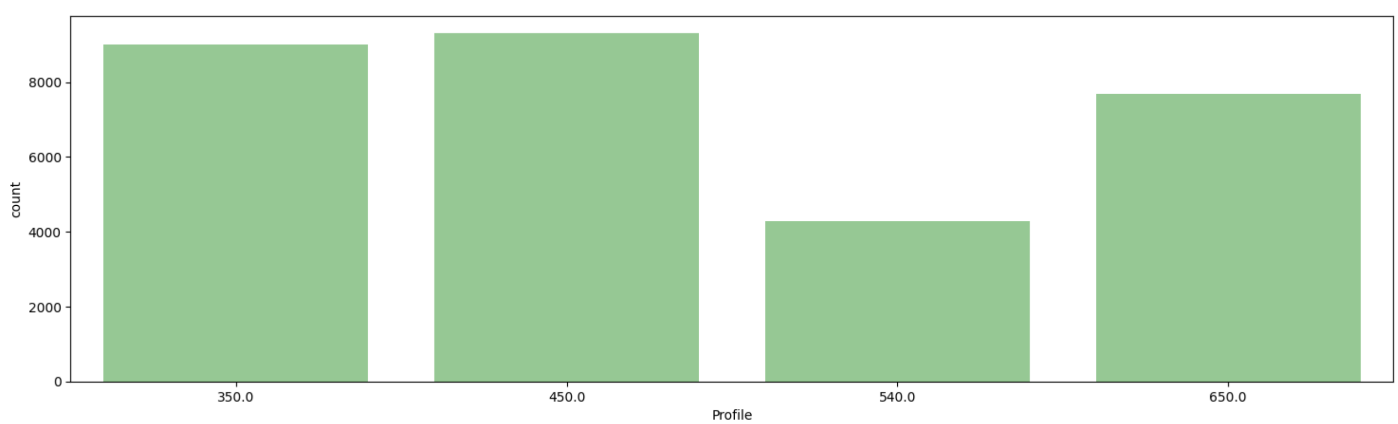

Figure 5-12: Target Vehicle Dataset Histogram

- The first n number of rows of samples in the dataset are as listed in table 5-7. It helps us to observe the dataset for any patterns or trends in the data, particularly when there are many columns or rows to consider.
- Visualize and understand the range of values and any anomalies present in the given list of the dataset for all attributes.
- We can see that the range of values for each variable is quite large, for example, ‘MotorSpeedRpm’ ranges from 271.75 to 7183.62. This suggests that the data is varying and may require normalization or scaling before analysis.
- In general, it can be seen that this dataset contains attributes with varying ranges based on the given units. It is for certain that input features must be scaled and normalized to achieve better accuracy.

Table 5-7: Target Vehicle Dataset Samples List

	MotorSpeedRpm	MotorTorqueNm	DCBusVoltage	DCBusCurrentAmps	PhaseCurrentAmps	MaxDriveTorqueNm	MaxBrakingTorqueNm	MotorCtrlTempDC	MotorTempDC	Profile
0	7183.625	921.600	2457.950	1843.250	5760.000	12134.400	12134.400	14.600	83.000	350.000
1	3855.625	972.800	2457.950	1971.250	5760.000	12185.600	12185.600	14.600	83.000	350.000
2	3599.625	972.800	2457.950	1971.250	5760.000	12185.600	12185.600	14.600	83.000	350.000
3	271.750	921.600	2457.950	1843.250	5760.000	12185.600	12185.600	14.600	83.000	350.000
4	527.625	972.800	2457.950	2073.650	5760.000	12185.600	12185.600	14.600	83.000	350.000

The sorted statistical summary based on the standard deviation of each attribute in the target vehicle dataset can be seen in the table 5-8. This table can be useful for quickly identifying key features of the dataset and for making decisions about how to preprocess and analyze the data.

- The descriptive statistics features including count, mean, standard deviation, minimum, 25th percentile, 50th percentile (median), 75th percentile, and maximum, of each attribute is shown in the table 5-8.
- The minimum and maximum values for each variable vary across a wide range, which suggests that the variables have different scales and ranges.
- There is significant variation the values which needs to be normalized to draw much better inferences.
- The dataset contains non-null values of 30276 samples/observations.
- The high standard deviation values for ‘MotorTorqueNm’, ‘MotorSpeedRpm’, ‘PhaseCurrentsAmps’, ‘MaxDriveTorqueNm’ and ‘MaxBrakingTorqueNm’, indicates that the data is widely dispersed and there may be some outliers. These attributes have relatively higher mean and maximum values compared to the other features. This suggests that they might be related which eventually could help in accurate estimation of target feature.

- The min and max values for ‘DCBusVoltage’ are also concerning, as they suggest that there may be some extreme values in the dataset that could skew the analysis.

Table 5-8: Statistical Analysis of Target Vehicle Data Attributes

	count	mean	std	min	25%	50%	75%	max
MotorTorqueNm	30276.00	5115.47	2909.62	0.00	2457.60	5785.60	6860.80	13056.00
MotorSpeedRpm	30276.00	3909.73	2242.11	14.00	2068.12	3860.00	5654.50	7960.88
PhaseCurrentAmps	30276.00	3107.62	1983.14	0.10	1433.80	2842.10	4941.00	6528.40
MaxDriveTorqueNm	30276.00	9866.93	1714.28	0.20	8448.00	9472.00	12134.40	13056.00
MaxBrakingTorqueNm	30276.00	9866.93	1714.28	0.20	8448.00	9472.00	12134.40	13056.00
DCBusCurrentAmps	30276.00	1476.56	1037.76	0.10	436.10	1383.05	2432.10	3252.05
Time	30276.00	415.65	275.45	0.01	178.57	384.82	639.72	999.72
DCBusVoltage	30276.00	2170.55	166.90	1741.15	2022.75	2253.15	2304.35	2457.95
Profile	30276.00	483.71	114.58	350.00	350.00	450.00	650.00	650.00
MotorTempDC	30276.00	132.69	23.11	83.00	121.00	131.00	156.00	165.00
MotorCtrlTempDC	30276.00	17.86	1.35	14.60	17.20	17.80	18.80	20.40

Table 5-9: Normalized Statistical Summary of Target Vehicle Data Attributes

	count	mean	std	min	25%	50%	75%	max
MotorSpeedRpm	30276.00	-0.00	1.00	-2.19	-0.73	0.08	0.79	1.60
MotorTorqueNm	30276.00	0.00	1.00	-2.25	-0.85	0.31	0.64	2.31
DCBusVoltage	30276.00	-0.00	1.00	-1.93	-1.00	0.39	0.81	2.36
DCBusCurrentAmps	30276.00	-0.00	1.00	-2.17	-0.92	0.13	0.91	1.42
PhaseCurrentAmps	30276.00	0.00	1.00	-2.46	-0.75	0.04	0.93	1.49
MaxDriveTorqueNm	30276.00	0.00	1.00	-4.66	-0.84	-0.25	1.35	1.93
MaxBrakingTorqueNm	30276.00	0.00	1.00	-4.66	-0.84	-0.25	1.35	1.93
MotorCtrlTempDC	30276.00	-0.00	1.00	-2.07	-0.58	-0.14	0.67	2.22
MotorTempDC	30276.00	0.00	1.00	-2.15	-0.51	-0.07	1.01	1.40

The statistical summary of target vehicle dataset after applying normalization technique can be seen in the table 5-9.

- The negative min values for some variables, such as ‘MotorSpeedRpm’, suggest that there may be some issues with the data collection process, such as incorrect sensor readings.

- All the attributes have the standard deviations of 1.0, indicating that their values are widely dispersed from the mean. Means that data is spread out but not too widely.
- The ‘MaxDriveTorqueNm’ and ‘MaxBrakingTorqueNm’ attributes have the lowest minimum value of -4.66, which is considerably lower than the other attributes. This suggests that there are outliers or errors in the data for this attribute. This can further be analyzed using box-plots (as in section 5.6.3.3).
- We can see that the mean of each variable is close to zero, indicating that the data is centered around zero. The standard deviation of each variable is close to one, indicating that the data is spread out. Overall, it is certain that the variables are normalized or standardized with z-score method (standard scalar).
- To mitigate these issues, it may be necessary to perform some data cleaning and preprocessing steps, such as identifying and removing outliers, correcting erroneous sensor readings, and ensuring that the data is properly scaled and normalized.

Therefore, to address the abnormalities or outliers in the inferred attributes, it may be necessary to perform some data cleaning and preprocessing steps, such as identifying and removing outliers, correcting erroneous sensor readings, and ensuring that the data is properly scaled and normalized.

5.4.3.2 Redundant & Missing Data Samples

Target vehicle dataset is checked for the duplicate samples and for any null values which helps in further improving the dataset for the model.

- Motor2 dataset is checked for the duplicate samples. It contains the missing data samples (figure 5-13A) in each column across the entire dataset. These missing values are known because the raw data attributes are recorded from messages with different periodicity.
- Forward fill imputation method is used to fill the missing rows of specific attribute. Other methods such as mean, median and interpolation can also be employed. Backward fill is not applicable in case of time-series data.
- The heatmap is plotted (figure 5-13 B) after applied imputation technique to recheck if any missing values are left. The dataset is clean plot with no missing values or null values.

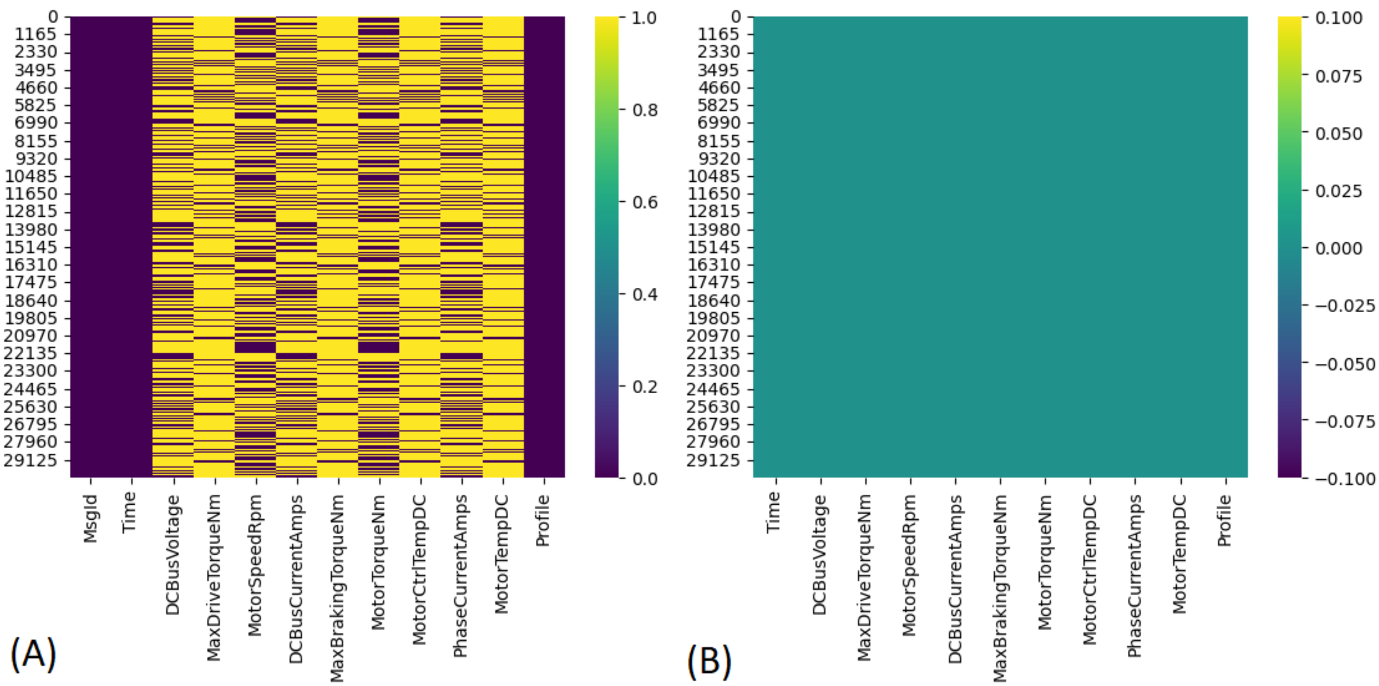

Figure 5-13: Missing Values Visualization through Heat Map for Target Vehicle Dataset

5.4.3.3 Outliers' Detection Analysis

Observations such as type of variance and the presence of outliers are inferred from the graphical boxplot representation of the target vehicle dataset (figure 5-14).

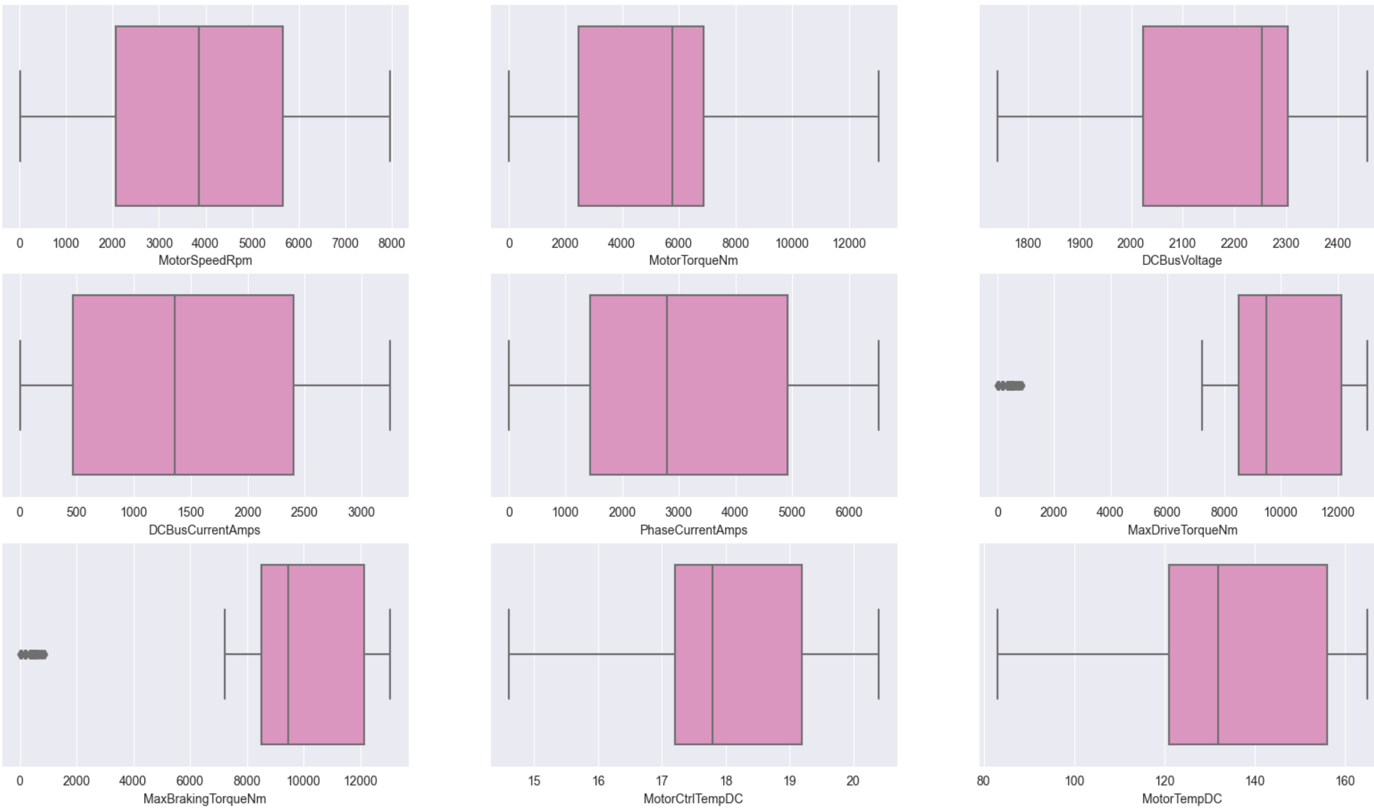

Figure 5-14: Target Vehicle Box-Plots Representation

- We can see whiskers for the attributes 'MaxDriveTorqueNm' and 'MaxBrakingTorqueNm'. Indicating the presence of outliers or genuine wide range of values. As be manually investigated and decided accordingly.
- Few attributes show equal variance along the median line such as 'MotorSpeedRpm' and 'DCBusCurrentAmps'.

