JÖNKÖPING UNIVERSITY
School of Engineering

Thermal Estimation of Electric Machine Rotor using ML for Volvo Cars

PAPER WITHIN *Artificial Intelligence*
AUTHORS: *Nadera Al-Areqi, Martin Hanke*
TUTOR: *Johannes Oetsch*
JÖNKÖPING *June 2024*

This exam work has been carried out at the School of Engineering in Jönköping in the subject area Artificial Intelligence. The work is a part of the Two-year Master of AI Engineering programme. The authors take full responsibility for opinions, conclusions and findings presented.

Examiner:	Lars Carlsson
Supervisor:	Johannes Oetsch
Scope:	30 credits
Date:	2024-06-07

Mailing address:	Visiting address:	Phone:
Box 1026	Gjuterigatan 5	036-10 10 00 (vx)
551 11 Jönköping		

Abstract

This thesis investigates the application of machine learning techniques to predict rotor temperature in induction motors, a crucial parameter for enhancing the safety, efficiency, and longevity of electric machines. It focuses on the development, training, and evaluation of three neural network models: Multilayer Perceptron (MLP), Thermal Neural Network (TNN), and 1-D Convolutional Neural Network (1-D CNN). These models are assessed for their ability to predict rotor temperature using measurable inputs such as rotor speed, slip speed, current, voltage, and cooling conditions, both with and without stator temperature measurements.

Experimental results show that the TNN model, which incorporates heat-transfer principles, outperforms other models in terms of accuracy and reliability. The analysis is based on a comprehensive dataset comprising 2,876,126 data points collected over approximately 82 hours, covering various operational scenarios represented by 87 profile IDs. The models were trained on rotor temperatures ranging from 21.6°C to 259.6°C, with 22°C representing idle room temperature and 250°C the rotor's critical threshold.

The findings suggest that machine learning models offer a more adaptable and precise approach to temperature estimation in electric motors, with significant potential to improve operational efficiency and safety protocols in the automotive and industrial sectors. The TNN model was particularly effective, achieving mean squared errors of 4.2 and 23.3 in two different normal driving scenarios. Additionally, including motor housing temperature data, both inside and outside, significantly enhanced rotor temperature prediction accuracy, potentially reducing costs by eliminating the need for stator temperature sensors.

Data collection and preprocessing were done in collaboration with Volvo Cars. Including motor housing temperature significantly improved rotor temperature estimation accuracy, suggesting cost savings by eliminating the need for stator temperature sensors. This research highlights the superiority of ML models over deterministic models for predicting EM temperature, enhancing safety, efficiency, and reliability. The TNN model's versatility in various driving conditions sets the groundwork for future ML integration into real-time Motor Control Systems.

This research demonstrates the superiority of machine learning models over deterministic models for induction motor temperature prediction, improving safety, efficiency, and reliability. The TNN model's ability to accurately predict rotor temperature under various conditions highlights the potential for integrating machine learning into real-time motor control systems, setting the stage for future advancements in this field.

Keywords Machine Learning, Induction Motors, Motor Control, Multilayer Perceptron, Thermal Neural Networks, 1-D Convolutional Neural Network, Volvo Cars

Sammanfattning

Denna avhandling undersöker tillämpningen av maskininlärningstekniker för att förutsäga rotortemperatur i induktionsmotorer, en avgörande parameter för att förbättra säkerheten, effektiviteten och livslängden hos elektriska maskiner. Den fokuserar på utveckling, träning och utvärdering av tre neurala nätverksmodeller: Multilayer Perceptron (MLP), Thermal Neural Networks (TNN) och 1-D Convolutional Neural Network (1-D CNN). Dessa modeller bedöms för deras förmåga att förutsäga rotortemperatur med hjälp av mätbara ingångar som rotorfart, slipfart, ström, spänning och kylförhållanden, både med och utan statortemperaturmätningar.

Experimentella resultat visar att TNN-modellen, som införlivar värmeöverföringsprinciper, överträffar andra modeller när det gäller noggrannhet och tillförlitlighet. Analysen baseras på en omfattande datamängd bestående av 2 876 126 datapunkter som samlats in under cirka 82 timmar, och som täcker olika operativa scenarier representerade av 87 profil-ID:n. Modellerna tränades på rotortemperaturer som varierade från 21,6°C till 259,6°C, där 20°C representerar rumstemperatur vid tomgång och 250°C rotorens kritiska tröskel.

Resultaten tyder på att maskininlärningsmodeller erbjuder ett mer anpassningsbart och precist tillvägagångssätt för temperaturberäkning i elektriska motorer, med betydande potential att förbättra driftseffektiviteten och säkerhetsprotokollen inom fordons- och industrisektorerna. TNN-modellen var särskilt effektiv och uppnådde medelkvadratiska fel på 4,2 respektive 23,3 i två olika normala körscenarier. Dessutom förbättrade inkluderandet av motorkåpans temperaturdata, både inuti och utanför, avsevärt noggrannheten i rotortemperaturprognoserna, vilket potentiellt minskar kostnaderna genom att eliminera behovet av statortemperatursensorer.

Datainsamling och förbehandling utfördes i samarbete med Volvo Cars. Inkluderingen av motorkåpans temperatur förbättrade avsevärt noggrannheten i rotortemperaturberäkningarna, vilket tyder på kostnadsbesparingar genom att eliminera behovet av statortemperatursensorer. Denna forskning belyser överlägsenheten hos ML-modeller jämfört med deterministiska modeller för att förutsäga EM-temperatur, vilket förbättrar säkerhet, effektivitet och tillförlitlighet. TNN-modellens mångsidighet under olika körförhållanden lägger grunden för framtida ML-integration i realtids motorstyrsystem.

Denna forskning visar överlägsenheten hos maskininlärningsmodeller jämfört med deterministiska modeller för att förutsäga induktionsmotortemperatur, vilket förbättrar säkerhet, effektivitet och tillförlitlighet. TNN-modellens förmåga att exakt förutsäga rotortemperatur under olika förhållanden lyfter fram potentialen för att integrera maskininlärning i realtids motorstyrsystem, och banar väg för framtida framsteg inom detta område.

Nyckelord Maskininlärning, Induktionsmotorer, Motorkontroll, Multilagerperceptron, Termiska neuronnätverk, 1-D Konvolutionellt neuronnätverk, Volvo Cars

Contents

List of Figures

List of Tables

Acronyms

1-D CNN 1-Dimensional Convolutional Neural Network.

AC Alternating Current.

AI Artificial Intelligence.

ANN Artificial Neural Network.

ASIL Automotive Safety Integrity Level.

CAN Controller Area Network.

CNN Convolutional Neural Network.

COD Coefficient of Determination.

CSI Current Source Inverter.

CV Cross-Validation.

DC Direct Current.

DNN Deep Neural Network.

ECU Electronic Control Unit.

EM Personal Computer.

EM Electric Machine.

EWMA Exponentially Weighted Moving Average.

EWMS Exponentially Weighted Moving Standard.

FIT Failure In Time.

HPO Hyperparameter Optimization.

IM Induction Motor.

IOS International Organization for Standardization.

LOCF Last Observation Carried Forward.

LPTN Lumped Parameter Thermal Network.

LSTM Long Short-Term Memory.

MAE Mean Absolute Error.

MAPE Mean Absolute Percentage Error.

MATLAB MATrix LABoratory.

MCS Motor Control System.

MCU Motor Control Unit.

ML Machine Learning.

MLP Multilayer Perceptron.

MSE Mean Squared Error.

ODE Ordinary Differential Equation.

OLS Ordinary Least Squares.

PID Proportional Integral Derivative.

PINN Physics-Informed Neural Network.

PMASM Permanent Magnet ASynchronous Motor.

PMSM Permanent Magnet Synchronous Motor.

PWM Pulse Width Modulation.

RMSE Root Mean Squared Error.

RNN Recurrent Neural Networks.

SMAPE Symmetric Mean Absolute Percentage Error.

TNN Thermal Neural Network.

VSI Voltage Source Inverter.

WMAE Weighted Mean Absolute Error.

1 Introduction

In recent years, the automotive industry has witnessed a significant shift towards electrification and environmentally sustainable energy sources. This change is evident in the growing popularity of electric vehicles and the integration of advanced software systems (Šare et al., 2015; Wolff et al., 2020). The motivation behind this thesis lies in the pressing need to implement machine learning directly on hardware, particularly within the embedded systems of vehicles.

However, this transition poses several challenges, especially concerning the need for accurate and fast computations within embedded systems to control motors and ensure safety (Talaei Khoei et al., 2023). The issue of heat generation in Induction Motor (IM) windings due to overcurrent is particularly concerning, as it can lead to temperature increases that degrade winding insulation and compromise industrial plant safety. Additionally, the computational demands placed on embedded systems underscore the necessity for innovative solutions to address these emerging challenges effectively (Madhavan et al., 2023).

Various machine learning models and techniques have been explored to predict Permanent Magnet ASynchronous Motors (PMASMs) temperature, ranging from complex approaches like Deep Neural Network (DNN), Convolutional Neural Network (CNN), Recurrent Neural Networks (RNN), Thermal Neural Network (TNN), simpler methods such as Ordinary Least Squares (OLS), decision tree-based methods like random forest and support vector regression (Guo et al., 2020; R. Hughes et al., 2023; Kirchgässner et al., 2020, 2021, 2023).

Each approach presents its unique advantages and trade-offs, underscoring the importance of selecting the most appropriate model based on the specific needs of the application.

Introducing TNN as proposed by Kirchgässner et al., 2023, an innovative approach that integrates heat-transfer-based lumped-parameter thermal networks with data-driven nonlinear function approximation using supervised machine learning. Unlike conventional methods reliant on expert knowledge for design, TNN learns directly from data, eliminating the necessity for prior material, geometry, or expert input (Kirchgässner et al., 2023).

In this thesis, the focus is on developing three predictive models: Multilayer Perceptron (MLP), TNN, and 1-Dimensional Convolutional Neural Network (1-D CNN), all aimed at studying how to accurately predict induction rotor temperature.

Through this research, the aim is not only to address the challenges posed by the automotive industry's transition to electrification but also to contribute a solution that improves predictive modeling in embedded systems. By leveraging the unique capabilities of TNN, MLP, and potentially 1-D CNN, and comparing their performance with established approaches, the goal is to achieve significant advancements in predicting rotor temperature, ultimately enhancing safety and efficiency in industrial operations.

1.1 Problem Statement

In the field of industrial automation and machinery, Electric Machines (EMs) are essential in applications ranging from manufacturing to transportation. These motors, however, are susceptible to efficiency losses and potential failures, primarily due to overheating. Traditional methods for monitoring electric motor temperatures are often inadequate in terms of accuracy, adaptability, and predictive capabilities. This inadequacy poses a significant challenge in ensuring operational efficiency, safety, and longevity of Electric Machines (EMs).

The stator temperature can often be measured by a sensor which gives us accurate readings that can be used for other calculations. However, the rotor temperature sensor is much more expensive and is

unfeasible in a practical environment as it would easily break. The temperature difference between the rotor and stator can, for some machines, be significant enough to cause problems, where we cannot simply infer rotor temperature from the stator temperature readings, as this could lead to other parameter estimations becoming statically or dynamically wrong. The machine might also need to be protected with current limitations at higher rotor temperatures even when the temperature is not too high in the stator. The mechanism of calculating the rotor temperature from measurable entities like current, voltages, and stator temperature often involves advanced models that can be difficult to tune. It is also important to handle the dynamics and initial values of estimating rotor temperature where generalized modeling is required to handle all potential motor states.

Existing deterministic mathematical models based on EMs physics face limitations due to sensor inaccuracies, manual calibration requirements, and the complexities of software and hardware implementation. These challenges result in compromised accuracy of motor control and temperature estimation, highlighting the need for more advanced monitoring and control systems aligned with International Organization for Standardization standards.

The transition from traditional internal combustion engine vehicles to electric vehicles, exemplified by Volvo Cars Sweden, further underscores the need for sophisticated motor control software capable of accurately predicting and managing electric motor temperatures in real ime. In scenarios where direct sensor data is unavailable, the software must intelligently estimate critical parameters such as rotor temperature to ensure optimal performance and safety.

To address these challenges, there is a pressing need for advanced monitoring and control systems that leverage Machine Learning (ML) models. ML models offer the potential to adapt to various conditions and provide more accurate predictions and control mechanisms, thereby enhancing the efficiency, safety, and reliability of electric machines in industrial and automotive applications.

1.2 Research Question

In particular, we are addressing the following research questions:

(Q1) Can a ML model be trained to accurately predict rotor temperature for IM using measured quantities such as rotor speed, slip speed, current, voltage, cooling, torque, and without stator temperature?

(Q2) Can this model predict the rotor and stator temperatures for normal driving scenarios, even though the training data did not primarily include such scenarios?