5.4.3.4 Variable Distribution

Density plots shows the distribution of variables as displayed in figure 5-15 for all attributes of target vehicle dataset.

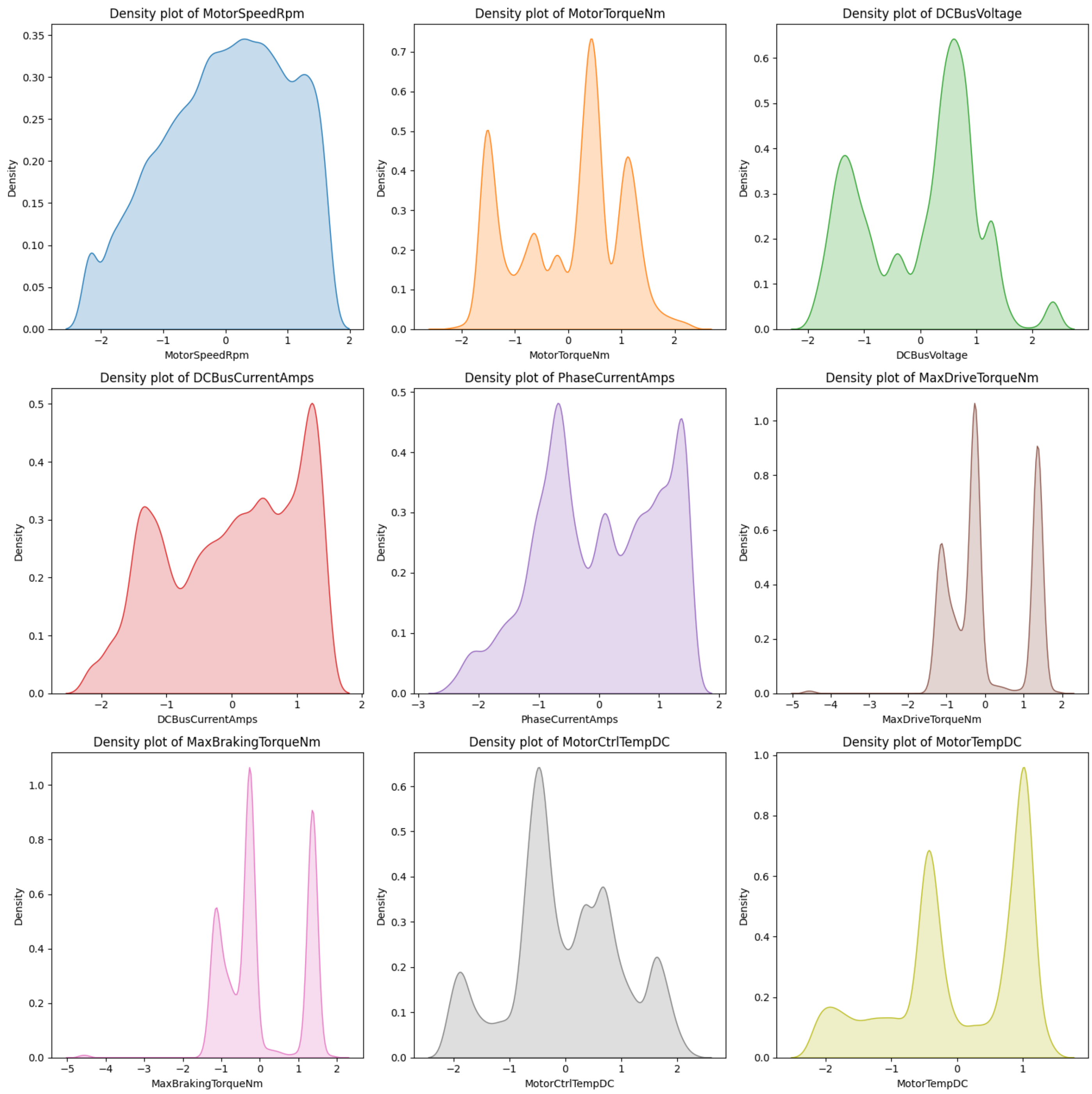

Figure 5-15: Target Vehicle Data Distribution Density Plots

Looking at the histograms following observations are found:

- We can see that some variables, such as ‘MotorTorqueNm’, ‘DCBusCurrentAmps’, and ‘PhaseCurrentAmps’ are skewed on the positive side because long tail is in the positive direction. In case of positive skews mean is larger than the median. There will be some large values that pull the mean towards the right causing the skewedness.
- Whereas ‘DCBusVoltage’ and ‘MotorTempDC’ attributes are negatively skewed. That means skewed on the negative side, i.e., long tail is in negative direction. Here median is larger than the mean. There will be some large values that pull the mean towards the left causing the skewedness.
- Attributes ‘MotorCtrlTempDC’ and ‘MotorTempDC’ have a relatively similar and normal distribution, with most of the data clustered around the mean.
- The histogram plot also shows the distribution of each variable for each load profile, which are identified by the ‘Profile’ variable. We can see that the distribution of some variables, such as ‘DCBusCurrentAmps’, ‘PhaseCurrentAmps’ and ‘MotorTorqueNm’ varies depending on the load profile, indicating that these variables may be affected by the conditions of the test. This helps in identifying the strong predictors for target variable.

Overall, the density plot gives us a visual representation of the distribution of each variable and how they vary across the different load profiles. It gives us an idea of how the variables may be related to each other.

5.4.3.5 Variable Correlation Analysis

Attributes correlation coefficients are highlighted in the heatmap illustrated as in the figure 5-16.

- The correlation coefficients are calculated between all pairs of variables in the dataset and it identifies the pairs of variables that have a correlation coefficient greater than 0.5 in absolute value.
  [('MaxDriveTorqueNm', 'MaxBrakingTorqueNm'), ('MaxDriveTorqueNm', 'MotorTempDC'), ('MaxBrakingTorqueNm', 'MotorTempDC'), ('MotorCtrlTempDC', 'MotorTempDC')]
- The absolute average correlation coefficient (*AbsAvgCorrCoef*) between *all pairs of variables* in the dataset is 0.2004, which suggests that there is a moderate degree of correlation among the variables. This suggests that some of the variables may be dependent on others and may not provide independent information.
- The *AbsAvgCorrCoef* for each attribute of the dataset are given below respectively:
  [(MotorSpeedRpm: 0.120), (MotorTorqueNm: 0.248), (DCBusVoltage: 0.268), (DCBusCurrentAmps: 0.196), (PhaseCurrentAmps: 0.178), (MaxDriveTorqueNm: 0.399), (MaxBrakingTorqueNm: 0.399), (MotorCtrlTempDC: 0.357), (MotorTempDC: 0.438)]

- Looking at the *AbsAvgCorrCoef* for each attribute, it can be seen that the attributes ‘MotorTempDC’ and ‘MotorCtrlTempDC’ have relatively high correlation coefficients (0.357-0.438).
- For instance, ‘MotorTempDC’ and ‘MotorCtrlTempDC’ have high *AbsAvgCorrCoef* i.e., 0.357-0.438 respectively, which implies that they are highly correlated with other variables in the dataset. Two or more variables with high *AbsAvgCorrCoef* values, may indicate that they both follow similar pattern. And we know that attributes are temperature readings of motor and its controller temperature measured in °C.
- On the other hand, ‘MotorSpeedRpm’, ‘PhaseCurrentAmps’, and ‘DCBusCurrentAmps’ have relatively low absolute average correlation coefficient, indicating that it is weakly correlated with other attributes in the dataset. Low value may indicate that it is not related to the other variables and may not be useful for analysis.
- It may be necessary to use algorithms that are less sensitive to multicollinearity, such as decision trees and random forest.

These findings are valuable for understanding the relationships between variables and take actions based on this information. To mitigate potential issues caused by the high correlation among some attributes, it may be necessary to perform feature selection or feature engineering. This involves removing highly correlated variables or combining variables to create new features that are less correlated. We are using this inference from *AbsAvgCorrCoef* values, in the selection of the attributes.

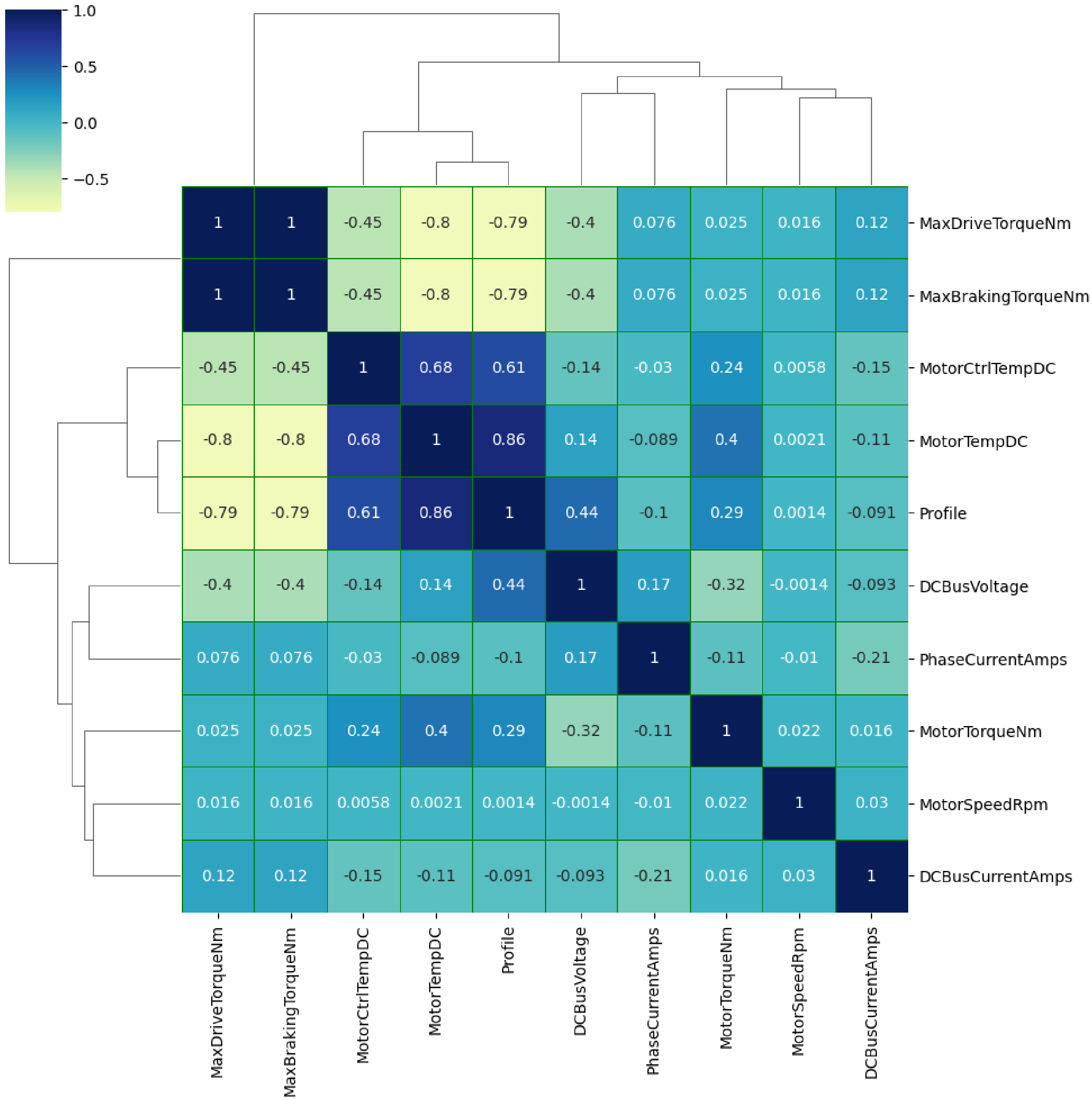

Figure 5-16: Data Correlation Matrix

Pairwise relationships between all the variables in the dataset is plotted using scatterplots as illustrated in figure 5-17. The diagonal plots show distribution of each variable.

- Variables 'MotorCtrlTempDC', and 'MotorTempDC', can be seen to have the direct strong linear relationship with each other. Hence, Predictor variables for anyone of them shall help to get the output for another variable. That's the reason, 'MotorCtrlTempDC' is recommended to be used as predictor variable, assuming based on above inferences, that they are linearly related.

- Similarly, 'MaxDriveTorqueNm' and 'MaxBrakingTorqueNm' are linearly related.