(Q3) Can this ML model replace deterministic models solely based on stator temperature through statistical evaluation? Additionally, How the model be effectively embedded in an electrical inverter's software platform, maintaining accuracy in a live system, and respecting hardware and timing constraints?

1.3 Purpose

Estimating EM parameters is essential for safety and efficiency, aiding in accurate torque calculation and optimizing resource utilization. However, traditional methods face challenges such as noise interference, complex dependencies, lack of efficiency and hardware requirements. This research aims to address these challenges by exploring the potential of ML in parameter estimation and intelligent prediction. Specifically, the focus on rotor temperature estimation in IM serves as a case study, with broader applicability across various domains. By collaborating with Volvo Cars, the research aims to determine if ML can replace traditional state observation methods for EM rotor temperature estimation, thereby enhancing machine protection and operational efficiency.

1.4 Objective

This thesis aims to delve into the realm of ML by developing, training, and implementing MLP, TNN, and 1-D CNN supervised models tailored specifically for estimating rotor temperature. The primary goals encompass creating ML models capable of predicting rotor temperature using measurable inputs, evaluating and comparing their performance, and ultimately selecting the most effective model. This involves the meticulous process of gathering raw data, cleaning it, and identifying the crucial features essential for accurate predictions.

Subsequently, the selected ML model is rigorously tested to ensure seamless real-time predictions without any system failures. The models should outperform traditional methods by demonstrating superior generalization across various motor states. Importantly, these predictions will be instrumental in estimating safety standards as explined in Subsection 2.6.

1.5 Contribution

This thesis makes several contributions to the field of ML applications for EM temperature prediction within the automotive sector, particularly focusing on enhancing the operational efficiency and safety of electric induction motors in Volvo cars:

1. We investigated the enhancement of IM performance amidst a prevailing focus on Permanent Magnet Synchronous Motor (PMSM). Specifically, we studied, designed, trained, and analyzed three distinct neural network models MLP, TNN, and 1-D CNN to predict rotor temperature in IM based on measurable operational parameters such as rotor speed, slip speed, torque, current, and voltage.

   The models were rigorously tested using a comprehensive dataset consisting of approximately 2.9 million data points across various operational conditions. Among these models, the TNN model demonstrated superior performance by incorporating heat-transfer principles into its predictions, offering a reliable and effective solution for temperature estimation without the need for direct stator temperature measurements. It is worth noting that the TNN model also boasts the smallest size and fastest speed compared to the MLP and 1-D CNN models, making it not only accurate but also efficient for real-time applications. Notably, the TNN model exhibited promising results in practical scenarios.

2. In partnership with Volvo Cars, we undertook a extensive process of data collection, cleaning, and preprocessing to ensure the robustness of the models developed. Furthermore, our study delved into the effect of input features on rotor temperature estimation. We discovered that incorporating motor housing temperature, both inside and outside, enhances the accuracy of rotor temperature estimation, especially when omitting stator temperature as an input. This finding not only improves predictive accuracy but also has cost-saving implications for future implementations, as it eliminates the need for stator temperature sensors.

3. This study significantly contributes to improving the safety, efficiency, and reliability of EM by replacing less adaptable deterministic models with flexible and more accurate ML models. The TNN model in this study proved to be versatile, as it could accurately predict rotor temperature under various driving conditions. It was capable of testing rotor temperature based on real driving cycles, This research paves the way for future integration of ML into real-time Motor Control System (MCS).

1.6 Thesis Structure

The thesis begins with an exploration of relevant topics in Chapter 2. This chapter thoroughly examines the automotive mechatronic system, covering the Motor Control Unit (MCU), various motor types, and the functioning of EM. The signal flow of inverters and current transformation mechanisms are also elucidated. Furthermore, factors influencing rotor temperature are discussed in detail.

Chapter 3 outlines the approach adopted for temperature prediction. It details the acquisition and preprocessing of data, including feature engineering techniques, and discusses the MLP, TNN, and 1-D CNN models in depth.

Chapter 4 and Chapter 5 elucidate the experiments conducted to validate the proposed approach. The focus is on determining the accuracy of the models developed in Chapter 3 through different experimental evaluations. It also presents the results obtained from the experimental evaluation. The outcomes are analyzed and interpreted to ascertain the efficacy of the proposed models in predicting rotor temperature. Finally, the thesis concludes with an evaluation of the methods, results, and discussions in Chapter 6.

2 Background and Related Work

The background chapter of this thesis provides an overview of the automotive mechatronic system pertinent to our study. It commences with an exploration of power inverters, elucidating the conversion of Direct Current to Alternating Current. This discussion segues into AC-driven Induction Motors and other motor types. Subsequently, emphasis is placed on motor control. Following this, an exposition on automotive mechatronic systems is provided, incorporating considerations of sustainability and ethical implications. Additionally, statistical learning techniques are introduced. Then we delve in machine learning algorithms and optimizers, pivotal in developing artificially intelligent models capable of predicting intricate non-linear relationships. Furthermore, traditional methodologies for rotor temperature prediction approaches are also addressed for comparison.

2.1 Automotive Mechatronic System

In Figure 1 we show the simplest diagram of the mechatronic system designed for engine control, serving as the central nervous system orchestrating the harmonious interplay among several pivotal components. At its core lies the inverter, assuming an active role in this intricate ensemble.

The inverter, a pivotal component, operates as a power converter facilitating the transformation of Direct Current (DC) into Alternating Current (AC). This conversion is indispensable for the majority of EM reliant on sinusoidal inputs, allowing for nuanced adjustments in both amplitude and frequency to achieve optimal speed control. This feat is accomplished through the utilization of either a line-frequency transformer at the battery level or a switching boost converter capable of voltage modulation, thus enabling the generation of a pseudo-sinusoidal waveform from a square wave DC input. While the resultant output may deviate from ideality, AC motors exhibit remarkable resilience, functioning satisfactorily even when supplied with such non-ideal alternating currents (A. Hughes & Drury, 2019).

Inverters can be categorized into two primary groups based on their DC power source: Voltage Source Inverters (VSIs) and Current Source Inverters (CSIs). VSIs employ shunt capacitors with substantial resistance to mitigate fluctuations in the DC input voltage, while CSIs harness the current source of the DC supply, often employing an extensive array of inductors on the DC-link side for conversion. Presently, VSIs reign supreme as the industry standard owing to the considerable inductor requirements associated with CSIs (Kim, 2017).

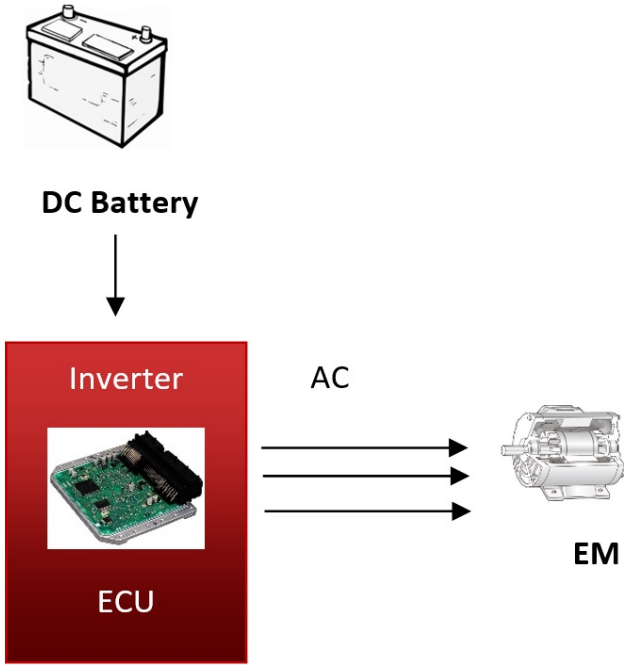
Transitioning from the inverter, the system directs the transformed AC power to the EM, poised to execute its designated tasks with fervor.

Subsequently, we encounter the Electronic Control Unit (ECU), the cerebral cortex of the system. The ECU interfaces with an array of sensors, including speed and temperature sensors, to gather pertinent data. Armed with this information, the ECU orchestrates the symphony of motor performance by generating command signals, thereby dictating the operational parameters of the entire system. Visualize the ECU as the conductor, seamlessly harmonizing the various elements of motor performance into a cohesive whole.

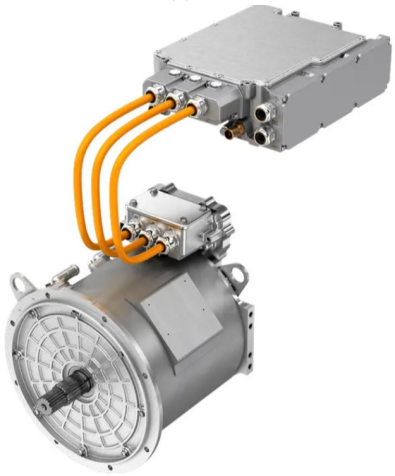
2.1.1 Motor Control Unit

Volvo Cars is using power converters catering to both PMSM and asynchronous IM. This discussion delves into the motor control of an asynchronous IM propelled by a three-phase current, which forms the central focus of this thesis.

[4] https://www.danatm4.com/products/systems/



(a)



(b)

Figure 1. Block diagram of the mechatronic system for engine control (a) and motor and inverter (b)[4].

Enter the MCU, a pivotal component in this orchestration. Tasked with executing commands from the MCU, its primary objective is to regulate the motor's performance. The MCU adeptly manipulates the AC frequency, voltage, and phase emanating from the inverter, ensuring precise control over the motor's behavior, be it smooth acceleration or maintaining a consistent speed.

At the heart of the system lies the star performer, /the EM. While passive in nature, it faithfully executes the instructions imparted by the MCU. With the correct AC parameters provided by the MCU, the motor springs into action, converting electrical energy into mechanical motion. It operates solely on the directives received, devoid of generating its own control signals.

Completing the ensemble is the DC battery, the foundational power source without which this intricate choreography would remain dormant. Acting as the catalyst, the battery initiates the entire process by

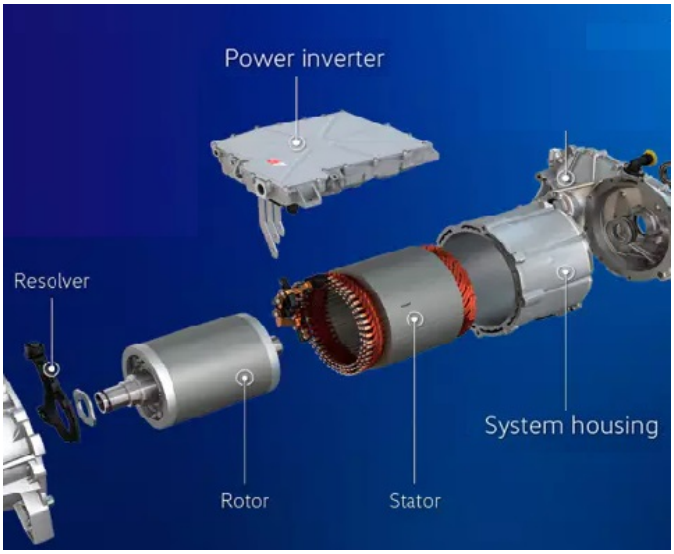


Figure 2. PSM engine from Volkswagen as used on the rear axle of the GTX[5].

furnishing the requisite voltage to the inverter.

Facilitating communication among these components is the Controller Area Network (CAN) communication protocol a simple two-wire differential serial bus system pioneered by Bosch in the early 1980s (Specification, 1991).

2.1.2 Motor Types

Within the realm of motors, a plethora of types exists, yet we shall primarily focus on introducing two commonly employed varieties within the industrial landscape: the PMSM and the IM, with the latter being particularly pertinent to the scope of this thesis.

Permanent Magnet Synchronous Motor: Diverging from the operational mechanism of an IM, which relies on induced currents to engender rotating magnetic fields, a PMSM harnesses the presence of a permanent magnet embedded within its rotor, obviating the need for field windings as shown in Figure 2. This design choice offers notable advantages, including heightened power density and enhanced dynamic responsiveness. However, it is worth noting that PMSM typically entail a higher cost compared to IMs. Furthermore, PMSM are further categorized into radial-flux and axial-flux variants, contingent upon whether the rotor rotates within or beside the stator, respectively (Kim, 2017).

Induction Motor: Figure 3 shows an asynchronous motor, an IM which operates utilizing either single-phase or three-phase alternating AC. Its fundamental structure comprises a stationary stator core and a rotor, the latter being the movable component positioned within the stator without direct contact. The region between the stator and rotor is referred to as the air-gap. Within the stator's iron core are slots housing windings, forming an electrical circuit. Upon the flow of current through the stator windings, a magnetic field is induced around the stator, subsequently inducing a current in the rotor, thus generating a magnetic field within it hence the name "induction motor." The interplay of axial currents in both the stator and rotor is pivotal in generating torque (A. Hughes & Drury, 2019).