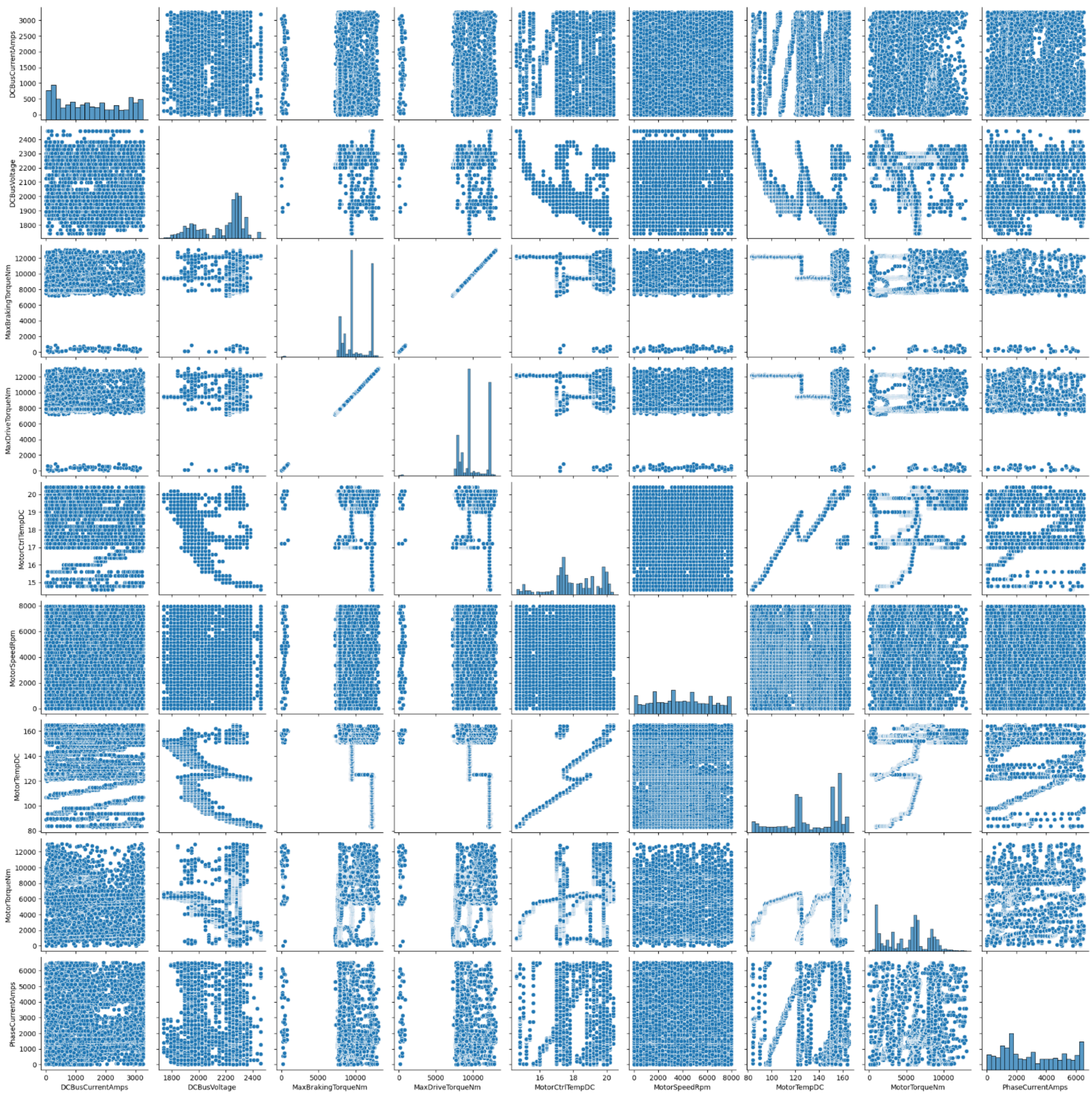

Figure 5-17: Attribute Pairs Plot Diagram

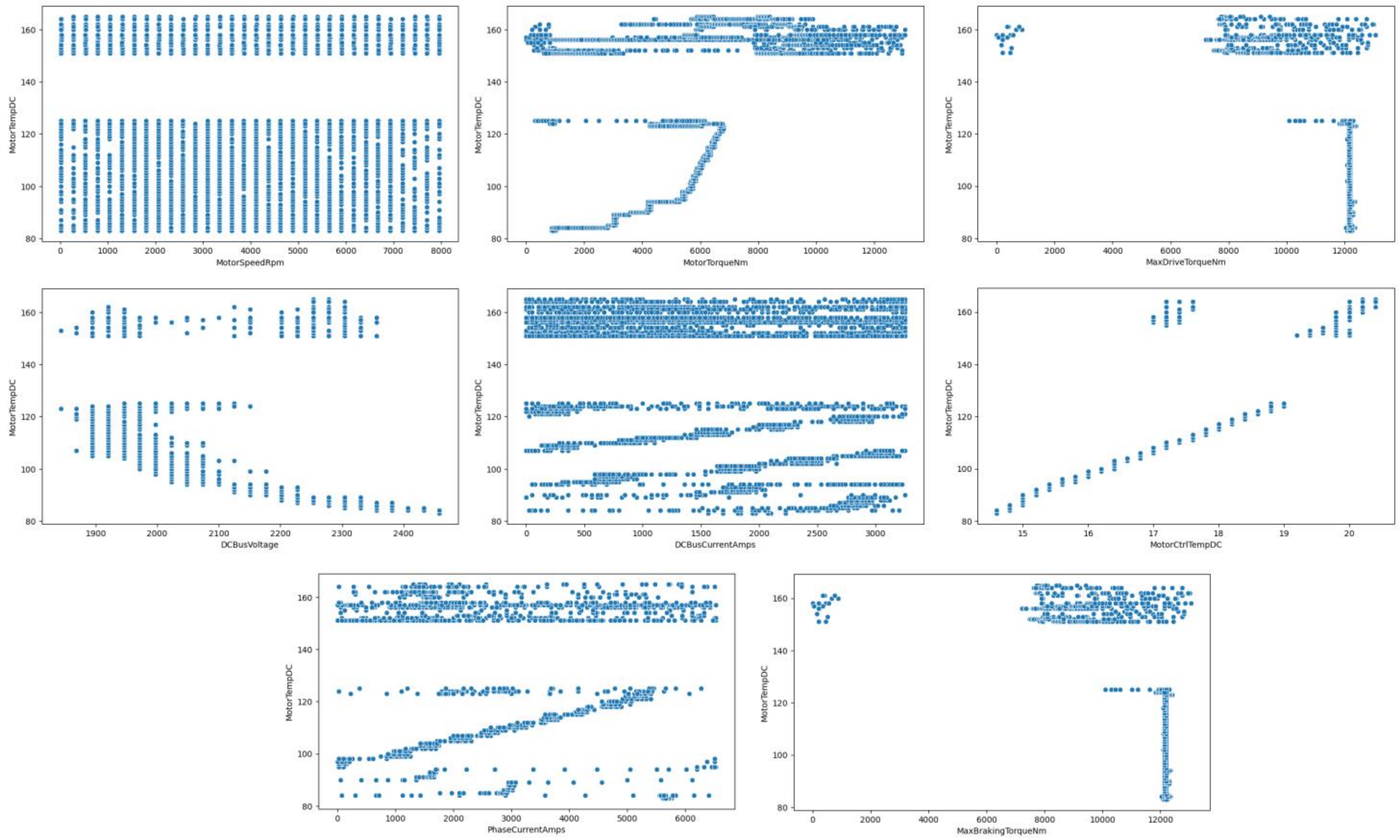

Figure 5-18: Scatter Plot for Input Features vs Target variable

5.4.3.6 Data Scaling Techniques Analysis

Target vehicle dataset is not normalized as we could see in previous analysis from table 5-7 that all the attributes have varied range of values in Motor2 dataset. Large variation in range of values is not desired when machine learning (ML) algorithms are applied reason being it would increase the computation time.

Table 5-10: Target Vehicle Dataset Normalized Samples List

	MotorSpeedRpm	MotorTorqueNm	DCBusVoltage	DCBusCurrentAmps	PhaseCurrentAmps	MaxDriveTorqueNm	MaxBrakingTorqueNm	MotorCtrlTempDC	MotorTempDC	Profile
17630	-1.303	0.469	-1.803	0.169	-1.469	-0.255	-0.255	1.216	148.000	450.000
7656	-1.306	0.593	-1.502	-1.234	1.071	1.350	1.350	0.671	123.000	350.000
23976	-1.002	-0.654	0.203	-0.777	-1.142	-1.132	-1.132	1.604	152.000	650.000
10401	-1.469	-1.527	1.275	-1.917	1.394	-0.285	-0.285	0.011	121.000	450.000
10875	-0.866	-1.554	1.275	-1.917	1.420	-0.285	-0.285	-0.143	121.000	450.000

- Standard scaler or the Z-score technique evaluates to have a better performance with lowest Mean Squared Error (MSE) of 117.25 when compared to other techniques (figure 5-19) such as Min-Max scaler, Robust scaler, Max Abs Scaler, and Power transformer with MSE values as specified in table 5-11.
- Normalized dataset attributes can be seen in the table 5-10.

Table 5-11: Data Scaling Evaluation Results

Data Scaling	MSE
Min Max Scaler	117.25
Standard Scaler	***117.05***
Robust Scaler	117.57
Max Abs Scaler	117.45
Power Transformer	122.28

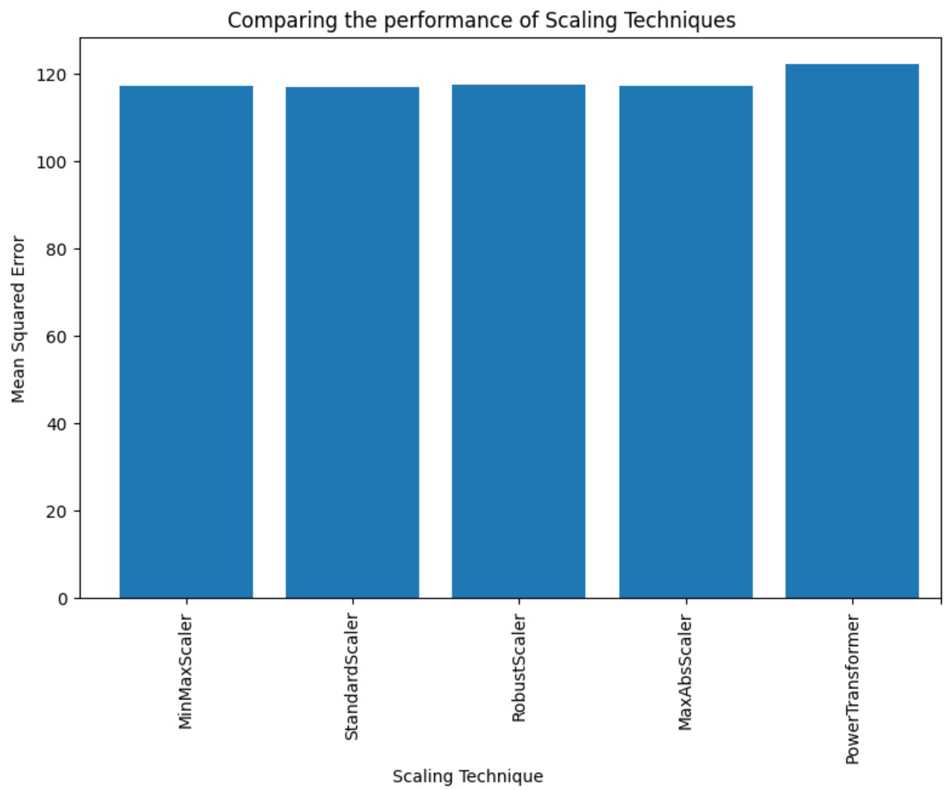

Figure 5-19: Scaling Techniques MSE evaluation

5.4.3.7 Feature Selection

After performing data pre-processing, analysis is done based on data distribution, outliers and variable correlation of attributes on the dataset.

- Inferences from boxplots, density plots and heatmap correlation matrix suggest the strong linear relationship between the attributes denoting temperatures readings i.e., ‘MotorCtrlTempDC’, and ‘MotorTempDC’ attributes.

- In order to gain a deeper understanding of the correlation among the two temperatures, we evaluate the plots (figure 5-20) representing the feature values for a selection of randomly chosen load profiles.
- The subplots validate that the two temperature features have a similar pattern. Amongst the two, the motor temperature exhibits the highest variability, followed by motor controller temperature. This disparity becomes prominent when there is a considerable fluctuation in the motor temperature.

Figure 5-20: Motor Temperatures analysis across all load profiles

- It can be inferred from the observed pattern of two temperatures i.e., 'MotorCtrlTempDC', and 'MotorTempDC', that they are influenced in the same manner based on behavior of the predictor variables. Both the temperature readings are from separate sources signifying sensors reading of target motor and its controller hence, design decision is taken that this correlated variable must be used as an input feature which will behave as one of the strongest predictors of the target variable.

- Hence, lets observe the target feature i.e., 'MotorTempDC' distribution over entire dataset (as in figure 5-20). Data values seems to be driven by load profiles and the torque, that's the reason for varied temperature values across the plot.

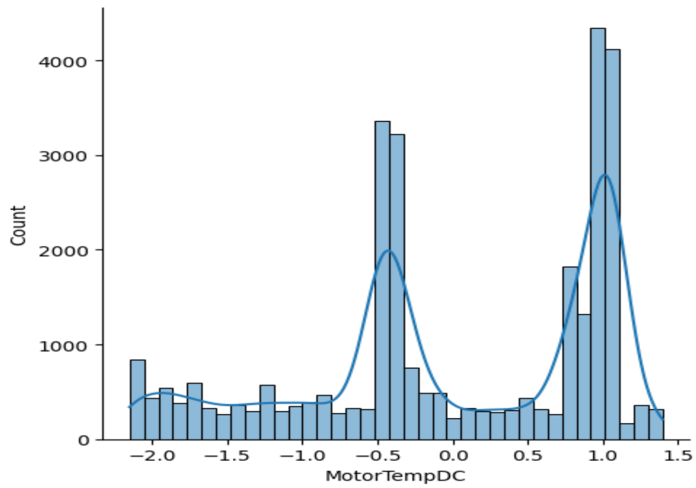

Figure 5-21: Motor Temperature of Target Vehicle Data Distribution Plot

- We analyze the target feature i.e., motor controller temperature vs input features such as torque and motor speed (as in figure 5-21).
- Relation can be observed between the motor temperature and torque (as in figure 5-22). When torque is varied, the change in motor temperature is observed.

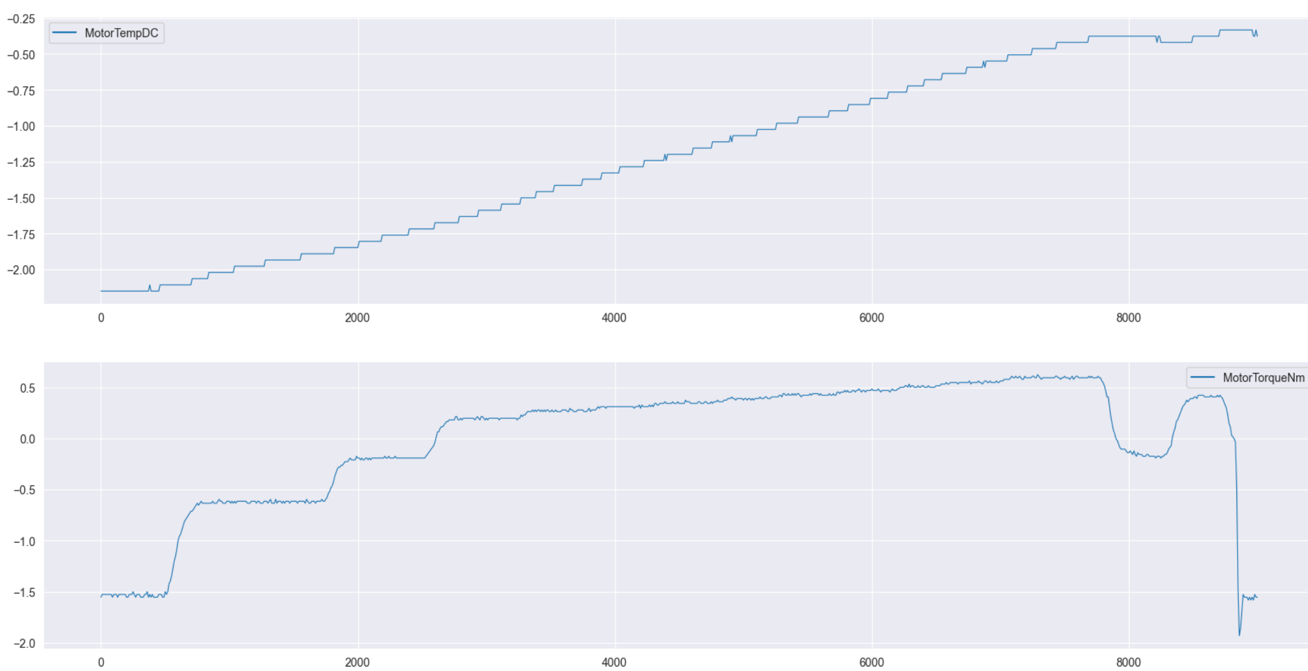

Figure 5-22: Motor Temperature vs Motor Speed or Torque Plot

5.5 Optimal Hyper Parameters

Experiments are performed using PMSM and Motor2 datasets (Section 5.3.2) and after multiple rounds of model training for targeted algorithms (section 4), optimal hyper-parameters are empirically obtained. The empirical parametric values are listed in the following sections for PMSM and Motor2 datasets.

5.5.1 RFR Parameters Configuration

List of model parameters for Random Forest Regressor (RFR) algorithm [15] are detailed in table 5-12.

Table 5-12: Hyperparameters for Random Forest Regressor

PARAMETER NAME	**DESCRIPTION**	**DEFAULT**	**VALUE (PMSM)**	**VALUE (Motor2)**
N Estimators	Defines the number of trees in the forest.	100	100	200
Criterion	Function used to measure split quality.	MSE	MSE	MSE
Max Sample	Determines the portion of the initial dataset assigned to each tree.	None	None	None
Max Features	The maximum number of features assigned to individual trees	1.0	1	4
Max Depth	Defines depth limit of each tree	None	None	None
Min Sample Split	Required minimum number of samples for splitting the tree	2	2	2
Max Leaf Nodes	To limit the further growth of nodes.	None	None	None
Min Samples Leaf	Allowed maximum number of samples in each node	1	1	1
Bootstrap	Set the samples to be used for training either random samples or the whole dataset is to be used to build each tree.	True	False	False (Whole data set is used)
Random State	Randomness of both the	None	0	1

	bootstrapping of the samples used for building trees and the selection of features considered for finding the best split at each node.			

5.5.2 XGBoost Parameters Configuration

List of model parameters for Extreme Gradient Boosting (XGBoost) model [16] are detailed in table 5-13.

Table 5-13: Hyperparameters for Extreme Gradient Boosting

PARAMETER NAME	DESCRIPTION	DEFAULT	VALUE (PMSM)	VALUE (Motor2)
Learning rate (eta)	Define the step size. Range [0,1]	0.3	0.3	0.3
Min Split Loss (gamma)	Leaf node is partitioned only in case of minimum loss reduction. Range [0,∞]	0	0	0
Max Depth	Defines the max allowed depth of a tree. Higher value may lead to overfitting. Range [0,∞]	6	6	6
N Estimators	Number of runs for model to learn	100	400	400
Sub Sample	Randomly sampling training data in the given ratio [0,1]	1	0.5	0.5
Column sample by tree	Ratio of sub sample when tree is constructed	1	0.9	0.9
Min Child Weight	Minimum required sum of instance weight for partitioning [0,∞]	1	1	1
Reg Alpha	Weights L1 regularization [0,∞]	0	0.3	0.3
Reg Lambda	Weights L2 regularization [0,∞]	1	0.7	0.7
Scale Pos Weight	Used to balance the weights.	1	03	03

5.5.3 LSTM Parameters Configuration

Hyper Parameters for LSTM configuration are briefly discussed in section 4.3, Experiments were conducted referring to different set of hyperparameters combinations used by prior research studies and our understanding while observing the empirical results. Figure 5-23 and 5-24 are the illustrations created using Netron library which shows the layers of LSTM model used for the prediction of motor temperature. It highlights the details like activation function, layer type, its units, along with the shape [5].

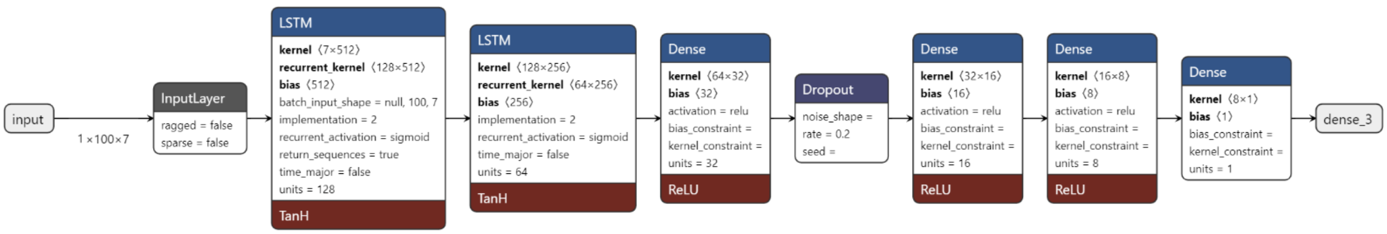

Figure 5-23: LSTM Model Layer Architecture when using PMSM Dataset

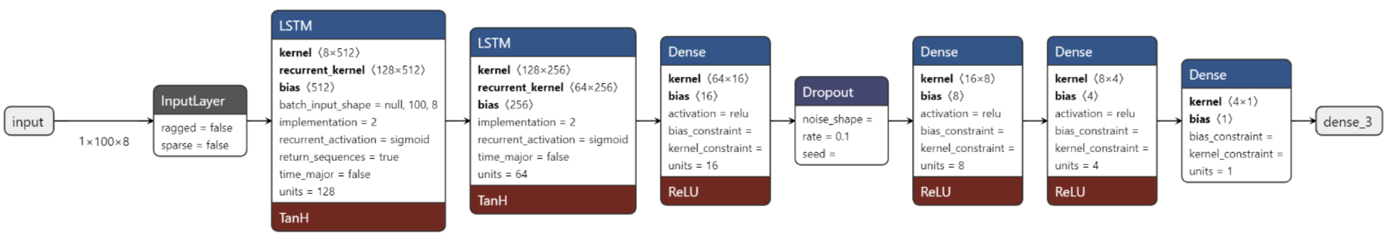

Figure 5-24:LSTM Model Layers Architecture when using Target Vehicle Dataset

List of model parameters for Long Short-Term Memory (LSTM) model are detailed in table 5-14.

Table 5-14: Hyperparameters for LSTM

PARAMETER NAME	DESCRIPTION	VALUE (PMSM)	VALUE (Motor2)
Loss function	Function used to the accuracy of predicted values against actual values.	MSE	MSE
Input sequence length	Length of input sequence fed to the network.	180	180
Hidden dimension	Number of memory cells used to capture temporal dependencies in the input sequence.	100	100
Output dimension	Number of neurons in the output layer.	1 * 4	1 * 1

Batch size (β)	The number of samples to use in each training batch.	64	64
Optimizer	Used to optimize loss function and update model weights during training.	Adam	Adam
Learning rate	The step size used in the weight updates.	0.0005	0.0005
Hidden Layer Activation Function	Utilized to introduce non-linearity into the network and increase learning capacity.	Tanh	Tanh
Dropout	It is used for regularization and determines what fraction of inputs are to be randomly set to zero during training. Purpose is to prevent overfitting and improve generalization.	0.1	0.1

5.5.4 CNN Parameters Configuration

Experiments were conducted for 1-D CNN, referring to different set of hyperparameters combinations used by prior research studies and our understanding while observing the empirical results. Figure 5-25 and 5-26, illustrates (created using Netron library) the layers of LSTM model used for the prediction of motor temperature. It highlights the details like activation function, layer type, its units, along with the shape [5].

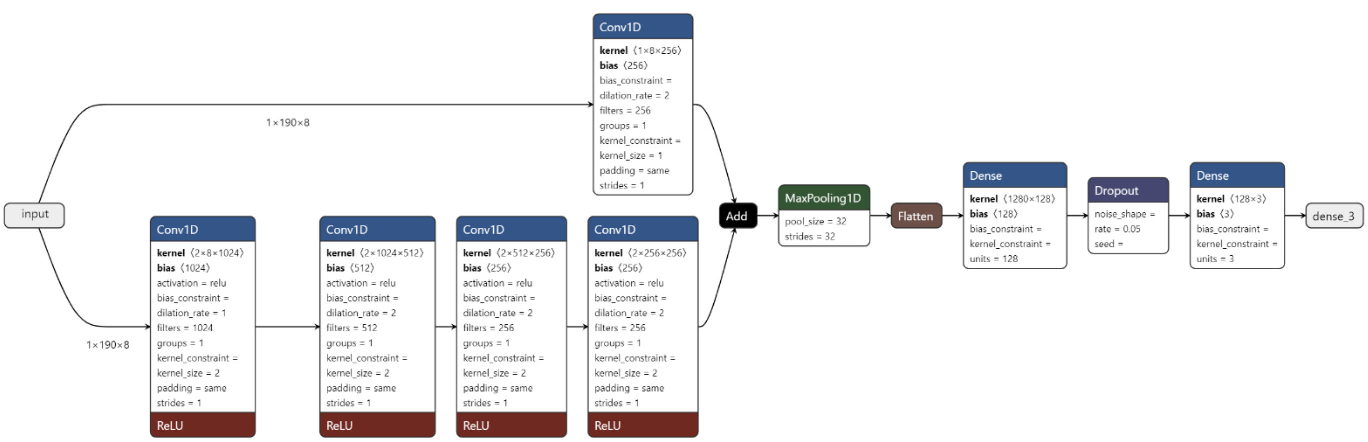

Figure 5-25: 1-D CNN Model Layer Architecture when using PMSM Dataset

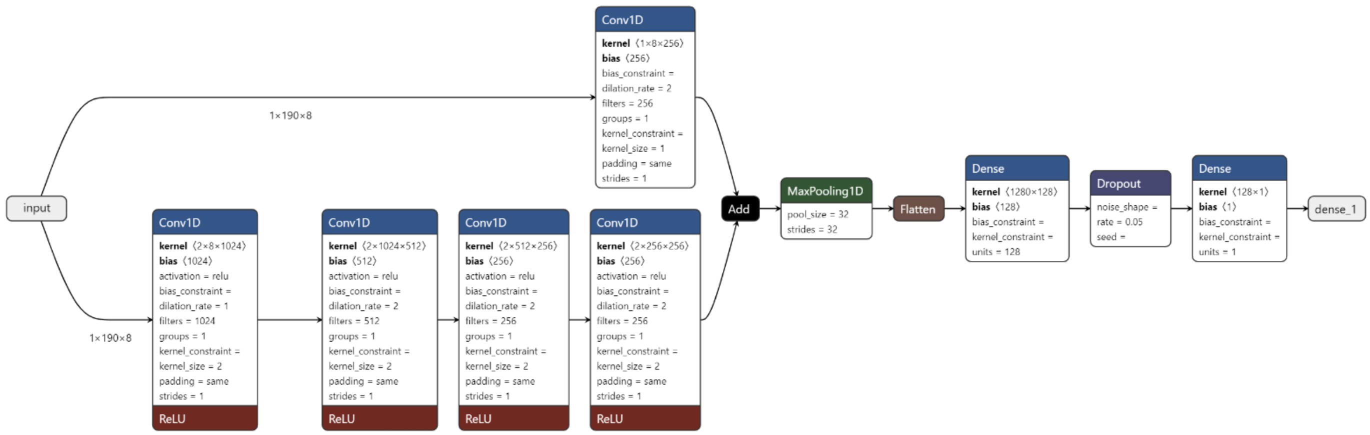

Figure 5-26: 1-D CNN Model Layers Architecture when using Target Vehicle Dataset

List of model parameters for Convolutional Neural Networks (CNN) model [5] are detailed in table 5-15.

Table 5-15: Hyperparameters for CNN

PARAMETER NAME	**DESCRIPTION**	**VALUE (PMSM)**	**VALUE (Motor2)**
Loss function	Measures the prediction error and adjusts model parameters during training to minimize the difference between predicted and actual outputs.	MSE	MSE
Input sequence length	Defines the size of the input sequence.	190	190
Hidden dimension	Determines the number of filters in each layer.	Figure 5-25	Figure 5-26
Output dimension	Number of filters in each layer of network.	3	1
Batch size (β)	Number of samples presented to the network at once during each training iteration.	603	603
Optimizer	Used to optimize loss function and update model weights during training.	Adam	Adam
Learning rate	The step size used in the weight updates.	0.05	0.05
Hidden layers activation function	Utilized to introduce non-linearity into the network and increase learning capacity.	Figure 5-25	Figure 5-26
Kernel Size	Utilized to control the size of the receptive field and extract features from input data.	2	2

Random State	Used to initialize the internal random number generator to ensure reproducibility and consistency of the results.	42	42
Epochs	The number of times the entire dataset is used to train the model.	50	50

5.6 Conclusion

In conclusion, this study conducted a design of experiments using two datasets: a PMSM dataset for experimentation and a target vehicle dataset for model building. The four models trained namely RFR, XGBoost, LSTM, and CNN, were evaluated for their performance in predicting the target variable. Based on the results, it can be concluded that each model has its strengths and weaknesses in terms of accuracy, efficiency, and interpretability. Overall, this study provides valuable insights into the use of different models in predicting the target variable, which can be used to inform future research and decision-making in this field.

In data analysis it is observed that, dataset has some pre-normalization done as detailed in section 5.4.2.4. It is important to carefully handle duplicate and null values in a dataset to ensure accurate and reliable analysis and modeling. Within short-period of time XGBoost model produces high outcome within limited computational resources needs.

Chapter 6

6 Results and Discussions

This chapter analysis the visualizations results of the models and compares the model performance of the experiments conducted with the available research papers. Upon completion of model training, the algorithms (which are detailed in section 4) are assessed based on identified evaluation metrics (as discussed in section 2.3).

6.1 Applied Model Results

Earlier in the sections, data pre-processing and feature selection is done to prepare the input and target features, which are fed to the targeted algorithms (section 4). Targeted algorithms are trained using optimal hyper-parameters (section 5.5) with prepared input and target features. Let us examine the results of different models when trained with PMSM and Target Vehicle Motor2 datasets (detailed in section 5.3.2).

One of the approaches used to determine the model training and how well the model fits the data is by analyzing residual plots. The residual plot is a scatter plot that displays the residuals (the difference between the actual and predicted values) on the y-axis and the predicted values on the x-axis. Residual errors variance score is a statistical measure that represents the proportion of the variance in the dependent variable that is predictable from the independent variables.

6.1.1 Random Forest Regressor

Now that the Random Forest Regression (RFR) model has been fitted, its performance can be examined by doing the analysis of different plots and metrics. Such as residual plots for train and test dataset, along with the actual vs predicted plot for the target feature.

- The illustrated plots as in figure 6-1 and 6-4, helps us visualize the distribution of errors and identify patterns and trends in the residuals for the RFR model.
- The residual plot is a scatter plot that displays the residuals (the difference between the actual and predicted values) on the y-axis and the predicted values i.e., *target feature attribute* on the x-axis.
- Another way to evaluate the performance of an RFR model is to create actual vs predicted graphs for four random test runs samples. The graph plots the number of samples on the x-axis whereas the motor temperature on the y-axis. This plot (figure 6-2 and 6-5) helps to determine how accurately the model predicts the target feature i.e., *Stator winding* and *Motor2* temperature from trained models of respective datasets.

- Figure 6-2 and 6-5 depicts the graph for four test runs samples randomly chosen from all test runs of dataset.
- On the first column of figure 6-2 and 6-5, graph is drawn for test data. Noise or fluctuations in the form of scattered red line can be observed in the predicted temperature when there is a variation observed in the actual signal value.
- The spread red lines (figure 6-2 and 6-5) shows variability in predicted value. The variability in predicted value is smoothened (second column of figure 6-2 and 6-5) by applying moving average method on the predicted values to improve the accuracy and performance of the model.

6.1.1.1 PMSM Dataset Results for RFR

- The residual scatter plot is plotted w.r.t predicted values i.e., *stator winding temperature* on the x-axis. It must be noted that dataset is normalized.
- The residual plot looks to be ideal as it shows the random scatter, indicating that the model is accurately capturing the variability in the data.
- The residual plot does not show any patterns or trends, which suggests that model is capturing the variable data accurately.

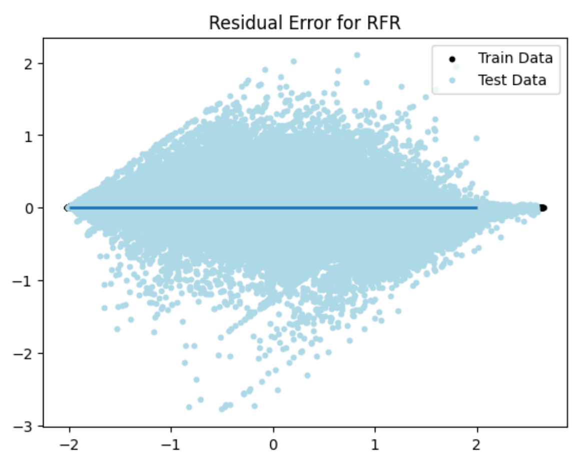

Figure 6-1: Residual Error Plot for RFR

- Overall MSE evaluation for RFR model is 0.006.
- Moving average method is used to smooth out the fluctuations in the predicted data. Here, window size of 100 is used.