[5] https://www.heise.de/hintergrund/Basic-technology-in-the-electric-car-types-of-three-phase-motor-7452474.html?seite=2
[6] https://www.heise.de/hintergrund/Basic-technology-in-the-electric-car-types-of-three-phase-motor-7452474.html?seite=2

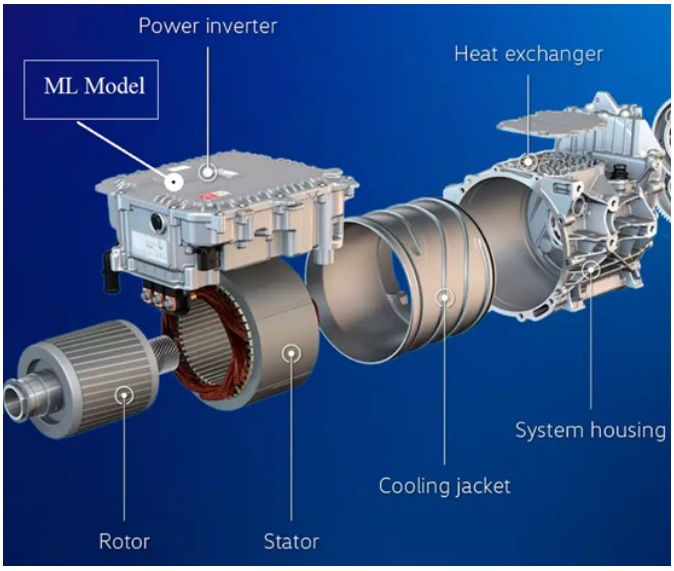


Figure 3. Asynchronous motor, as Volkswagen builds on the front axle, for example[6].

2.1.3 Functioning of an Electric Motor

The concepts discussed are drawn from two primary sources. Firstly, Electric Motor Control (Kim, 2017) provides a comprehensive explanation of the mathematics underlying inverters and IM , accompanied by insightful figures. Secondly, Electric Motors and Drives: Fundamentals, Types, and Applications (A. Hughes & Drury, 2019) offers a detailed exploration of principles.

Rotor and Stator: The EM comprises two primary components: the rotor, responsible for rotation, and the stator, which remains stationary. Their interaction is pivotal, as it induces a rotating magnetic field essential for the motor's functionality.

Speed and Torque: The motor's rotational speed correlates directly with the frequency of the electrical power supplied to it. Higher frequencies result in faster rotation. Torque denotes the rotational force exerted by the motor. The relationship between speed and torque varies depending on the motor's design and the applied load. Often, increasing the load may lead to a decrease in speed.

Slip Speed: It refers to the disparity between the synchronous speed (the speed of the rotating magnetic field) and the actual speed of the rotor. Ideally, in a motor devoid of slip, the rotor would match the magnetic field's speed. However, in reality, some slip is inevitable.

The Current: The I_q and I_d supplied to the motor encompasses two distinct components: the quadrature (q) component and the direct (d) component. These components are integral to Park's transformation[7], a mathematical representation employed in motor system control. The q component aligns with the rotor flux's magnetic field, while the d component is orthogonal to the rotor flux's magnetic field.

[7] https://se.mathworks.com/solutions/electrification/clarke-and-park-transforms.html

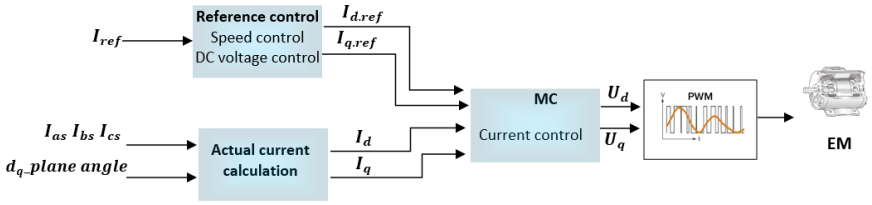


Figure 4. Signal flow of inverter.

Voltage: Input determines the magnitude of the magnetic fields generated by the stator. Ohm's Law ($V = IR$) elucidates the relationship between voltage, current, and resistance.

Coolant and Rotor Temperature: Many motors, particularly those in industrial settings, incorporate cooling systems to dissipate heat generated during operation. Monitoring rotor temperature is crucial, as excessive heat can compromise insulation and other components, thereby affecting the motor's performance and longevity.

2.1.4 Signal Flow of Inverter

Figure 4 illustrates the simplest block diagram of fundamental signal flow within the inverter[8]. The torque, as defined in Equations 1 and 2, is contingent upon the current. Hence, when the ECU mandates a specific torque, it actually transmits a reference current, (I_ref), to the MCS of the inverter, as depicted in Figure 4

These reference values undergo processing within a reference control stage, which incorporates considerations for speed and DC voltage control. Here, new reference currents, (I_dref) and (I_qref) , are computed. Leveraging knowledge of the rotor's angle relative to the stator and the phase currents (I_as, I_bs, and I_cs), the actual values of (I_d and I_q) are derived by transforming the three-phase system into a two-phase system using current transformation techniques.

Subsequently, a Proportional Integral Derivative (PID) controller matches the reference currents with the actual currents, generating new values for I_d and I_q and minimizing the error towards I_dref and I_qref. These currents are then converted into voltages, (U_d and U_q), which are utilized in Pulse Width Modulation (PWM) control.

Upon completion of modulation, a new optimized three-phase system is attained by employing the same current transformation as earlier, albeit in reverse order. This intricate process ensures precise control over the motor's operation, facilitating efficient performance as per the ECU's directives.

2.1.5 Current Transformation

In order to effectively control the three-phase system, it becomes necessary to simplify the control process. This is achieved through the utilization of a $d-q$ current regulator, wherein we mathematically transform the three-phase system into a two-phase system and vice versa. This transformation involves the establishment of both a stationary frame and a rotating reference frame. The stationary frame features axes labeled α, positioned at $\theta = 0$, and β, situated at a 90-degree angle relative to α. Meanwhile, the

[8] https://se.mathworks.com/solutions/electrification/field-oriented-control.html

rotating reference frame encompasses axes labeled d (the reference axis) and q, positioned at a 90-degree angle ahead of d. This framework enables the expression of the three currents I_{as} and I_{bs}, and I_c pertaining to each respective phase, in terms of two currents, I_α and I_β. The angular relationship between the reference frame and the stationary frame varies over time and is contingent upon the angular velocity, ω, of the reference frame, expressed as $\theta = \omega t$. This dynamic relationship facilitates the derivation of a rotor coordinate system from the two-phase system, as depicted in the following equation:

$$I_d = I_\alpha \cos(\theta) + I_\beta \sin(\theta) \tag{1}$$

$$I_q = I_\alpha - \sin(\theta) + I_\beta \cos(\theta) \tag{2}$$

2.1.6 Factors Influencing Rotor Temperature

The generation of output torque within the motor is heavily influenced by the losses incurred in the stator iron and copper windings, which collectively produce air-gap power. Additionally, rotor copper loss contributes to these losses, stemming from the rotor's inherent resistance, which is influenced by the presence of rotor harmonic currents at any given time. Notably, the rotor resistance is subject to variation based on the operating switching frequencies of the inverter, attributable to the skin effect phenomenon.

Skin effect, a well-known phenomenon in electromagnetism, manifests in alternating electric AC by concentrating current density predominantly near the surface of a conductor, with diminishing density as depth increases. Consequently, a significant portion of electric current flows along the outer surface skin of the conductor (Lamb, 1883). While PWM techniques can help mitigate these losses, it's crucial to note that harmonic losses are unevenly distributed throughout the motor, primarily affecting the rotor. This asymmetry can elevate rotor temperature, exacerbating rotor resistance within enclosed spaces, potentially surpassing stator resistance (A. Hughes & Drury, 2019).

The induction of current in the rotor windings is instigated by the rotating stator magnetic field, prompting the rotor to counteract changes in rotor-winding currents by aligning with the stator magnetic field. As a result, the rotor accelerates until induced rotor current and torque achieve equilibrium with the load on the rotor. Temperature escalation is predominantly attributed to the flow of current through the conductor, influenced by factors such as thermal capacity in watts, thermal resistance, and rotor slip.

In instances where the stator winding experiences anomalies such as short circuits or grounding between turns or phases, there is a likelihood of increased current and temperature. Remedial measures typically involve reinforcing insulation in the affected area or replacing the entire winding to ensure optimal motor performance and safety.

2.2 Parameter Estimation using Traditional Methods

Estimating rotor temperature poses challenges due to the intricate relationships among motor properties, necessitating a deterministic mathematical function for expression. Traditional methods for rotor temperature estimation without sensors can be categorized into thermal models, active parameter estimation, and state observers.

Thermal Models: Employ mathematical representations or simulations of thermal behavior over time, incorporating parameters like heat generation, conduction, convection, and radiation. For instance, The

heat transfer coefficient coupled with a transient thermal model is utilized to ascertain temperature rise in stator windings by Odvářka et al., 2010.

Active Parameter Estimation As demonstrated by Iandola et al., 2016, involves injecting high-frequency signals into a surface PMSM to induce excitation in stator and rotor impedances, enabling direct inference of permanent magnet temperature.

State Observers: State observers also known as state estimators, leverage measurable quantities such as current and voltage measurements to estimate variables that are challenging to directly measure, such as rotor flux or magnetic field. A flux observer is employed to estimate demagnetization in an interior PMSM, exploiting the temperature dependence of permanent magnet flux to infer rotor temperature (Specht & Böcker, 2010).

While traditional models offer speed and determinism, simplifying implementation, they necessitate the establishment of complex physical and mathematical relationships, making tuning challenging. Moreover, they may fail to adapt to dynamic environments and may require additional components. Signal injection-based methods can be impractical due to their reliance on extra components and complexity.

Contrary to traditional methods, a contemporary approach employs ML for parameter estimation, utilizing measurable quantities for inference. ML models excel in generalization and adaptability, enabling easy addition or removal of parameters for retraining without extensive tuning. They can capture complex, nonlinear relationships that traditional equations struggle to describe.

However, ML models require training on qualitative data, which can be time-consuming and prone to noisy or faulty data. Furthermore, inference time and integration complexity are elevated, when compared to traditional methods due to the advanced nature of ML models. Despite these drawbacks, ML offers a promising avenue for parameter estimation, particularly in scenarios where traditional methods fall short. We elaborate on methods of ML in the next section.

2.3 Statistical Learning

Statistical learning necessitates a grasp of the underlying mathematical connections and characteristics of the dataset before delving into ML model training or predictive analysis. This section elucidates the mathematical principles underpinning statistical concepts, encompassing feature correlations, standardization, accuracy quantification, regression, and t-tests.

2.3.1 Correlation

Correlation measures the degree to which two variables are associated and change together (Altares, 2003). Figure 5 illustrates how two variables can be correlated with one another. The relationship between two variables can be quantified with a numerical value ranging from -1 to 1. Given two variables x_i, y_i and a correlation function $corr(x_i, y_i)$:

$$corr(x_i, y_i) = \begin{cases} -1, & \text{a perfect negative or anticorrelation} \\ 0, & \text{no correlation} \\ 1, & \text{a perfect positive correlation} \end{cases}$$

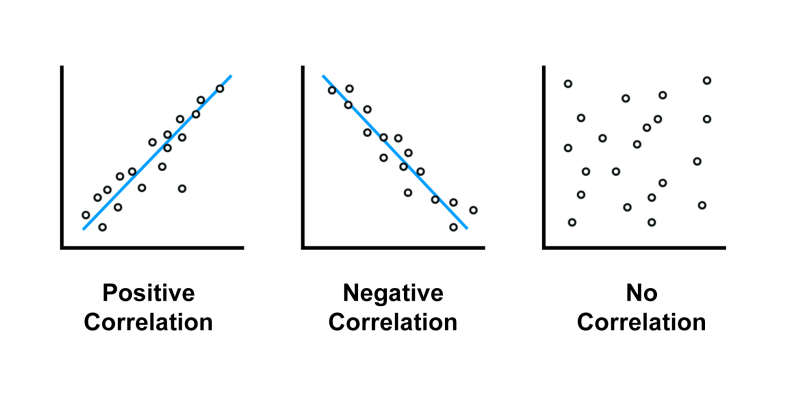


Figure 5. Example depiction of correlation.

In a positive correlation, the relationship between two variables is such that they can be seen as moving in the same direction. As the value of one variable increases, the other variable also increases. Similarly, as one variable decreases then so will the other.

In a negative correlation, there is an inverse relationship between the two variables. In this case, as the value of one variable decreases the value of the other variable tends to increase.

If there is no correlation between two variables, then it can be said that any change in the value of one variable does not affect the other variable.

Indeed, in statistics and machine learning, discovering correlations between one or more variables is a key step in making accurate predictions (Riddiough and Thomas, 1998).

While there are several ways of calculating the correlation of two variables, the most familiar one is *Pearson's correlation coefficient*, defined in Equation 3:

$$\text{corr}(y,\hat{y}) = \frac{\sum_{i=1}^{N}((y_i-\bar{y})(\hat{y}_i-\bar{\hat{y}}))}{\sqrt{\sum_{i=1}^{N}(y_i-\bar{y})^2}\sqrt{\sum_{i=1}^{N}(\hat{y}_i-\bar{\hat{y}})^2}} \tag{3}$$

where N is the total number of data points, y_i are individual data points, $\hat{y}$ are the dependent data points, $\bar{y}$ is the average value of the data points, $\bar{\hat{y}}$ is the average value of the dependent data points.

2.3.2 Regression

Regression is a tool used to estimate continuous values of variables such as income, prices, velocity, and temperatures. In statistical machine learning, using regression is one of the most important tools for estimating the relationship between independent and dependent variables. These variables can display linear or non-linear relationships (correlations) and therefore might require a different method of regression analysis. When it comes to selecting the right features for the model, it is useful to know how strongly the individual variables are correlated to the dependent variable you are trying to predict (Riddiough and Thomas, 1998). For example, from the section on motor types, we know that the induction motor utilizes AC, which follows a sinusoidal curve. If there is a relation between the current supplied to the motor and a temperature rise, it would have to be nonlinear. However, a limitation is that the correlation of features to certain outcomes does not guarantee that the feature is causally related to those outcomes (Simoes et al., 2024).

Statistical ML deals with the statistical inference problem of finding a predictive function based on the

given data. Statistical learning theory has many useful applications in fields such as computer vision, speech recognition, and forecasting problems. Nonetheless, a major problem that arises in ML is that of overfitting. Overfitting is an undesirable result of a model and is when the model can accurately predict the given dataset but does not do well on unseen data. Because learning is a prediction problem, the goal is not to find a function that most closely fits the data but to find one that will most accurately predict output from future input. This problem of doing well on the training set without overfitting is also known as the bias-variance tradeoff. While there are many techniques to tackle this issue, this means that models can never be perfect and will always make some errors (Kumar, 2019).

However, in essence, the task of ML is to create a function that given a set of inputs will correctly map to a possibly unknown and correct output. Among the currently best-known estimators of such a function is the artificial neural network and its variants.

2.4 Machine Learning

ML encompasses various forms, notably classification tasks aimed at categorizing discrete class labels, and regression tasks focused on predicting continuous values. Supervised learning involves training models with labeled data, while unsupervised learning entails discovering patterns and associations in unlabeled data. Reinforcement learning utilizes training feedback coupled with a reward mechanism (Goodfellow et al., 2016). Additionally, there exists a multitude of model types to select from, contingent upon the data structure. This chapter primarily delves into the regression and supervised learning aspects of Artificial Neural Network (ANN), alongside fundamental concepts in ML.

2.4.1 Artifical Neural Network

The ANN's roots are biologically inspired by the human brain. It mimics the connections between neurons allowing a network of interconnected artificial neurons to perform tasks such as regression, classification, and clustering. They can learn and model from data otherwise too complex for humans to find patterns by hand alone. We will present a short introduction to the topic. To the avid readers, we recommend the work by Nielsen (Nielsen, 2015) for more in-depth information.

ANN generally are comprised of an input layer, one or more hidden layers, and an output layer. Each layer is made of individual neurons that can store a number and each neuron is connected to other neurons in the next layer. The edges are typically made of two additional parameters called weights and biases which after passing through an activation function simulate the firing of a neuron (whether a neuron sends a signal) to the next neuron. The process of a neural network learning is primarily through the adjustment of these weights and biases in response to the input it receives. This is done during the training phase where the network is fed a large dataset. The goal of training is to minimize the difference between the actual output of the network and the expected output, known as the error or loss. This is typically achieved through an optimization algorithm, with backpropagation being one of the most common methods used for training neural networks.

Backpropagation involves two main phases: the forward pass and the backward pass. During the forward pass, input data is passed through the network, layer by layer, until it reaches the output layer. The output of the network is then compared to the expected output, and the difference between them is calculated to measure the error. During the backward pass, this error is propagated back through the network, layer by layer, and the weights and biases are adjusted in a way that minimizes the error. This adjustment is usually done using an optimization technique such as gradient descent.

The activation functions play a crucial role in the network's ability to capture non-linear relationships.

Without these functions, the network would not be able to model the complex patterns often found in real-world data. Commonly used activation functions include the sigmoid, tanh, ReLU (Rectified Linear Unit), and softmax for the output layer in classification tasks.

As the network trains, it gradually improves its predictions, learning the patterns and relationships within the data. Once the training process is complete, the network can then be used to make predictions on new, unseen data. It is this ability to generalize from the training data to new instances that makes neural networks so powerful for tasks like image recognition, natural language processing, and many other applications.

Nowadays, the development and implementation of neural networks have been facilitated by advancements in computing power and the availability of large datasets. Additionally, frameworks such as Tensorflow[9] and PyTorch[10] have made it easier for researchers and practitioners to build and experiment with neural networks, pushing the boundaries of what these models can achieve.

However, neural networks are not without their challenges. Training ANN effectively usually requires large amounts of data and can be computationally intensive. Furthermore, the internal workings of a trained network can be difficult to interpret, commonly referred to as the "black box" problem. Despite these issues, research and development in neural network architectures, training methodologies, and applications continue to push forward the field of Artificial Intelligence (AI).

2.4.2 Evaluation Metrics

Evaluation metrics provide us with a form of measuring how well our model performs. Each metric gives a score that can be compared with other similar models (Goodfellow et al., 2016). To determine a model's accuracy for temperature prediction, the following common metrics will be utilized and are defined as follows:

Mean Absolute Error (MAE): The MAE measures the average magnitude of the errors between the ground truth and the predicted values. A mean absolute error score of 0 indicates that the model has made no errors in its predictions. The smaller the value is, the better.

$$\mathrm{MAE}(y,\hat{y}) = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{N} \tag{4}$$

Mean Squared Error (MSE): The MSE measures the average of the squares of the errors. It is also known as the mean squared deviation (MSD).

$$\mathrm{MSE}(y,\hat{y}) = \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N} \tag{5}$$

Root Mean Squared Error (RMSE): Taking the square root of the MSE gives the RMSE. It measures the average difference between the true value and the predicted value.

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

R^2/ Coefficient of Determination (COD): The COD ranges from $(-\infty, 1]$ and is a measure of the proportion of variance in y explainable by the independent variables (features). Having a score of 1 means

[9] https://www.tensorflow.org/
[10] https://pytorch.org/

100% of the variation is explained by the input features.

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

While the chosen metrics (MAE, MSE, RMSE, and COD) provide a comprehensive evaluation of model performance in general scenarios, other metrics were considered but not utilized for specific reasons. The Weighted Mean Absolute Error (WMAE) assigns different weights to errors based on the temperature range, but since no temperature ranges are more critical in our context, it was omitted. Mean Absolute Percentage Error (MAPE) focuses on the relative size of the error, which is less relevant when the absolute size of the error is more important for our application. Similarly, Symmetric Mean Absolute Percentage Error (SMAPE) was not used for the same reason. These decisions ensure that our evaluation metrics align with the goal of capturing overall model accuracy rather than emphasizing specific ranges or relative errors.

2.5 Related Work

Guo et al., 2020 introduced an innovative DNN framework aimed at accurately forecasting stator winding temperature in the PMSM. They utilized Support Vector Regression to capture the intricate non-linear correlations between input parameters and temperature. While decision tree-based methods offer interpretability and non-linear handling capabilities, they might fall short in delivering the requisite predictive precision for crucial scenarios. To address this, ensemble learning techniques like Random Forest and AdaBoosting were investigated as well for temperature prognosis in the PMSM. The DNN model proposed by Guo et al., 2020 exhibits substantial enhancements in stator winding temperature prediction accuracy for the PMSM. Its effectiveness in capturing temperature dynamics is evidenced by robust coefficient of determination (R-squared) values and minimal mean absolute percentage error. Future investigations could delve into the model's adaptability across diverse motor configurations, operational settings, and its applicability to other motor variants. This work has also several limitations. First, it heavily relies on high-quality training data, which may not always be available. Second, interpreting the model's internal workings can be challenging due to its complexity. Additionally, deployment in real-world motor systems requires careful integration and validation

Ullah et al., 2020 explore various deep learning techniques to enhance fault detection efficiency and accuracy. Their proposed methodology aims to tackle challenges arising from limited and non-uniform motor data, as well as the complexities associated with deep learning methods. Beyond internal assessment, the evaluation extends to benchmarking against existing procedures, providing context for the obtained results. However, a potential drawback of the methodology is its focus on specific fault types, notably irreversible-demagnetization fault and bearing fault, which might limit its applicability to other fault types present in the PMSM.

However, more advanced neural network architectures further improved prediction accuracy to levels suitable for commercial applications. For instance, the CNN was employed by Kirchgässner et al., 2020 with promising results approaching the accuracy of Lumped Parameter Thermal Networks (LPTNs) (Wallscheid & Böcker, 2015). Nonetheless, a drawback is that the CNN entails numerous internal weights and requires extensive tuning time, as demonstrated in recent studies (Kirchgässner et al., 2020), with inference times significantly higher than those of OLS models (Kirchgässner et al., 2021).

Kirchgässner et al., 2021 conducted a study on data driven PMSM temperature estimation using supervised ML techniques. The objective was to develop accurate models with intermediate complexity while ensuring computational efficiency, particularly focusing on estimating the temperature of permanent magnets in the PMSM. Various types of neural networks including RNN, MLP, and CNN were trained and evaluated. CNN emerged as the most accurate among the neural network models, closely followed by the

statistically based OLS model. Conversely, models such as K-Nearest Neighbors, Random Forest, and Support Vector Regression exhibited poor performance despite hyperparameter optimization and model capacity adjustments (Al-Gabalawy et al., 2024; R. Hughes et al., 2023; Yadav, Prabhakaran, et al., 2023).

Analyzing the results further, it becomes apparent which models might be suitable for integration into Volvo Cars' software inverter. While the studies did not specify the hardware or timing requirements, it emphasizes the necessity of a time-critical system for inverter logic and model inference. Given this constraint, large models are impractical. Although CNN demonstrated superior performance, its computational speed was approximately eight times slower than MLP (Kirchgässner et al., 2021). Additionally, k-NN, RF, and SVR can be deprioritized due to their larger model sizes and inferior accuracy compared to other models (Kirchgässner et al., 2021). The study also evaluated a non-ML thermal model, LPTN, which exhibited good accuracy and model size but was approximately four times slower than MLP. LPTN are an alternative modeling approach used in various industries to describe the thermal behavior of systems (Kirchgässner et al., 2023).

LPTN are a system of Ordinary Differential Equations (ODEs) that represent the thermal properties of a system using discrete elements. These networks concentrate the thermal characteristics into specific components, similar to how electrical circuits use discrete elements such as resistors, capacitors, and inductors (Kirchgässner et al., 2021). Wallscheid and Böcker, 2015 showcase how parameters of LPTN can be fine-tuned using particle swarm optimization, resulting in one of the most precise models according to a comparison conducted by Kirchgässner et al., 2023.

A in particular promising variation of machine learning architectures discussed is the TNN, which previously demonstrated superior performance among all modeling techniques by integrating neural networks to combine the concepts of LPTN and universal differential equations (Kirchgässner et al., 2023). The thermodynamically inspired principles of LPTN, which estimate losses to incorporate heat generation, combined with the non-linearities captured by neural networks, present a robust modeling architecture for real-time thermal modeling.

R. Hughes et al., 2023 present an innovative ML-based modeling approach that integrates advanced feature engineering grounded in physical principles. ML algorithms, specifically OLS and MLP, have demonstrated promise in capturing intricate data relationships. OLS provides a linear mapping between input features and output targets, while MLP's capacity to learn non-linear correlations renders it suitable for capturing complex temperature dynamics within electric machines. Moreover, the utilization of empirical data to parameterize computationally efficient models, such as Linear Parameter Varying Transfer Functions and machine learning architectures, underscores the approach's practicality in real-time electric equipment temperature monitoring. However, a potential drawback of the proposed methodology is overfitting, particularly when employing complex machine learning models like the MLP (Yadav, Prabhakaran, et al., 2023). Given the thesis's focus on ML, traditional methods such as the LPTN were not investigated. The thesis primarily explores ML techniques, excluding traditional methods like LPTN. Instead, the focus was on MLP, CNN, and TNN.

2.6 Safety Compliance

In automotive engineering, adherence to safety standards is crucial, as outlined by International Organization for Standardization (IOS) safety guidelines, including those specified by Automotive Safety Integrity Level (ASIL) and Failure In Time (FIT) values ("ISO 26262-1:2018", 2018). For electrical systems integrated into mass-produced vehicles, adherence to IOS 26262 standards is imperative. This standard, established by the IOS, serves as a pivotal framework ensuring functional safety in road vehicles ("ISO 26262-1:2018", 2018). The focus of this thesis aligns with the objectives of the United Nations' sustainable development goals, particularly goal 9, which emphasizes advancements in industry, innovation, and infrastructure (United Nations, n.d.).