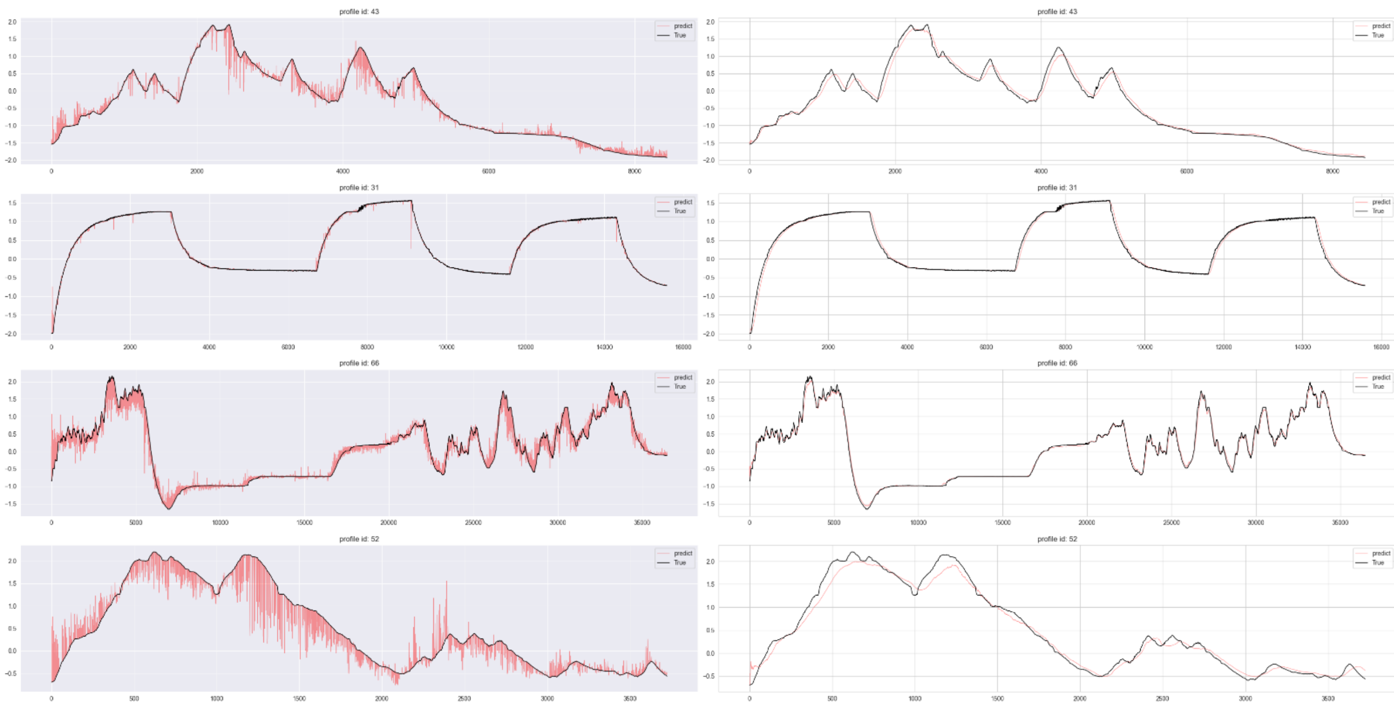

Figure 6-2: RFR Actual Vs Predicted for Random Samples (With and Without Smoothing)

- Test run with profile id 76, is chosen for the validation purpose and its results are compared with research papers [2], [3]. Figure 6-3 shows the actual vs predicted plot for the same.

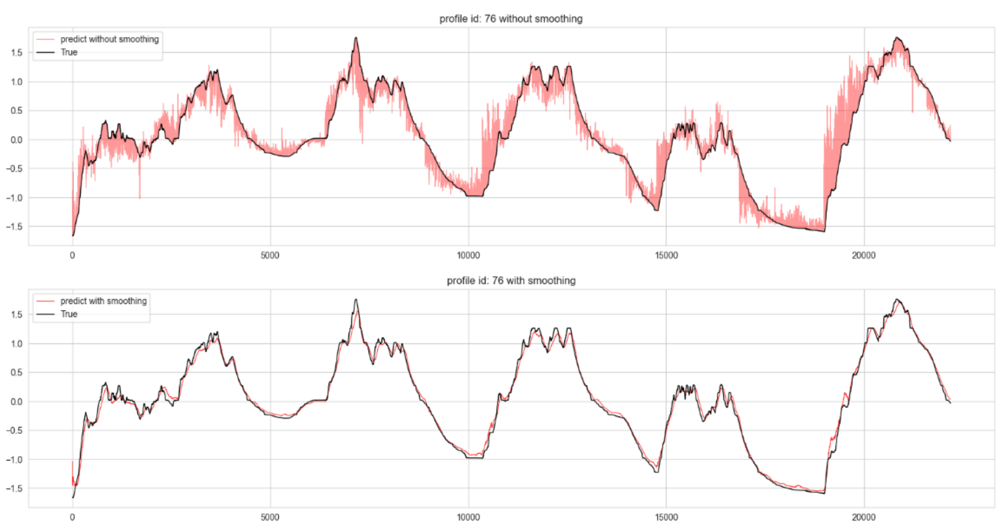

Figure 6-3: RFR Actual Vs Predicted (With and Without Smoothing)

6.1.1.2 Target Vehicle Dataset Results for RFR

After examining the experimental dataset results, let us now observe the behavior of the Random Forest Regression (RFR) model with target vehicle dataset (section 5.3.2.2).

- Target vehicle dataset is normalized during the data pre-processing stage as detailed in section 5.4.3.6.
- The residual plot (figure 6-4) looks to be ideal as it shows the random scatter, indicating that the model is accurately capturing the variability in the data. In some instances, negative values such as -0.3, -0.5, and -0.8 shows that they may be underfitting the results.
- Residual errors variance score for train and test data is 0.999 and 0.997 respectively. This means that model explains a very high proportion of the variability in both the training and test data. This suggests that the model is likely to be a good fit for the data and is able to predict the dependent variable with a high degree of accuracy. However, it is important to note that the performance of the model on new data (i.e., the test data) is not as good as the training data.

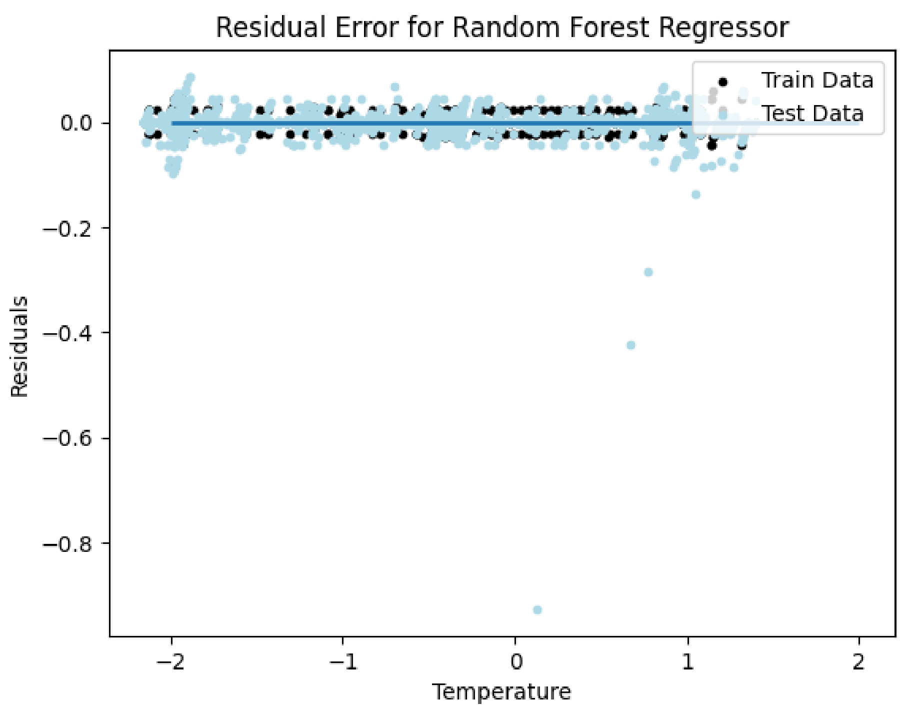

Figure 6-4: Residual Error Scatter Plot for RFR (Target Vehicle)

- Overall MSE evaluation for RFR model is 0.1140.
- The noise or fluctuations in the predicted values is smoothened with moving average method with *window size of 10*, the size is kept minimal since the number of samples in the dataset are less.
- Load profile 450, is chosen for the validation purpose. Figure 6-6 shows the actual vs predicted plot for the target vehicle motor temperature prediction.

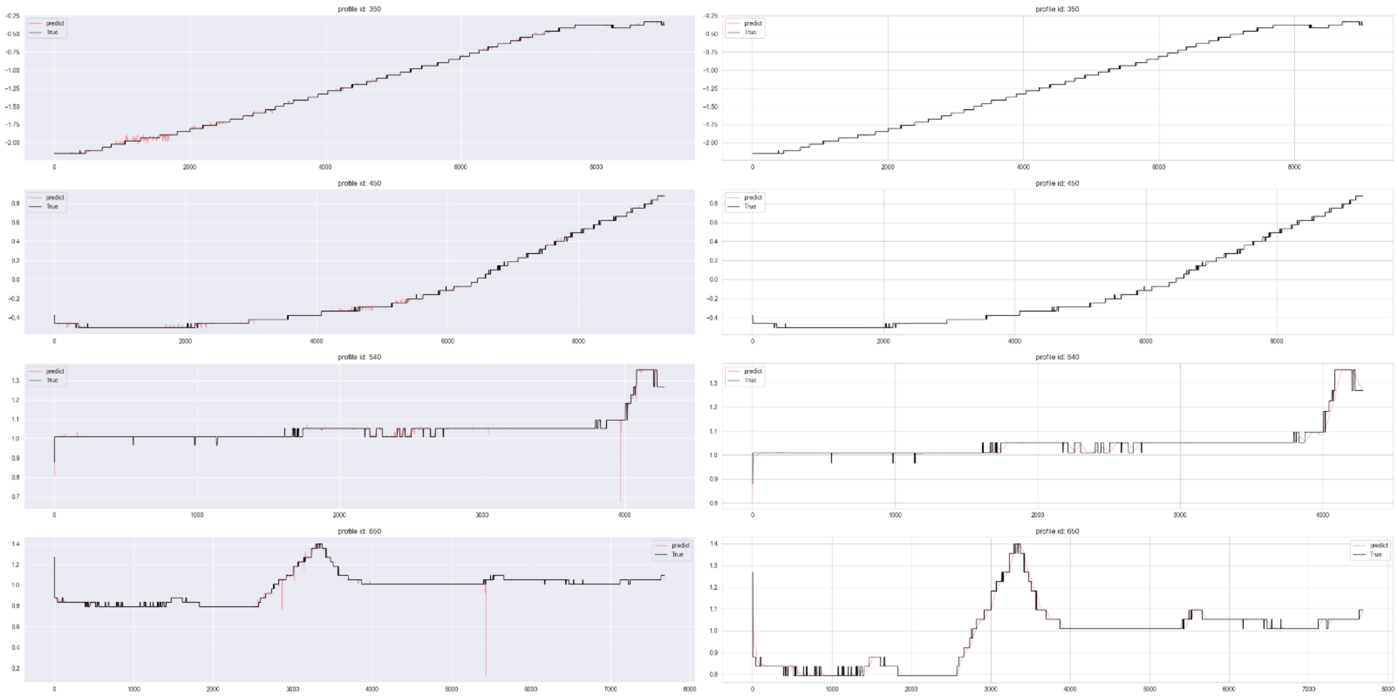

Figure 6-5: RFR Actual Vs Predicted for all load profiles (With and Without Smoothing)

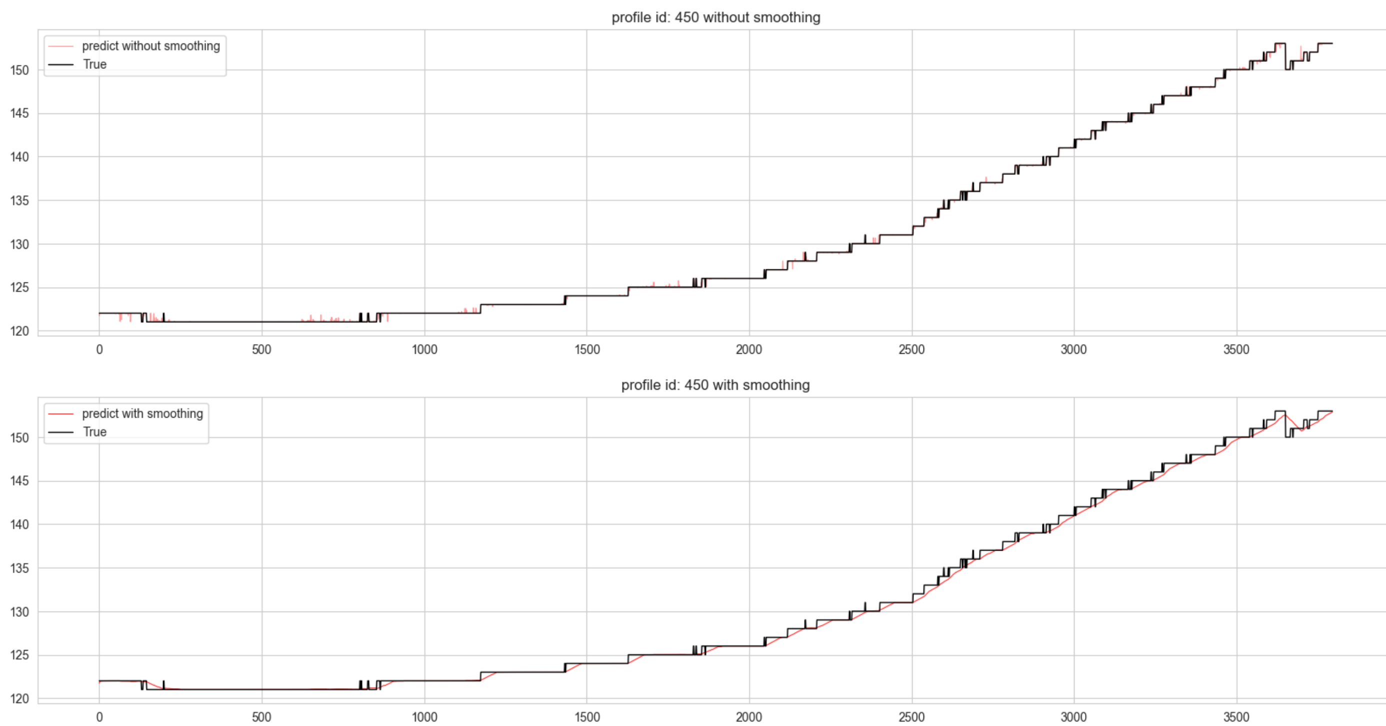

Figure 6-6: RFR Actual Vs Predicted (With and Without Smoothing)

6.1.2 Extreme Gradient Boosting

Trained model i.e., Extreme Gradient Boosting (XGBoost) must be evaluated. Its performance can be analyzed by examining different plots and metrics. Such as residual plots for train and test dataset, along with the actual vs predicted for the target feature.

- The illustrated plots as in figure 6-7 and 6-9, helps us visualize the distribution of errors and identify patterns and trends in the residuals for the trained XGBoost model.
- The *residual scatter plot* illustrates the residuals (the difference between the actual and predicted values) on the y-axis and the predicted values i.e., *target feature attribute* on the x-axis.
- XGBoost model performance is visualized by plotting the graphs with number of samples on the x-axis whereas the motor temperature on the y-axis. This plot (figure 6-7 and 6-9) helps to determine how accurately the model predicts the target feature i.e., *Stator winding* and *Motor2* temperature from trained models of respective datasets.
- Figure 6-8 and 6-10 depicts the graph for actual vs predicted stator temperature of the respective datasets for the XGBoost model.

6.1.2.1 PMSM Dataset Results for XGBoost

- The residual scatter plot is plotted w.r.t predicted values i.e., *stator winding temperature* on the x-axis. It must be noted that dataset is normalized.
- The residual plot does not show any patterns or trends, which suggests that model is capturing the variable data accurately.
- Minimal variability is observed around 60 and 70 (°C) in the figure 6-7 and does not affect the model performance.
- Residual errors variance score for train and test data is 0.9987 and 0.9975 respectively. This means that model explains a very high proportion of the variability in both the training and test data. This suggests that the model is likely to be a good fit for the data and is able to predict the dependent variable with a high degree of accuracy. However, it is important to note that the variance score of the model on new data (i.e., the test data) is not as good as the training data.
- Train and Test accuracy of the model is 99.87 % and 99.75 % respectively.
- Overall MSE evaluation for XGBoost model is 2.0529.