The IOS 26262 standard encompasses various safety phases and levels, including functional safety, product development, and design, delineated across hardware and software requirements. Of particular importance is the conceptual phase, which involves hazard analysis and risk assessment conducted by the vehicle manufacturer (“ISO 26262-1:2018”, 2018).

2.6.1 Automotive Safety Integrity Level (ASIL)

Every potential hazard is carefully evaluated and categorized into ASIL levels, ranging from A to D, with A representing the most critical and D the least. Achieving a higher ASIL requires the mitigation of hazards at lower levels first. For instance, to attain ASIL B, all hazards at level A must be addressed before mitigating those at level B. Determining the ASIL of a hazard involves analyzing a hazard matrix that accounts for various potential accidents. The ASIL assignment is based on the risk associated with each hazard, typically calculated as the product of the expected loss in case of an accident and the probability of the accident occurring (“ISO 26262-1:2018”, 2018).

Quantifying the risk of hazards often involves utilizing a metric known as FIT. FIT measures the likelihood of a signal providing an incorrect response due to a failure within a specified timeframe. For instance, in the context of determining torque based on a speed signal from a motor sensor, if the sensor malfunctions once per hour, leading to an inaccurate speed reading, this incident contributes to the FIT value of that signal. FIT is crucial because even slight inaccuracies in signals can lead to safety estimations that are insufficiently mitigated, potentially resulting in accidents.

2.7 Sustainability and Ethical Considerations

In the development of software for power inverters supplying electrical motors, several sustainability and ethical factors merit consideration. The impact of the software on power consumption and memory usage is paramount, as it directly influences energy efficiency and equipment longevity. Prioritizing these aspects can minimize energy wastage and reduce wear on the machinery. Furthermore, adherence to safety guidelines is imperative to mitigate risks to individuals operating vehicles on the road. Implementing features such as safe torque alerts and motor trip mechanisms can effectively prevent potential hazards. Advancing towards a sustainable future involves reducing carbon emissions, wherein transitioning to eco-friendly alternatives like electric vehicles plays a pivotal role. Enhancing inverter software not only contributes to improving energy efficiency but also facilitates the electrification of the automotive industry, aligning with sustainability objectives.

3 Approach for Temperature Prediction

In this critical section, we explore the innovative methodologies employed to predict the temperature of induction motor rotors, a key aspect of maintaining operational efficiency and safety in electric motors. The chapter outlines the systematic approach used to collect, preprocess, and analyze data, alongside a detailed description of the machine learning models developed for this purpose. Emphasizing the importance of accurate temperature predictions, this section delves into the selection and optimization of neural networks tailored for this task, including MLP, TNN, and 1-D CNN. Through a combination of theoretical underpinnings and practical application, this approach not only enhances the understanding of motor temperature dynamics but also sets a new benchmark for predictive accuracy in real-world applications.

3.1 Research Process

In this section, the methodology for the development of estimating the rotor temperature of an induction motor will be described, as depicted in Figure 6. It includes data analysis, a detailed explanation of how the model was selected, and a description of the experiments conducted.

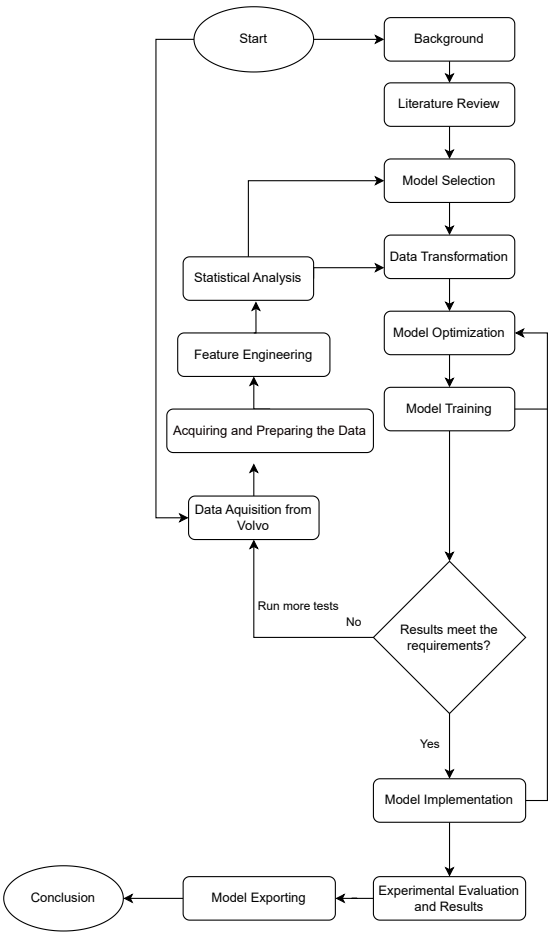


Figure 6. A flowchart depicting the research process.

3.2 Acquiring and Preparing the Data

The datasets we gathered from Volvo comprise of MATrix LABoratory (MATLAB) files of different recorded sessions that exhibit varying driving profiles. The datasets also vary in what features were recorded. There are over 360 feature columns that may include the current and voltage in the q-d coordinate system, the rotor temperature, the torque, and the motor speed. Each row in the dataset captures 0.1 seconds in time.

The **q-d coordinate system** is also known as the Park's transformation. It is a useful mathematical tool for simplifying the analysis of three-phase electrical machines by transforming three-phase variables into two orthogonal components. This transformation effectively converts a system that varies with both time and position into a system that varies only with time, making the analysis and control of AC machines more straightforward (Vittal et al., 2019).

The **direct axis (d)** is aligned with the rotor's magnetic field. Where, in the case of synchronous machines, the **d** component represents the flux-producing component of currents and voltages. The **quadrature axis (q)** leads the direct axis by 90 degrees in the direction of rotation. The **q** component typically influences the torque production.

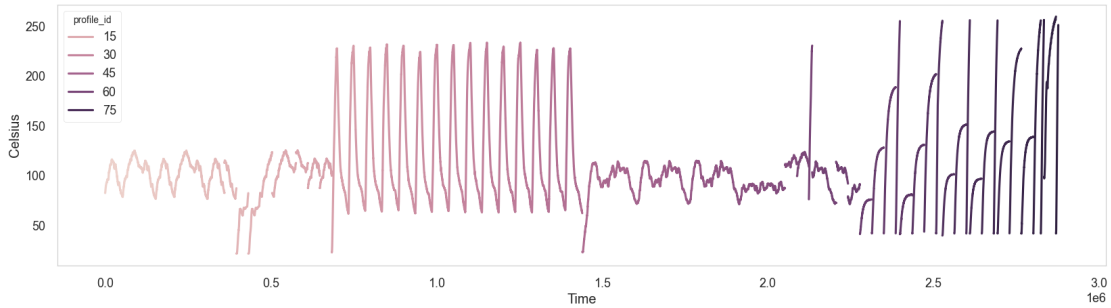
The datasets revealed various types of artefacts and needed to be cleaned first. There were anomalies such as the rotor temperature jumping to $1000^{\circ}C$ at the end of certain captures, which could be attributed to sensor noise and does not reflect the physical state of the rotor. In addition, the dataset also contained noises such as sporadic spikes. For example, the rotor temperature would be gradually increasing, but all of a sudden show a large temperature spike of roughly $20^{\circ}C$ every few seconds. Furthermore, the dataset also contained a small amount of missing values for the current and voltage.

The importance of having data on the current and voltage supply resulted in us using a smaller subset of the initial total dataset. After cleaning, there remained around 20 hours of training data to work with, which would later prove to be insufficient. The remaining usable data capture was also primarily focused on high-speed runs and contained little distinguishable patterns for the models to learn from. As a result, a requisition was made to acquire more data with mixed patterns and varying profiles. The new batch contained roughly 60 hours and was merged with the previous dataset for a total of 80 hours, as shown in Figure 7 and in Appendix A.1. The new dataset contained some profiles based on repetetive driving cycles while some were just based on some random driving cases. There are also several hours of testing with inputs that could vary randomly (accelerating and breaking).

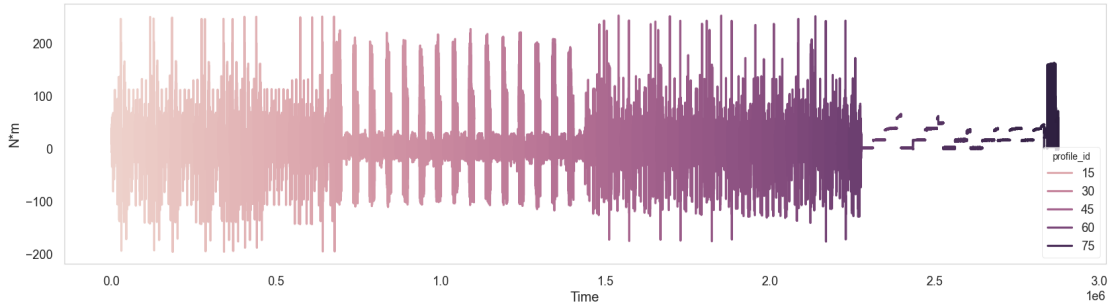
The new measurement data contains additional features not present in the old measurement data, including motor housing temperature in/out, in locations 72 and 73 as depicted in Figure 8, as well as coolant out temperature. However, these three features were excluded from the full concatenated dataset due to their unavailability in the older, high-speed measurement data. A significant aspect of our work involved investigating the impact of incorporating these additional features into our training model and assessing their effect on rotor temperature estimation. To accomplish this, we exclusively utilized the new measurement data and compared the results with those obtained using the full dataset. Furthermore, we sought to evaluate the influence of excluding the stator feature from the input and instead estimating both rotor and stator temperatures, with and without the inclusion of motor housing temperature in/out and coolant out temperature. Section 5 presents the results and evaluation of our models, examining whether these three features enhance rotor temperature estimation when stator temperature is not utilized as an input feature.

3.2.1 Choice of Input Features

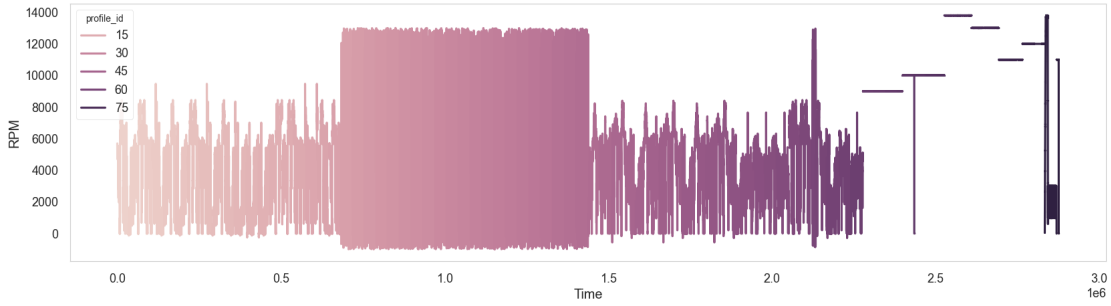
To meet ASIL-C requirements as per ISO 26262 2.6, our choice of parameters for rotor temperature estimation is restricted to a specific set deemed safe. The safety evaluation of each parameter is contingent



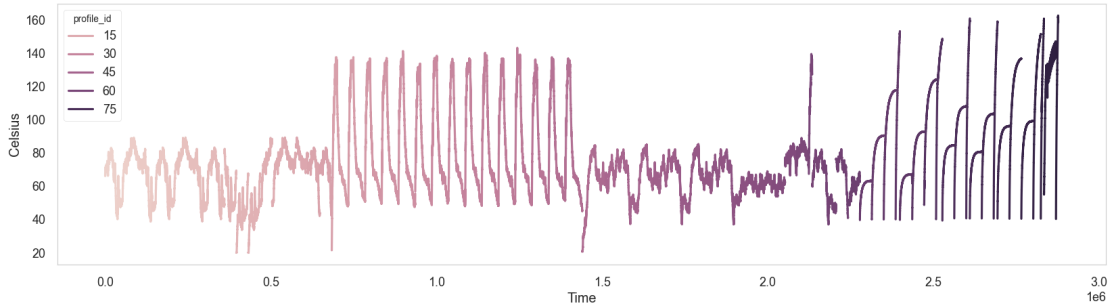
(a)



(b)



(c)



(d)

Figure 7. A plot showing the (a) Rotor Temperature (b) Torque (c) Speed (d) Stator Temperature. Each time column represents 0.1 seconds.

upon its FIT value 2.6.1. Any parameter failing to meet this safety criterion cannot be utilized for inferring rotor temperature. The set of safe parameters suitable for input features related to IM motors includes:

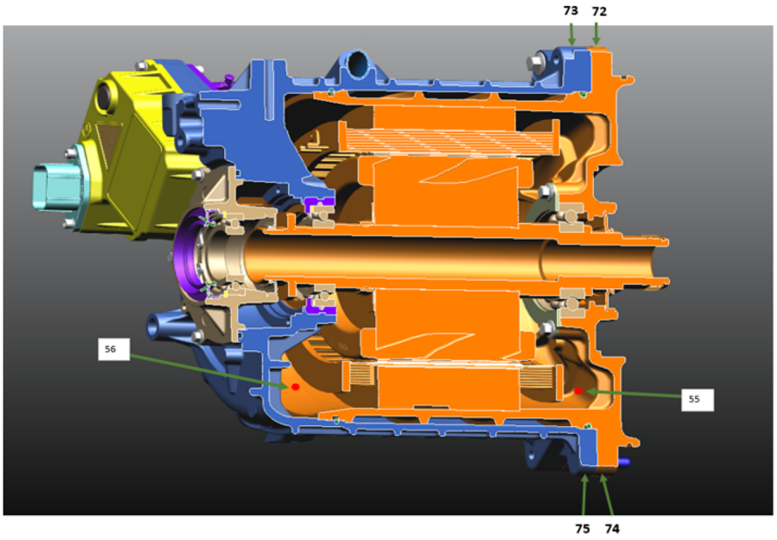


Figure 8. The cross section of the induction motor depicting housing and specific numbered sensor locations.

- Coolant Temperature,
- Rotor Temperature,
- Motor Speed,
- Torque,
- Stator Temperature,
- Slip Frequency,
- Voltage (U) in $q-d$ coordinate system,
- Current (I) in $q-d$ coordinate system,
- Winding Temperature,
- Ambient Air Temperature, and
- Motor Housing Temperature in/out.

Furthermore, we can leverage Pearson's correlation coefficient to examine the relationships between parameters efficiently as shown in Figure 9 for the old measurement and Appendix A.2 for the new measurement. This analysis primarily focuses on identifying correlations between input features and the target output, i.e., rotor temperature. Parameters with higher correlation coefficients are expected to perform better as input features. Table 1 provides a summary statistic of the complete dataset.

While considering additional parameters in the future, it is crucial to be mindful of the curse of dimensionality. At present, dealing with (13–15) parameters keeps the feature space manageable, negating the need to eliminate any features. Therefore, multiple combinations of input features have not been evaluated due to time constraints, as it would introduce additional dimensions to hyperparameter tuning.

Figure 9. A heatmap showing the correlation of the input features for the old and new measurements.

Table 1

Summary statistics of the complete dataset.

Feature	**Mean**	**Std**	**Min**	**Max**
Coolant Temperature	26.52	10.79	6.03	49.44
Rotor Temperature	112.65	43.07	21.58	259.62
Motor Speed	6046.95	3617.36	-1008.59	13804.10
Torque	12.97	29.54	-195.35	253.16
Stator Temperature	77.19	24.65	20.01	162.68
Slip Frequency	1.67	2.93	-11.03	16.96
U_q	87.22	63.47	-36.11	242.13
U_d	-12.03	20.09	-98.28	75.19
I_q	57.26	103.35	-484.32	662.75
I_d	71.10	39.12	-0.79	269.67
Winding Temperature	80.50	28.06	20.00	188.36
Ambient Air Temperature	4.28	8.36	0.00	25.00

3.3 Data Preprocessing

To address the problems in the first dataset with 20 hours, we decided to crop the captures that displayed this extremely large temperature jump at the end. By analyzing the tail of each session, we could determine when the large jump occurred, which was usually in the last 20–50 rows of the data (or 2 to 5 seconds). These rows were then dropped from the dataset. The additional medium spikes that emerged intermittently in the middle of the session were handled by running that column through a *median filter*. The median filter is a commonly used digital filtering technique in image and signal processing to remove or reduce noise. The signal is processed by taking each entry and replacing it with the median of the

neighboring entries using a sliding window. The window size was empirically determined through trial and error and depends on the features.

Missing values were somewhat rare and mostly only presented themselves in 3 of the interesting features—the current, voltage, and ambient air temperature. It was decided that the best course of action was to impute these missing values. For the affected sessions, we observed that the current and voltage were turned off before the periods that contained the missing values. Other features that were not affected during the same time indicated a temperature decline that corroborated our observation of a steady state of 0 for the current and voltage. This allowed us to perform a Last Observation Carried Forward (LOCF). However, most of the new data did not contain ambient air temperature. Empirically, there seemed to be little correlation between the air temperature and rotor temperature. But a quick training run demonstrated that the model performed better with ambient air temperature included. For the newer dataset, the ambient air temperature was simply set to 22°C to mimic an average room temperature. In addition to handling missing values, column names were renamed where appropriate for brevity and to avoid confusion.

After treating each dataset session file separately, all files were merged into one final dataset file. An additional column was created to identify the individual session IDs for testing and training on specific profiles. This session ID column, as well as the time column, would be dropped during training.

3.3.1 Feature Engineering

Two features were engineered before training and added to the rest of the data by taking the magnitudes of the current and voltage in their $q-d$ coordinate system as follows:

$$I_s = \sqrt{I_q^2 + I_d^2} \tag{8}$$

$$V_s = \sqrt{V_q^2 + V_d^2} \tag{9}$$

where, I_s, V_s are the current and voltage magnitudes respectively and I_q, I_d, V_q, V_d are the current and voltages in their $q-d$ coordinate system.

3.4 Model Selection

The model selection step is a systematic process that involves evaluating various machine learning algorithms to determine the best fit for the objective of predicting the thermal state of an electric motor and its components. Criteria for model selection include the model's ability to process the specific types of data collected, its predictive performance, computational efficiency, and interpretability. The process begins by identifying a range of potential models suitable for the regression task. This includes various types of neural networks or machine-learning models, noted for their capability in handling complex, non-linear relationships in data.

This study aims to compare the predictive capabilities of a baseline ANN against a Physics-Informed Neural Network (PINN) (Kirchgässner et al., 2023) for temperature prediction. The baseline ANN serves as a reference model, leveraging only the input-output relationships present in the training data, while the PINN integrates additional physical constraints related to temperature dynamics. In addition to these two neural networks, a 1-D CNN was tested since related work has shown that they are quite accurate, but perhaps too heavy to be deployed in embedded systems. A 1-D CNN is a one-dimensional adaptation of the 2-D CNN. While 2-D CNNs are used to process two-dimensional signals such as images and videos,

Table 2

Grid search parameters for the MLP architecture.

Parameters	Search space	Best result
Activation Function	[ReLu, Leaky-ReLu, Sigmoid, Tanh]	Leaky-ReLu
Optimizers	[Adam, SGD]	Adam
Layers	[50,20]	[50,20]
Learning rate	[0.001, 0.01, 0.1]	0.001
Batch size	[64, 128, 256]	64
Dropout	[0.3, 0.5]	0.5
Patience	20	20

Table 3

The MLP model architecture. For the Leaky-ReLu, α is the negative slope.

Layer	Activation Function	Output	Parameters
FC	Leaky-ReLu	[input_size, 50]	$\alpha = 0.01$
FC	Leaky-ReLu	[50, 20]	$\alpha = 0.01$
Dropout			Dropout rate = 0.5
FC		[20, output_size]	

1-D CNNs process sequential data by performing 1-D convolutions on one-dimensional data (Kiranyaz et al., 2021). This makes them significantly lighter and a viable candidate for real-time time-series analysis. Furthermore, all experiments were conducted using the frameworks Tensorflow[11], PyTorch[12], and MATLAB.

While OLS regression has been shown to give good results by (Kirchgässner et al., 2021) on their dataset, our tests on our dataset did not yield satisfactory results. Therefore, the OLS results were not included in the analysis. A possible reason for this discrepancy is the difference in motor types. The original dataset used a PMSM, while our study focuses on an IM. These two motor types exhibit different characteristics and behaviors, which likely introduced complexities that OLS could not handle effectively.

3.4.1 Multilayer Perceptron

The MLP architecture was conceived through a combination of literature review and experiments done at Volvo. The idea behind this was to keep the model as small as possible and translates to minimizing the number of neurons and layers in the network. For the MLP, a grid search was employed using *scikit-learn*'s *GridSearchCV* module to explore the possible architecture space. The dataset was divided into 80% training and 20% testing/validation. The testing portion was removed from the end part of the dataset and left unshuffled to not tamper with any temporal dependencies. The model class itself is a dynamic architecture allowing us to add or remove layers, as well as, layer activation functions on the fly. A grid search was then performed on the following set of parameters from Table 2. Furthermore, early stopping was implemented with a patience of 20 epochs. The final model architecture is presented in Table 3.

[11] https://www.tensorflow.org/
[12] https://pytorch.org/

3.4.2 Thermal Neural Network

A Thermal Neural Network (TNN) represents a specialized neural network architecture that has been specifically crafted for modeling heat transfer processes (Kirchgässner et al., 2023). Unlike conventional neural networks, which can handle various data types and tasks, the TNN is tailored to handle heat transfer phenomena. This makes them invaluable for applications such as real-time temperature estimation and thermal control in systems where heat dissipation and management are crucial. By integrating laws and principles from thermodynamics and heat transfer theory, the TNN can accurately capture temperature dynamics from complex interactions and behaviors within a system. Traditionally, the modeling of heat transfer dynamics relied on Lumped-Parameter Thermal Networks (LPTNs) (Incropera et al., 2011), a technique used to represent thermal behavior using the lumped-parameter model. The lumped-parameter model is a simplification technique used in engineering and physics to model complex systems by dividing them into discrete elements or nodes. In this model, the system's behavior is represented by a network of interconnected elements, each characterized by specific parameters such as resistance, capacitance, and inductance. Furthermore, each element's temperature is described by their ODEs. These elements are considered to be localized at specific points within the system, hence the term "lumped." Another way of thinking about lumped elements is through Newton's equation of gravity, where the force of each attracting body is assumed to stem from a point at the center of mass. The lumped element model assumes that the system's properties and interactions can be adequately described by these discrete elements, neglecting spatial variations or gradients within the system.

Within the TNN framework, as depicted in Figure 10, LPTNs form the foundational framework for modeling heat transfer dynamics. Each thermal element in the LPTN corresponds to a neuron in the neural network, with the temperature of the element analogous to the activation state of the neuron. The connections between thermal elements, characterized by thermal resistances and capacitances, mirror the connections between neurons in the neural network. The dynamics of the LPTN, including heat transfer between thermal elements over time, are captured by solving the associated ODEs. Similarly, the dynamics of the neural network, including information propagation and neuron activation, are governed by the network's mathematical equations, often involving nonlinear transformations and activation functions.

In the TNN architecture, the equations governing the LPTN dynamics were reformulated using a first-order Euler discretization method. This method approximates the continuous-time dynamics by updating state variables at discrete intervals. Specifically, temperature estimates at each LPTN node are updated at each time step using the Euler method. The temperature estimate $\hat{\theta}[k+1]$ at the i-th node at time step $k+1$ is expressed as:

$$\hat{\theta}[k+1] = \hat{\theta}_i[k] + T_s \cdot \kappa_i[k](\pi_i[k] + \sum_{j \in N/i} (\hat{\theta}_j[k] - \hat{\theta}_i[k])\gamma_{i,j}[k] + \sum_{j=1}^{n} (\tilde{\theta}_j[k] - \tilde{\theta}_i[k])\gamma_{i,j}[k]) \tag{10}$$

where

- T_s is the sample time,
- $\kappa_i[k]$ is the feed-forward ANN output,
- $\pi_i[k]$ represents the power losses of the components,
- N is the set of neighboring nodes,
- $\tilde{\theta}_j[k]$ represents the ancillary temperatures such as the ambient and coolant temperatures, and
- $\gamma_{i,j}[k]$ represents the thermal resistances found between each node.

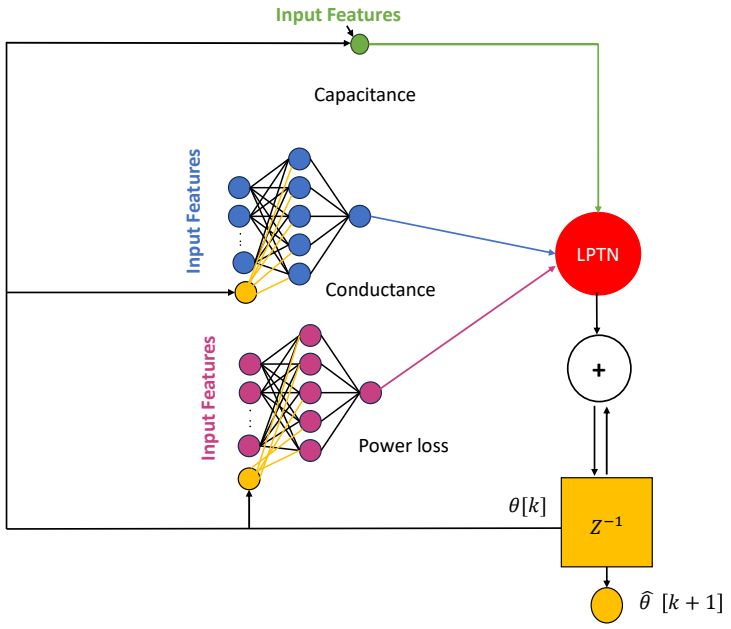


Figure 10. The capacitance is an estimated parameter, while the conductance and power loss are both estimated with sub-neural networks. The three parameters are then used to solve the LPTN, yielding the temperature prediction for $\hat{\theta}[k+1]$.