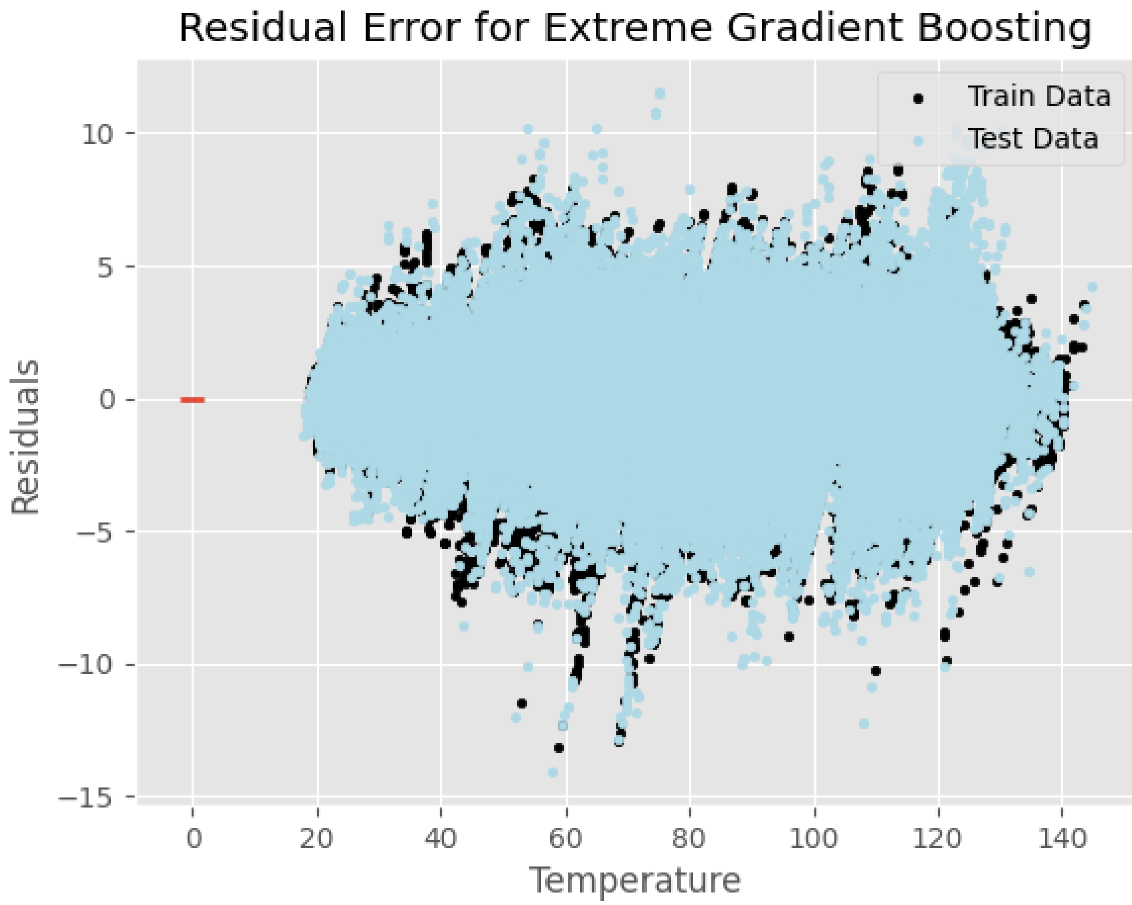

Figure 6-7: Residual Error Plot for XGBoost

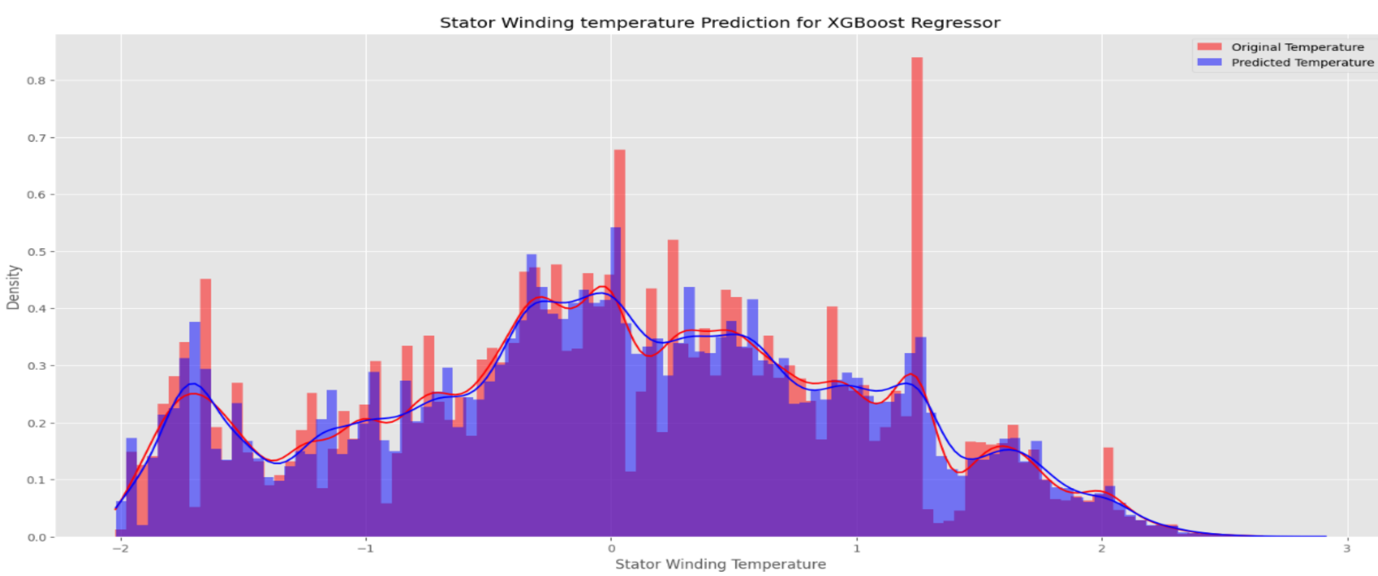

Figure 6-8: Actual vs Predicted Temperature for XGBoost

6.1.2.2 Target Vehicle Results for XGBoost

Experimental dataset results of the Extreme Gradient Boosting (XGBoost) model with target vehicle dataset (section 5.3.2.2) are listed below.

- Target vehicle dataset is normalized during the data pre-processing stage as detailed in section 5.4.3.6.
- The residual plot is equally scattered around zero, indicating that the model is accurately capturing the variability in the data. In some instances, negative values of test data shows that they may be underfitting the results buts its quite minimal hence may not impact the overall accuracy of the model.
- The residual plot does not show any patterns or trends, which suggests that model is capturing the variable data accurately.
- Variability is high at 1 on the x-axis, which shows that predicted values are lower than the actual. Occurrence of variability is less frequent, so impact is considered to be minimal.

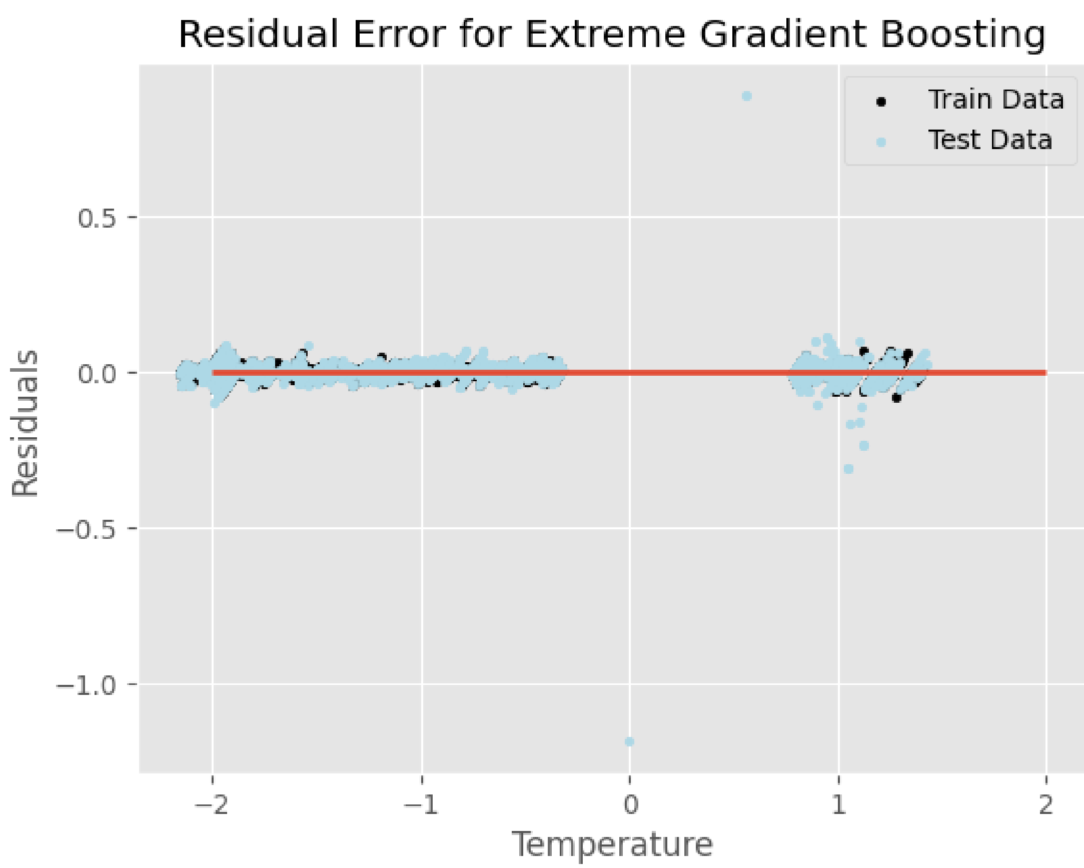

Figure 6-9: Residual Error Scatter Plot for XGBoost (Target Vehicle)

- Residual errors variance score for train and test data is 0.9999 and 0.9995 respectively. This means that model explains a very high proportion of the variability in both the training and test data. This suggests that the model is likely to be a good fit for the data and is able to predict the dependent variable with a high degree of accuracy. However, it is important to note that the performance of the model on new data (i.e., the test data) is not as good as the training data.
- Train and Test accuracy of the model is 99.99 % and 99.95 % respectively.
- Overall MSE evaluation for XGBoost model is 0.0007.

Table 6-1: Actual Vs Predicted with XGBoost (Target Vehicle)

Original Temperature	Predicted Temperature
157.0	156.979599
157.0	157.003693
87.0	86.589165
156.0	156.175079
158.0	157.925110

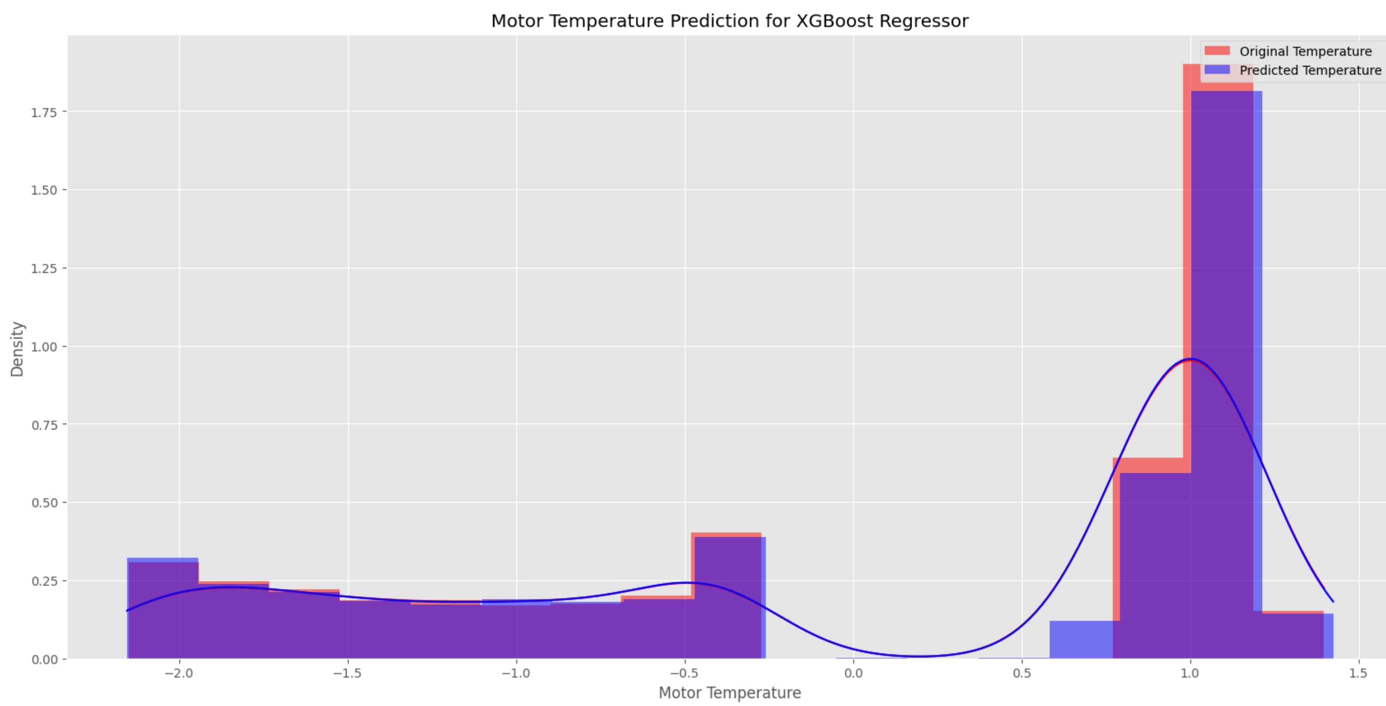

Figure 6-10: Actual vs Predicted Temperature for XGBoost

6.1.3 Long Short-Term Memory

LSTM model is trained for sequence prediction task. When training an LSTM model, it's important to keep track of two things: accuracy and loss. The accuracy tells us how well the model is doing at making predictions, while the loss tells us how far off those predictions are from the correct values. During training, the model's accuracy and loss are continuously updated as it sees more data. By plotting the accuracy and loss against the number of training epochs (or iterations), we can see how the model is improving over time.

The loss plot (figure 6-11 and 6-13) shows how the model's loss decreases over time. We want the loss to get as low as possible, indicating that the model is making accurate predictions. However, if the loss is low on the training data but high on the validation data, it may be overfitting, meaning it's too specialized to the training data and doesn't generalize well to new data.

The plot of MSE (or accuracy) vs epoch is called the 'history' of the LSTM model during training. This plot (figure 6-11 and 6-13) shows how the model's accuracy changes over the course of training, with the training accuracy shown in blue and the validation accuracy shown in red.