To validate the effectiveness of the TNN, a series of experiments were carried out using the data gathered by Volvo. The dataset underwent initial preprocessing where unnecessary columns were dropped, and the ambient air temperature column was adjusted to 22°C for consistency. The dataset was then split into features and targets, with normalization applied to both temperature-related and non-temperature-related features to manage scale disparities.

Hyperparameter Optimization (HPO) efforts were made and tested within the computational capabilities provided. A comprehensive overview of these hyperparameters, along with their search ranges and the optimal values discovered, is detailed in Table 4 and 5. The selection of interval limits aimed to prevent the expansion of parameters into dimensions that would be impractical for real-time computation. The "step size" specifies the number of forward passes aggregated and averaged for each backward pass during the training phase, utilizing error backpropagation. Five different step sizes that varied in powers of two, starting with 64 and ending with 1024, were tested. Each experiment was trained with 100 epochs, of which, a step size of 256 proved to be most fruitful. Subsequent experiments were then carried out using hyperparameters of Experiment 3. However, HPO was not solely focused on estimation accuracy, but also the model size. For practical deployment, smaller and more efficient models are crucial.

The dataset explored in this study consists of various measurement records of differing lengths. These records were divided into separate sets for training and validation purposes. This approach ensures no record is used in both the training and testing sets simultaneously, preventing the potential exploitation of record-specific peculiarities. Three exclusive sets of profiles are defined, functioning as either validation or test sets depending on the iteration. This strategy, excluding the model from observing both validation and test sets during training phases, incorporates an early stopping mechanism after each training epoch

Table 4

The TNN model architecture.

Layer	**Activation Function**	**Output**	**Parameters**
Input Layer	-	Varies	-
Conductance Network	Sigmoid	*n_conds*	Depends on *n_conds*
Power Loss Network Layer 1	Tanh	16	Depends on input size
Power Loss Network Layer 2	Linear	State Size	Depends on layer 1 output
Output Layer	-	State Size	-

Table 5

TNN experimental results.

Parameters	**Profile**	**Exp 1**	**Exp 2**	**Exp 3**	**Exp 4**	**Exp 5**
Epoch	-	100	100	100	100	100
Step size	-	64	128	256	512	1024
MSE	1	13.8	26.0	12.1	14.5	23.9
	11	47.4	21.8	56.2	26.7	78.1
	84	251.0	175.8	25.4	92.4	57.6
MAE	1	8.5	8.8	7.4	7.6	11.7
	11	16.7	12.5	18.2	10.7	13.6
	84	20.7	18.3	12.5	15.0	13.0

based on the validation set's performance. The test set is evaluated only once after training to determine the overall performance, with the final score being an average of both sets' outcomes.

Additionally, a distinct set of profiles is reserved during the HPO phase to avoid overfitting related to hyperparameter selection. This set, referred to as the generalization set, helps in evaluating the true generalization errors of the optimal hyperparameters. Specific profile groups serve as validation sets, while another set acts as the generalization set, facilitating an overlapping two-fold Cross-Validation (CV) strategy with an additional held-out set for assessing the hyperparameter optimum's true generalization capability.

3.4.3 1-D Convolutional Neural Network

A distinguishing feature of Convolutional Neural Networks (CNNs) (Hosseini et al., 2022) is their capability for spatially local connectivity, which allows for efficient parameter sharing across layers, enhancing their learning capabilities. These networks are not only noted for their excellent performance but also for their leading-edge effectiveness in analyzing sequential data. The convolutional layer (Conv) in particular, is crucial for extracting features from the data. As data progresses through the Conv layers, it undergoes convolution with specific kernels present in each layer. This convolution process, essentially a dot product between the entire set of input data and the kernels, results in the creation of a series of feature maps. Given that the data in question is one-dimensional sequential data, each Conv layer processes a 1-D input, denoted as $x(n)$. The convolution of this input with a 1-D kernel, as shown in Equation 11, $w(n)$, produces a feature map, $z(n)$, as (Zhao et al., 2019):

$$z(n) = x(n) * w(n) + \sum_{m=-l} x(m) \cdot w(n-m) \tag{11}$$

Table 6

The 1-D CNN model architecture.

Layer	**Activation Function**	**Output Shape**	**Parameters**
Conv1D-1	ReLU	$(n\_features, n\_steps - 6, 64)$	Filters: 64, Kernel Size: 7
Conv1D-2	ReLU	(None, $n\_steps - 12$, 64)	Filters: 64, Kernel Size: 7
MaxPooling1D	-	(None, $\frac{n\_steps-12}{2}$, 64)	Pool Size: 2
Flatten	-	(None, variable)	-
Dense-1	ReLU	(None, 32)	-
Dense-2 (Output)	-	(None, 1)	-

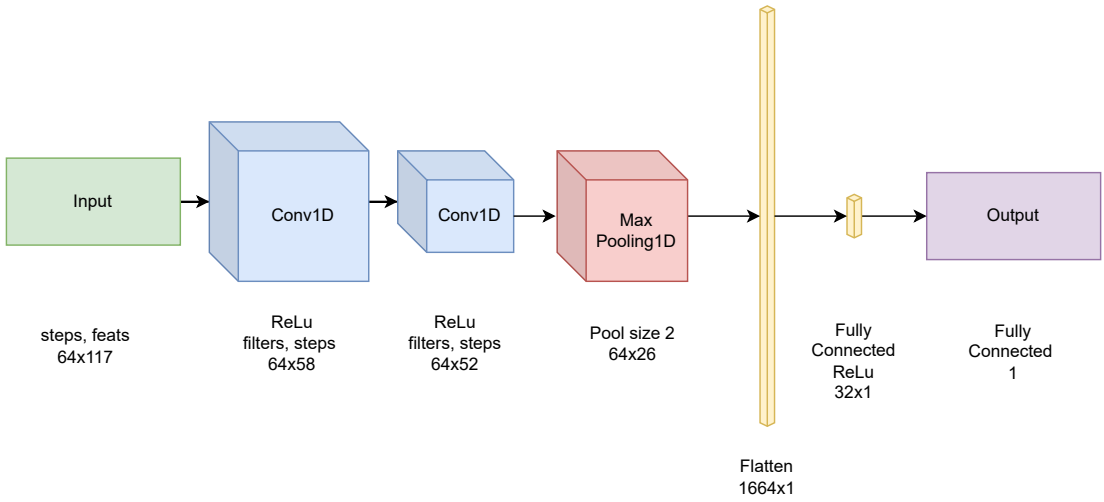


Figure 11. 1-D CNN architecture.

where

- $x(n)$ represents the one-dimensional input data at position n,
- $w(n)$ represents the one-dimensional kernel (or filter) used for convolution,
- $z(n)$ is the resulting feature map after the convolution process at position n,
- l denotes the size of the kernel.

The introduced 1-D CNN model is designed with two layers, incorporating a total of 134529 trainable parameters. Table 6 provides a detailed overview of the model's connectivity patterns, including specifics about each layer. A depiction of the model architecture is also available in Figure 11.

The exact output shapes and parameter counts for some layers (like Flatten and Dense) are dependent on the input data dimensions, specifically "n_steps" and "n_features", and the model's internal configuration. The Flatten layer's output shape, in particular, would be a function of the size of the data coming in from the previous MaxPooling layer. Similarly, the parameters for the Dense layers depend on the sizes of their input and output layers.

The CNN model is defined using TensorFlow and Keras. The model consists of convolutional layers, max pooling, flattening to a vector, dense layers, and a final output layer. It is designed to work with time series data. The model is compiled with the MSE as the loss function and Adam optimizer. It also tracks

the MAE and accuracy as metrics. Various callbacks are set up to save the best model, stop training early if validation loss does not improve, and adjust the learning rate. The model is trained using the training generator with validation data for a specified number of epochs. During training, the model's performance is evaluated on the validation data.

The columns of the dataset are first organized to separate input features from the target variable rotor temperature. A specific column "profile_id" is used to identify unique profiles within the data. The dataframe is rearranged to ensure the input features, target variable, and "profile_id" are in the desired order to facilitate easier manipulation and analysis later in the process. The number of input features "n_features" is determined for later use in the model.

To maintain uniformity in data size across different profiles, zero padding is applied. This involves adding rows of zeros before each profile's data. The "profile_id" is maintained in the last column of these zero rows. The dataset is then divided into chunks, each representing the zero-padded data of a unique "profile_id" followed by the actual data of that profile.

Scaling data is a crucial step in preparing for time-series forecasting as it can significantly enhance the performance of the DNN. In this work, we utilized the standard scalar method to normalize the data by deducting the mean and dividing by the standard deviation for all samples, as demonstrated in Equation 12:

$$x_{\text{scaled}} = \frac{x - \mu(x)}{\sigma(x)} \tag{12}$$

Here, x represents the original data, with $\mu(x)$ and $\sigma(x)$ indicating its mean and standard deviation, respectively, based on the physical measurements such as total current and voltage in Equations 8 and 9. It is important to note that when making predictions on the test dataset, the target values were adjusted back to their original scale.

To further refine the data, we introduced new features using Exponentially Weighted Moving Average (EWMA) and Exponentially Weighted Moving Standard (EWMS), applying different span values as outlined in Equations 13 and 14, where $\alpha = \frac{2}{(s+1)}$ and s represents an arbitrary span or look-back period. Consequently, at each time step t, four varied spans [500, 2200, 6000, 9000] are incorporated into the models as supplementary inputs. Given the high variability and dynamic nature of the input series, these exponentially weighted features help encapsulate some of the historical context of the inputs, thereby enhancing model performance. An example of a feature, along with its EWMA and EWMS across various spans, is illustrated.

The following equations outline the calculation of the exponentially weighted moving average (EWMA) and exponentially weighted moving standard deviation (EWMS):

$$\mu_{(\text{x})} = \frac{\sum_{i=0}^{t}(1-\alpha)^i x_{\text{t-i}}}{\sum_{i=0}^{t}(1-\alpha)^i} \tag{13}$$

where s represents the span and $\alpha = \frac{2}{(s+1)}$.

$$\sigma_{(\text{x})} = \frac{\sum_{i=0}^{t}(1-\alpha)^i x_{\text{i}} - \mu_{(\text{t})}}{\sum_{i=0}^{t}(1-\alpha)^i} \tag{14}$$

where $\mu_{(\text{t})}$ is the EWMA at time t.

Data is normalized using the *StandardScaler* from **scikit-learn** to ensure that all input features have a mean of 0 and a standard deviation of 1. This is crucial for models that are sensitive to the scale of input

Table 7

CNN experimental results. The best results are from Experiment 2.

Parameters	Profile	Exp 1	Exp 2	Exp 3
Epoch	-	20	22	18
Batch size	-	1024	512	2048
Step size	-	128	64	256
MSE	11	1.67	1.02	1,03
	84	367.96	155.78	642,28
MAE	11	1.02	0.69	0,42
	84	16.31	11.40	21,32
R2	11	0.98	0.99	1.00
	84	0.83	0.93	0,71

features.

The dataset is split into training, validation, and test sets based on "profile_id". This allows the model to be trained on one set of data, tuned on another, and finally evaluated on unseen data to test its predictive capability.

3.4.4 Implementation and Training of the Models.

Prior to settling on the models detailed in Tables 6 and 7, we applied an early stopping strategy, which allowed both models to achieve their optimal states within 18 to 22 training epochs, using varying step sizes and batch sizes to determine the best results. We experimented with multiple 1-D CNN and Long Short-Term Memory (LSTM) networks, varying in the number of layers and architectural designs, and documented their performance on the test dataset.

There are several reasons to vary step sizes and batch sizes. Larger batch sizes require more memory, and depending on the hardware capabilities, there is a limit to how large a batch size can be before running into memory issues. Also, smaller batch sizes often result in a more noisy gradient estimation, which can help the model escape local minima, potentially leading to better generalization on unseen data. Larger batches provide a more accurate estimate of the gradient but might lead to premature convergence to suboptimal minima. There is often a trade-off between the efficiency of learning and the ability of a model to generalize from the training data to new, unseen data. As will be demonstrated shortly, empirical evidence suggests that smaller batch sizes may lead to better generalization in some cases.

The step size in a time series model is the number of time steps the model looks back to make a prediction. This size determines how far back in time the model can see. If the step size is too small, the model may not be able to learn dependencies that span several time steps. The length of the sequence can impact how effectively the gradient can flow backward through the model during training. A longer sequence length allows the model to potentially learn more complex patterns. However, it also increases the risk of overfitting. If the sequence is too long, it can lead to problems like vanishing or exploding gradients. Likewise, longer step sizes mean that each training example is more computationally expensive to process, as the network needs to maintain state over a longer sequence during both forward and backward passes. This can increase both the time and memory requirements for training. Therefore, choosing the right batch size and step size is a balance between computational efficiency, model performance, and resource availability, which can depend on the specific characteristics of the data.