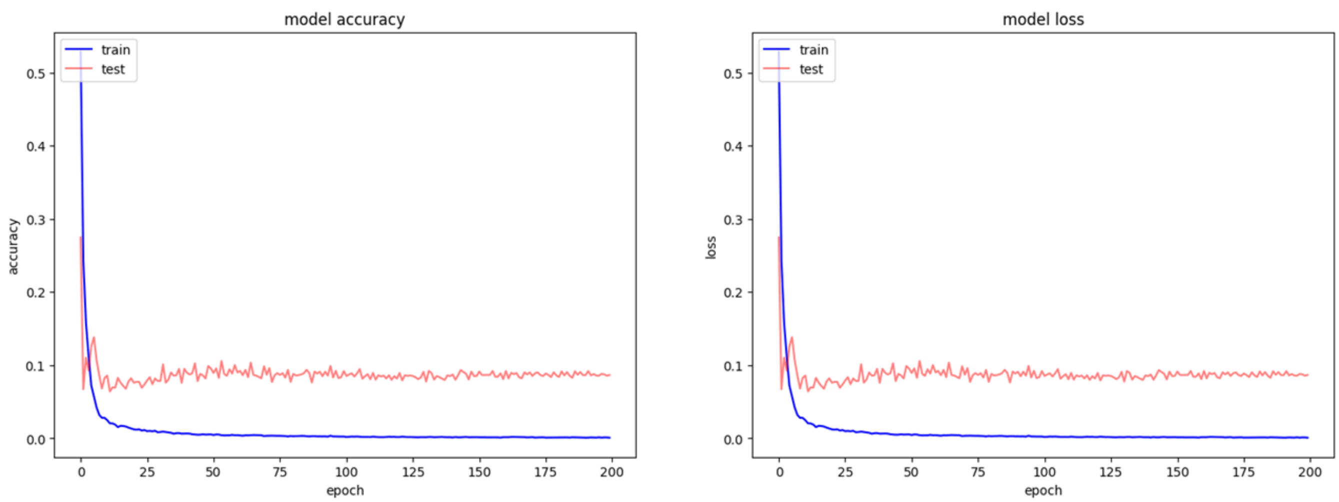

Figure 6-11: LSTM Accuracy and Model Loss vs Epoch Graph

6.1.3.1 PMSM Dataset Results for LSTM

- We can see in the figure 6-11, that the training accuracy improves over time and converges to a stable value, indicating that the model is learning to make accurate predictions on the training data. Similarly, we observe that validation accuracy does not follow similar pattern, it rather varies and it can be seen that it does not converge to a stable value, indicating that the model is not able to generalize well to new, unseen data.

- Training accuracy converges well and becomes stable hence it is not overfitting, hence able to generalize well to new data.

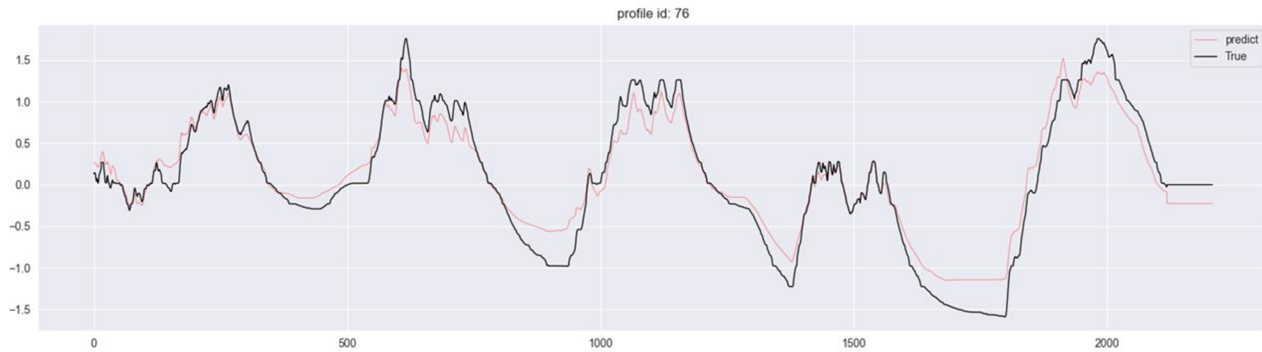

Figure 6-12: Actual vs Predicted Temperature for LSTM

- Figure 6-12 shows the pictorial representation of actual vs predicted stator winding temperature for test run profile id 76.
- Loss can be seen bit high on validation data as compared to training. It seems to be overfitting, meaning it's too specialized to the training data and doesn't generalize well to new unseen data.

Overall, the plot of MSE vs epoch provides important insights into the behavior of the LSTM model during training, and can help guide adjustments to the model architecture and training process to improve its performance.

6.1.3.2 Target Vehicle Results for LSTM

LSTM model is trained with target vehicle data sets. One of the key metrics to evaluate its performance is model’s training and validation accuracy, which is often measured by the mean squared error (MSE).

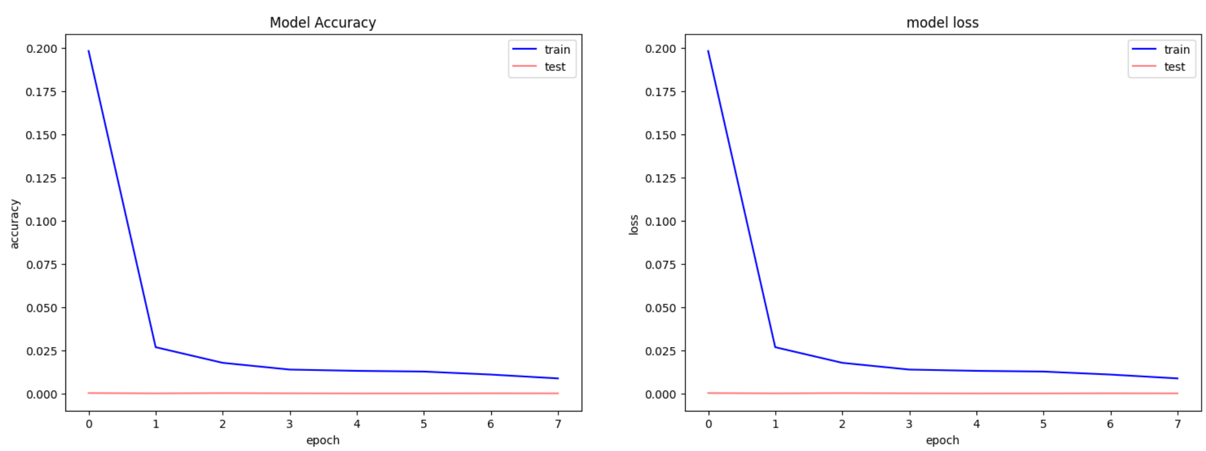

Figure 6-13: LSTM Accuracy and Model Loss vs Epoch Graph

- We can see in the figure 6-13, that the training accuracy converges to a stable value, but the number iterations is quite less that means, that means model did not learn well.
- The loss plot (figure 6-13) shows how the model's loss decreases over time. The loss decreases with number of epochs as low as possible, indicating that the model converges in a very small number of iterations. MSE of total test loss is 0.0020.
- Figure 6-14 shows the pictorial representation of actual vs predicted motor temperature for all load profiles.
- We can see that currently the results are overfitting as per the graphs.

Overall, the plot of MSE vs epoch can provide important insights into the behavior of the LSTM model during training, and can help guide adjustments to the model architecture and training process to improve its performance.

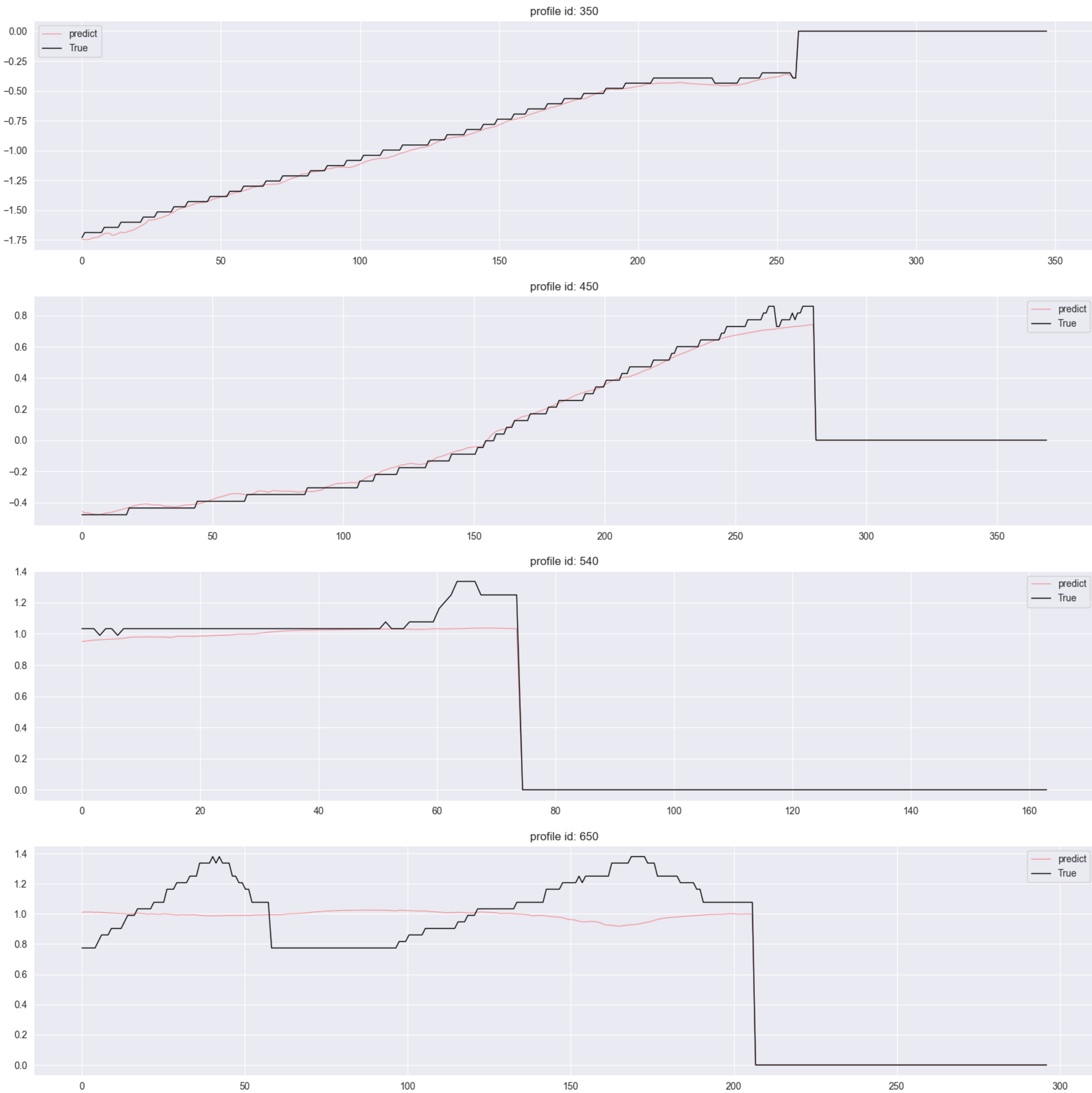

Figure 6-14: Actual vs Predicted Temperature for LSTM

6.1.4 Convolutional Neural Networks

6.1.4.1 PMSM Dataset Results for CNN

The Convolutional Neural Network (CNN) model is trained on the PMSM dataset (section 5.3.2.1), with a specific configuration as described in table 5-15, the results of model and its explanation w.r.t target parameters i.e., stator temperatures. The testing stage is then conducted, and the results are visualized to evaluate the model's performance.

- The plot in the figure 6-15, shows the training and validation mean squared errors (MSE) for each epoch of training.
- It helps to visualize the training process of the model, showing the trend of the training and validation MSE across epochs. The purpose of this graph is to visualize the performance based on training process of the model. In addition to monitoring if the model is overfitting or underfitting.
- We can see that, the training error (green dots) are not considerably lower, than the validation error (blue line), which indicates that the model might be slightly overfitting, but not significant.

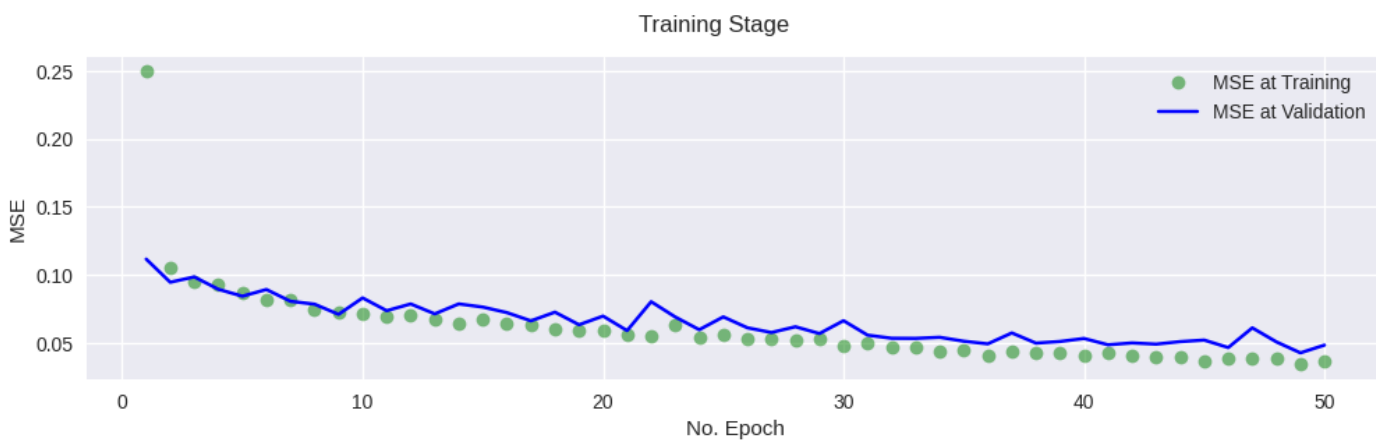

Figure 6-15: MSE at Training and Testing Stage for CNN

- The plot in the figure 6-16, shows a scatter plot for each target column comparing the predicted values to the actual values. Each scatter plot also shows the R2 score, MSE, and RMSE for the target column. The plot provides a visual representation of how well the model is predicting the target stator temperatures.
- The R2 score measured is 94 %, which tells how well the model is able to explain the variance in the actual target stator winding values, while the MSE (0.059) and RMSE (0.24) indicate the magnitude of the errors between the predicted and actual stator winding temperature values.

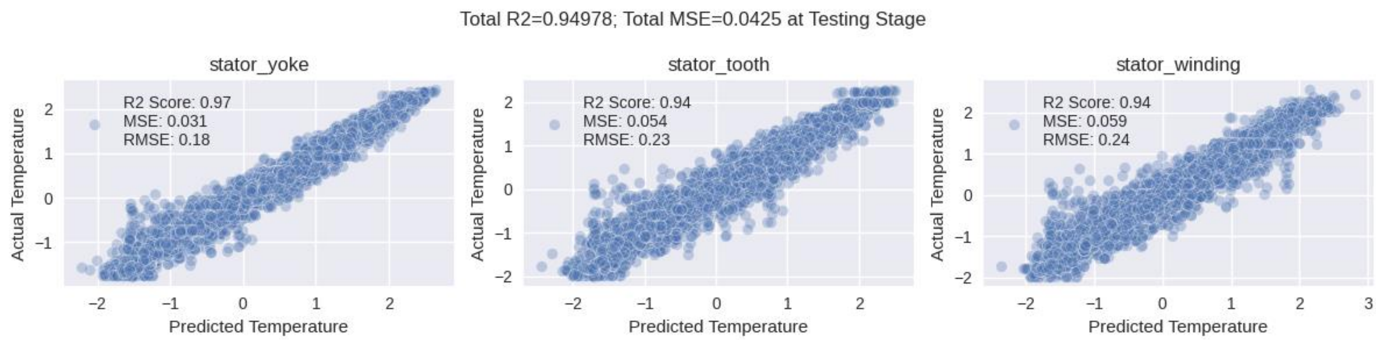

Figure 6-16: Actual vs Predicted Temperature for CNN

6.1.4.2 Target Vehicle Results for CNN

The Convolutional Neural Network (CNN) model is trained on the target vehicle Motor2 dataset (section 5.3.2.2), with a specific configuration as described in table 5-15, the results of model and its explanation w.r.t target parameters i.e., stator temperatures. The testing stage is then conducted, and the results are visualized to evaluate the model's performance.

- The plot in the figure 6-17, shows the training and validation mean squared errors (MSE) for each epoch of training.
- It helps to visualize the training process of the model, showing the trend of the training and validation MSE across epochs. The purpose of this graph is to visualize the performance based on training process of the model and monitor whether the model is overfitting or underfitting.

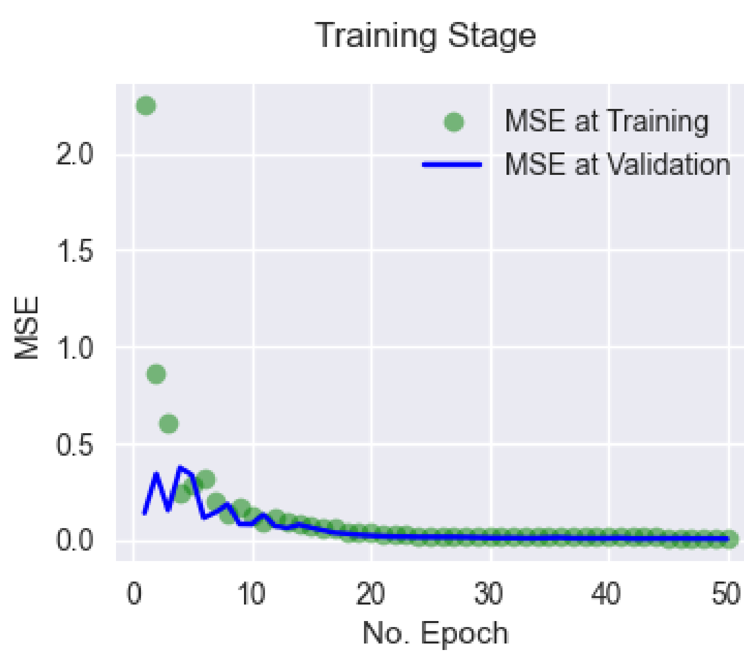

Figure 6-17: MSE at Training and Testing Stage for CNN (Target Vehicle)

- We can see that the training error (green dots) are aligned with the validation error (blue line), which indicates that the model might has fitted well.
- The plot in the figure 6-18, shows a scatter plot for each target column comparing the predicted values to the actual values. Each scatter plot also shows the R2 score, MSE, and RMSE for the target column. The plot provides a visual representation of how well the model is predicting the target stator temperatures.
- The R2 score measured is 99.30 %, which tells how well the model is able to explain the variance in the actual target motor temperature values, while the MSE (0.007) indicate the magnitude of the errors between the predicted and actual motor temperature values.

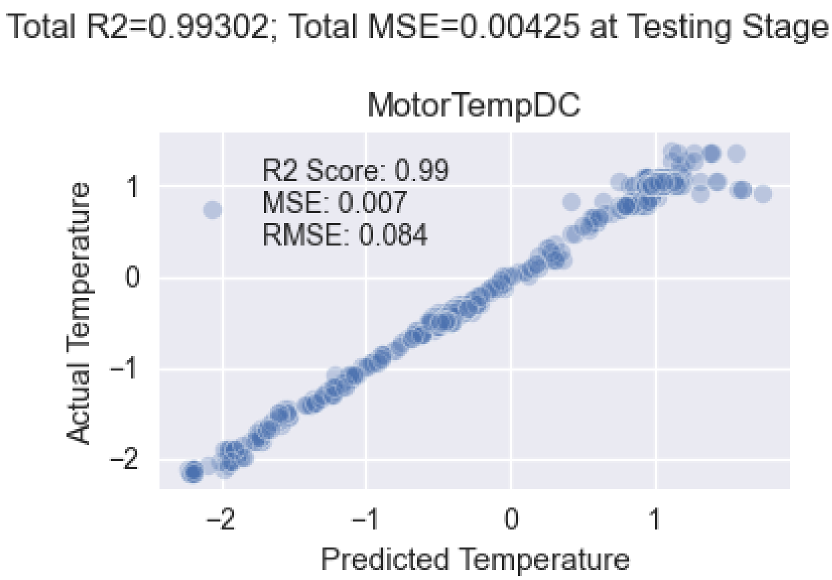

Figure 6-18: Actual vs Predicted Temperature for CNN (Target Vehicle)

6.2 Performance Evaluation

6.2.1 PMSM dataset Performance

Analysis of model training results with PMSM data set (section 5.3.2.1) is listed in table 6-2 with different evaluation metrics. These are overall test set performance results. They are compared with available research studies done on similar problem statement as ours, i.e., stator temperatures prediction of electric motor temperature.

Table 6-2: Comparative results for PMSM data set

MODEL	MAE	MSE	R2 Score	RMSE
RFR	0.0635	0.0065	99.01 %	0.25
RFR-Sampaio *et al.* [2]	-	-	-	0.0026 (TRAIN) 0.092 (TEST)
RFR-Savant *et al.* [3]	-	-	99.30 %	-
XGBOOST	0.0373	0.0026	99.74 %	0.19
XGBoost-Al-Gabalawy *et al.* [4]	-	-	-	1.226 (TRAIN) 0.8291 (TEST)
LSTM	0.1592	0.0360	94.41 %	0.40
LSTM-Hosseini *et al.* [5]	-	5.62	-	-
Global Attention-based EnDec LSTM-Li *et al.* [1]	8.75	2.82	-	-
Cen *et al.* [20]	0.2222			0.2674
CNN	0.0974	0.0204	97.95 %	0.31
CNN-Hosseini *et al.* [5]	-	3.34	99.54 %	-

- Evaluation results of applied models are compared with available research as in table 6-1.
- R2 Score for RFR 99.01%, which may not be better than available research [3] but still gives good results provided its robustness to outliers. Hence, always a preferable choice to go for.
- XGBoost out performs the results of Al-Gabalawy *et al.* with approximately more than 70% improvement in RMSE metric with test set.
- Computational time for CNN was observed to be more with large dataset.

6.2.2 Target Vehicle dataset Performance

Model training results with Target Vehicle Motor2 data set (section 5.3.2.2) are listed in table 6-3 with different evaluation metrics. These are overall test set performance results. They are compared with available research studies done on similar problem statement as ours, i.e., prediction of abnormalities in motor temperature.

Table 6-3: Evaluation Metrics results for Target Vehicle data set

MODEL	MAE	MSE	R2 Score	RMSE
RFR	0.0037	0.0003	99.97 %	0.06
XGBOOST	0.0104	0.0007	99.95 %	0.10
LSTM	0.0079	0.0020	99.16 %	0.17
CNN	0.0554	0.0204	99.30 %	0.22

- RFR and XGBoost results outperformed the other models in terms of RMSE and R2 Score.
- In-spite of CNN having better evaluation metric values of MAE and MSE, XGBoost or RFR would be preferred reason being the computational time and resource needs of deep neural network models is high. It is not wise to use such resources for less complex systems. Though, it also depends how the models are going to be applied. In case its one-time training then even CNN can be used.

Graphical user interface is created for demonstration purpose below as shown in the figure 6-19.

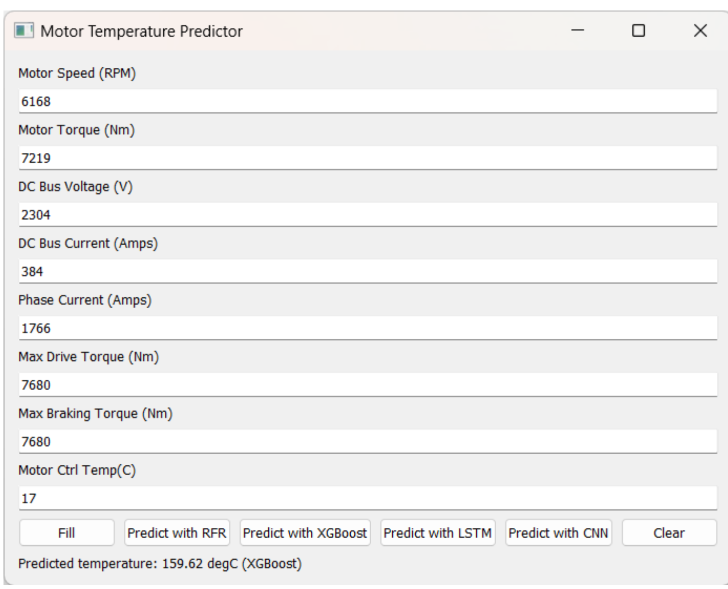

Figure 6-19: Graphical Interface for the Demonstration

Chapter 7

7 Conclusions and Future Work

This chapter concludes the thesis work and it highlights the areas where there is scope of improvement.

7.1 Conclusion

This thesis work has covered many areas in terms of architecture of the system, its state of art performance results and in what way it can actually be applied to the real time embedded software for off-road electric vehicles (EVs), which is our main objective (objectives detailed in section 1.2) of the thesis work.

This thesis work details the system architectural flow stages (section 1.3), and its sub processes (section 3.3) which when followed can lead to the ease of using AI based models to fulfill application needs and boost the approaches which can assist for example in our case the thermal management of electric motor drives when any abnormality warning or critical temperature rise is detected (section 3.3.4).

ML and DL models were trained initially with PMSM dataset (section 5.3.2.1) for experimentation to find optimal hyper-parameters (as listed in section 5.5), which were used as the starting point to use them and start building model with the target vehicle Motor2 dataset (section 5.3.2.2). It is to be noted that, in our project scope we had two motors one for traction and another for driving implements. As of date, raw data for Motor2 which is used to drive implements were available from the lab. Both of the motors are controlled separately hence, do-not seem to have any dependency. Design of experiments conducted is highlighted in figure 5-1.

Since the requirement need is not complex, such kind of problems can be dealt with one time model training which is then used for the prediction of motor temperature when real time sensor data is fed as input to the trained model. With less complex problem and time-series data. Ensemble regression based extreme gradient boosting (XGBoost) model outperforms in short period of execution time and within limited computational resources.

Evaluation metric results (table 6-3) obtained by application of targeted models i.e., RFR, XGBoost, LSTM and CNN, to the actual vehicle data-sets (section 5.3.2.2) of target off-road vehicle i.e., electric tractor, shall enrich the use-case of artificial intelligence-based algorithms. The objective of these evaluation results is to achieve a state-of-the-art study work of performance results which is one of its kind, and they will serve as valuable benchmarks for researchers who wish to compare the performance of various AI models applied to off-road vehicles.

7.2 Future Work

- Data capture process (as detailed in section 3.3.1) needs to be automated to incorporate new captured records and prepare raw data at ease. This shall help in many ways, i.e., ease of preparing dataset from the new lab results to minimize the manual integration errors to improve model accuracy in prediction.
- Development and deployment operations to be integrated with embedded system architecture design shall help to more accurately understand the AI model performance.
- Enhancement to the deployment can be made where Edge AI and IoT can be used together to address complex problems where model is expected to train at run-time with new set of real time sensor data. Such applications demand architectural updates in existing hardware for any vehicle. Reason being one part of computation i.e., reading sensors data on the vehicle, preparing raw data out of it and then sending this data to cloud. Second part of computation i.e., training with new data shall be done using cloud services which shall then send the predicted information back to the vehicle.
- Trained model with optimal hyper-parameters must be integrated with embedded eco-system to deploy it to the actual vehicle, in turn see the results of our AI application into off-road electric vehicles (EVs) at run-time.
- Currently models are trained with "Measured Input" attributes as is from the sensors recorded data. These measured inputs can further be used to extract more "Derived Input" features. Work needs to on this side dependent on domain knowledge and inclined with application requirement needs.

REFERENCES

[1] Li, Jun & Akilan, Thangarajah. (2022). "Global Attention-based Encoder-Decoder LSTM Model for Temperature Prediction of Permanent Magnet Synchronous Motors". 10.48550/arXiv.2208.00293.
[2] Scalabrini Sampaio, Gustavo, Arnaldo Rabello de Aguiar Vallim Filho, Leilton Santos da Silva, and Leandro Augusto da Silva. 2019. "Prediction of Motor Failure Time Using An Artificial Neural Network" Sensors 19, no. 19: 4342. https://doi.org/10.3390/s19194342.

[3] R. Savant, A. A. Kumar and A. Ghatak, "Prediction and Analysis of Permanent Magnet Synchronous Motor parameters using Machine Learning Algorithms," 2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC), Bengaluru, India, 2020, pp. 1-5, doi: 10.1109/ICAECC50550.2020.9339479.

[4] Al-Gabalawy, M., Elmetwaly, A.H., Younis, R.A. et al. "Temperature prediction for electric vehicles of permanent magnet synchronous motor using robust machine learning tools". J Ambient Intell Human Comput (2022). https://doi.org/10.1007/s12652-022-03888-9.

[5] S. Hosseini, A. Shahbandegan and T. Akilan, "Deep Neural Network Modeling for Accurate Electric Motor Temperature Prediction," 2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Halifax, NS, Canada, 2022, pp. 170-175, doi: 10.1109/CCECE49351.2022.9918222.

[6] Wilhelm Kirchgässner, Oliver Wallscheid, & Joachim Böcker. (2021). "Electric Motor Temperature" [Data set]. Kaggle. "https://doi.org/10.34740/KAGGLE/DSV/2161054"

[7] Design Of Experiments (DOE) reference from "Six Sigma: Complete Step-by-step guide" "https://www.sixsigmacouncil.org/wp-content/uploads/2018/08/Six-Sigma-A-Complete-Step-by-Step-Guide.pdf"

[8] D.-G. Kim and J.-Y. Choi, "Optimization of Design Parameters in LSTM Model for Predictive Maintenance," Applied Sciences, vol. 11, no. 14, p. 6450, Jul. 2021, doi: 10.3390/app11146450.

[9] Grand View Research. (2022). Edge Computing Market Analysis Report, Size, Share, Trends, and Segment Forecasts, 2022-2030. Retrieved from https://www.grandviewresearch.com/industry-analysis/edge-computing-market.

[10] W. Kirchgässner, O. Wallscheid and J. Böcker, "Estimating Electric Motor Temperatures With Deep Residual Machine Learning," in IEEE Transactions on Power Electronics, vol. 36, no. 7, pp. 7480-7488, July 2021, doi: 10.1109/TPEL.2020.3045596.

[11] O. Wallscheid and J. Böcker, "Global Identification of a Low-Order Lumped-Parameter Thermal Network for Permanent Magnet Synchronous Motors," in IEEE Transactions on Energy Conversion, vol. 31, no. 1, pp. 354-365, March 2016, doi: 10.1109/TEC.2015.2473673.

[12] J. Ruan, C. Wu, H. Cui, W. Li and D. U. Sauer, "Delayed Deep Deterministic Policy Gradient-based Energy Management Strategy for Overall Energy Consumption Optimization of Dual Motor Electrified Powertrain," in IEEE Transactions on Vehicular Technology, doi: 10.1109/TVT.2023.3265073.

[13] Curtis Controller Data-Sheet , [ONLINE] "https://cdn.curtisinstruments.com/products/datasheets/1239E_datasheet_en.pdf"

[14] Curtis Controller Manual, [ONLINE] "https://www.hpevs.com/Site/images/jpeg/documentation/1239/econtroller_532_up/ auto1239_Controller_532_revA_6-9-17.pdf"

[15] Sklearn Random Forest Regressor, [ONLINE] "https://scikit-learn/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html".

[16] XGBoost Parameters 2.0.0-dev documentation," XGBoost Parameters — xgboost 2.0.0-dev documentation. [ONLINE] "https://xgboost.readthedocs.io/en/latest/tutorials/model.html".

[17] Understanding LSTM Networks colah's blog, "https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[18] Chen, Tianqi & Guestrin, Carlos. XGBoost: A Scalable Tree Boosting System. 2016, "https://doi.org/10.1145/2939672.2939785"

[19] Google Colaboratory, "https://colab.research.google.com/".

[20] Y. Cen, C. Zhang, G. Cen, Y. Zhang, and C. Zhao, "The Temperature Prediction of Permanent Magnet Synchronous Machines Based on Proximal Policy Optimization," Information, vol. 11, no. 11, p. 495, Oct. 2020, doi: 10.3390/info11110495.

[21] H. Guo, Q. Ding, Y. Song, H. Tang, L. Wang, and J. Zhao, "Predicting Temperature of Permanent Magnet Synchronous Motor Based on Deep Neural Network," Energies, vol. 13, no. 18, p. 4782, Sep. 2020, doi: 10.3390/en13184782.