As shown in Table 7, Experiment 2 yielded the best results. This model was trained using the Adam optimization algorithm, starting with a learning rate of 0.001, a batch size of 512, and a step size of 64.

The plots can be found in the results section in Figures 40 and 41.

The *TimeSeriesGenerator* utility from Keras is used to prepare the training, validation, and test datasets for time series prediction. It generates batches of temporal data, which are suitable for training models that predict a future value from a sequence of past values.

Building upon the theoretical framework of the TNN model, the subsequent experimental section applies these principles to a set of real-world data. This approach not only validates the theoretical model but also demonstrates its practical implications in predicting rotor temperature under various operational conditions.

4 Experimental Evaluation

This chapter is pivotal for validating the machine learning models developed in earlier chapters, focusing on their practical application in predicting motor temperatures. The experiments aim to assess the accuracy and efficiency of different models using a set of predefined features. Key areas include testing the models against real-world data, optimizing them through iterative refinement, and evaluating their performance using standard metrics like Mean Absolute Error and Root Mean Squared Error. Additionally, we explore the challenges of exporting these models for use in embedded systems within electric vehicles, highlighting the role of MATLAB in facilitating model integration and execution. Through rigorous testing and evaluation, this chapter aims to bridge the gap between theoretical research and practical application, ensuring the models are both accurate and deployable in real-time environments.

4.1 Purpose of the Experiments

Firstly, the purpose of these experiments is to answer the question whether a machine learning model can be trained to accurately predict the motor temperature given a set of features or values, as previously discussed in Section 3. Secondly, the experiments allow us to test the efficacy of different types of models for this task. How does the data effect how the model is trained? If the model's performance is not satisfactory, we can consider revisiting earlier steps such as feature engineering, model selection. Given that a suitable model has been found, we would also like to determine how the model can be improved. This entails experimenting with different settings and hyperparameters and observing how these affect the outcome. The experiments help us iteratively refine our model until we achieve the desired level of performance. An additional crucial aspect is the inference time of the given model. The model should be able to accurately predict the motor temperature in a reasonable time. Finally, the experiments will involve determining whether the model can be optimized to run on embedded systems, in particular, a chip found in an electric vehicle.

4.2 Determining the Accuracy of the Model

The accuracy of the models will primarily be evaluated based on their error from the ground truth data through means of cross-validation. The dataset will be divided into two portions, one used for training the model, and the other portion is used for testing. This is done to prevent the model from overfitting with the trained data and to test the model's efficacy to generalize with new unseen data. Given the set of features in the test dataset, the models will make temperature predictions for those features which we can evaluate with the appropriate metrics. For temperature prediction, common metrics include MAE and MSE.

4.3 MATLAB

MATLAB stands for MATrix LABoratory and is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB is widely used for its interactive environment in a range of applications, including matrix and linear algebra, numerical analysis, signal and image processing, control systems, and financial modeling.

A significant portion of this research was dedicated to exploring the exportability and deployability of the tested models. One request in particular was to have the models in MATLAB and Simulink. MATLAB has several libraries to aid in such endeavours, including having a Deep Learning Toolbox, which works

similarly to Tensorflow or PyTorch. The Deep Learning Toolbox comes with many useful pretrained neural networks and allows you to define and add your own custom layers (MATLAB, 2024). Another benefit of using MATLAB is that the code can be exported to the C programming language for easy deployment. MATLAB also has the capability to invoke Python codes in its environment. The original Python code can thus be invoked in MATLAB without much additional effort.

4.4 Model Exporting

Due to the unique properties and configurations of the tested models, exporting them using the ONNX format or the specific save methods provided by TensorFlow and PyTorch was largely unsuccessful. The current structure of the TNN model makes tracing the model impossible. It is unclear how to implement truncated backpropagation through time with MATLAB's Deep Learning Toolbox, given its limitations. Conversely, while the 1-D CNN model is traceable, preprocessing the data poses an additional challenge due to the need to calculate the exponentially weighted moving averages. Attempts to export the models from Python and then import them into MATLAB could be circumvented by reconstructing the models directly in MATLAB. For the TNN model, attempts were made to simplify or to create alternate versions of the model that could be traced, or to determine if certain properties could be approximated to facilitate exporting. Sometimes, reducing complexity can aid in compatibility without significantly compromising performance. Future updates from MATLAB or alternative toolboxes might provide new functionalities or workarounds that were not previously available. We also investigated using middleware or conversion tools that specialize in translating models between different frameworks. Tools like MMdnn (a comprehensive toolkit for converting between different deep learning frameworks) might offer some pathways. Furthermore, specific solutions or similar case studies can be found in the user documentation or community forums for TensorFlow, PyTorch, and MATLAB. Transferring the model to MATLAB remains a task for future work as it could offer more model utility and an enhanced workflow.

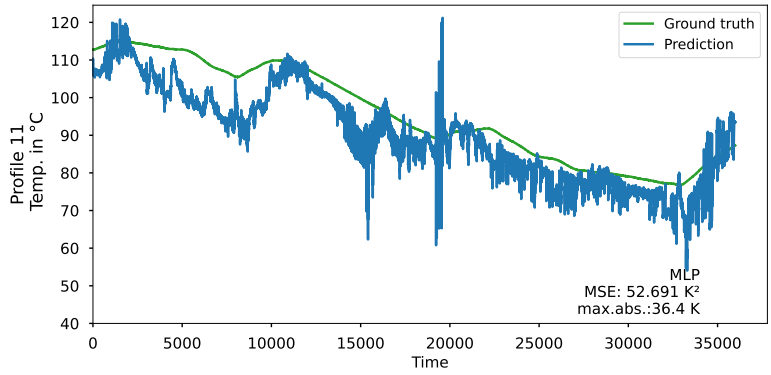


Figure 12. Estimating the rotor temperature using the MLP model for profile ID 11.

5 Results

This section presents the comprehensive results of our experimental study focused on the effectiveness of various neural network architectures in predicting rotor temperatures in induction motors. The experiments were systematically designed to evaluate the performance of MLP, TNN, and 1-D CNN across multiple test profiles. Unlike other test profiles that featured high-speed runs with high temperatures, similar to the training data conditions, the normal driving scenario data did not include any similar profiles in the training dataset, making it a unique challenge as shown in Subsection 5.5.

Our models results were compared with the existing model at Volvo Cars; however, due to company privacy regulations, the comparison results are not shown in the figures. We will explore how each model performs in terms of accuracy and reliability, highlighting the TNN model's superior ability to integrate heat-transfer principles for enhanced prediction accuracy. As hypothesized in our models description, the inclusion of motor housing temperature data was expected to enhance prediction accuracy. This chapter details how this integration significantly improved the accuracy of rotor temperature predictions, confirming our theoretical assumptions and reinforcing the model's relevance

5.1 MLP Model's Results

Results from Table 2 for the MLP model, as detailed in Section 3.4.1, demonstrated that this architecture performed the best across the grid search. Figures 12 and 13, depict the plotted rotor temperature estimates for profile IDs 11, and 84, respectively. In each figure, the green line represents the actual values, the blue line corresponds to our MLP estimation model.

In Figure 12, the MLP model starts off close to $110^{\circ}C$ before seesawing down closer to the ground truth. Notably, there is a large spike in the middle of the plot around the 19000th time step where all three lines converge when the MLP model achieves a closer fit.

In Figure 13, the MLP model predictions follow the ground truth closely until the 10000th time step. After that point, the MLP model begins to predict sharp temperature spikes. The MLP predicts sharp repeating waves around the 7500th time step, but seem to closely follow the ground truth approximately.

It is clear from the two figures that the benchmark MLP model can approximate the trend line, although very noisily. It consistently underestimates the rotor temperature. The best MLP was then tested on strategically chosen profile IDs (15, 30, 45, 60) to represent different measurement scenarios, with the input features mentioned in Subsection 3.2.1. The same input features were used for training the model

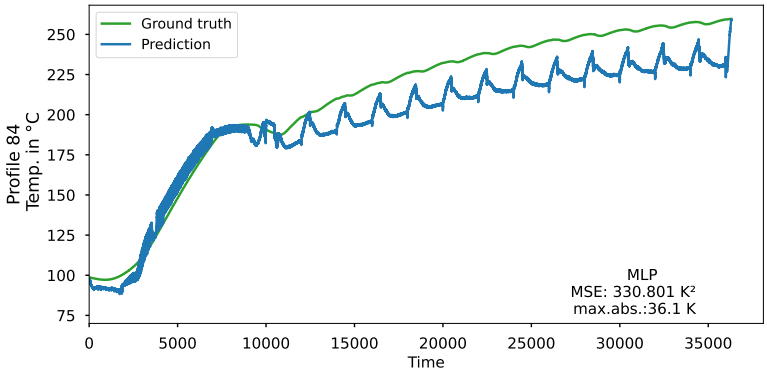


Figure 13. Estimating the rotor temperature using the MLP model for profile ID 84.

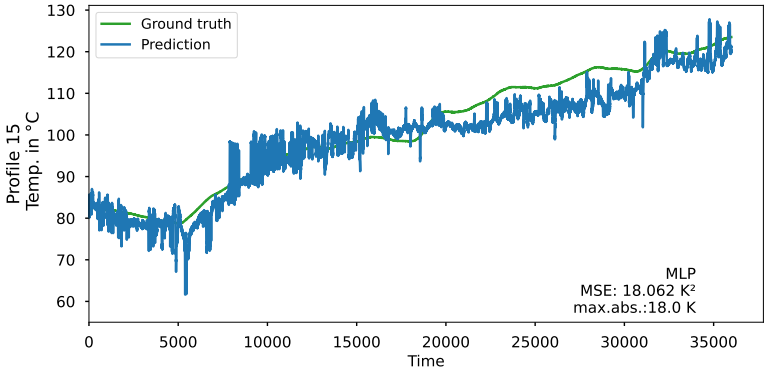


Figure 14. Estimating the rotor temperature using the MLP model for profile ID 15.

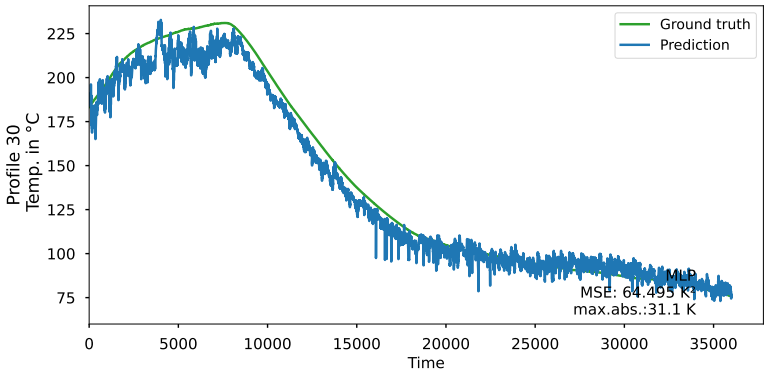


Figure 15. Estimating the rotor temperature using the MLP model for profile ID 30.

throughout, apart from the Motor Housing Temperature feature, which was excluded due to its absence in previous measurements. In total, 81 out of 85 profiles were used for training. Figures 14, 15, 16, and 17 illustrate the performance of the MLP model on the four selected profiles.

In Figure 14, the temperatures start around 90°C, drop to near 70°C, and rise back towards 130°C by the end of the iterations. The temperature predicted by the MLP model closely follows the overall trend of the ground truth but includes frequent spikes and some variability, especially noticeable in the middle sections of the graph. An MSE of 18.1 K², and a similar max absolute error, suggests a reasonably good

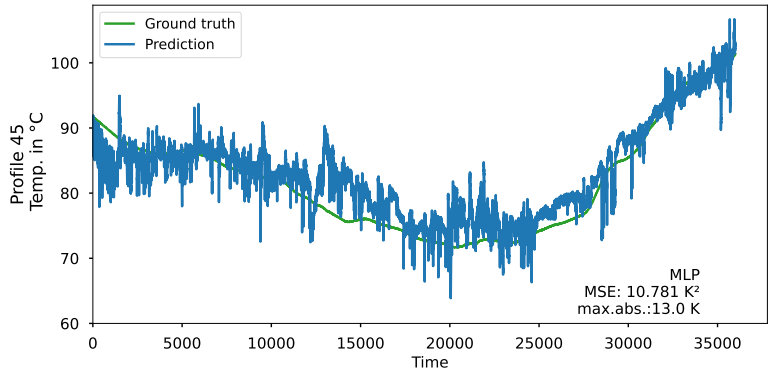


Figure 16. Estimating the rotor temperature using the MLP model for profile ID 45.

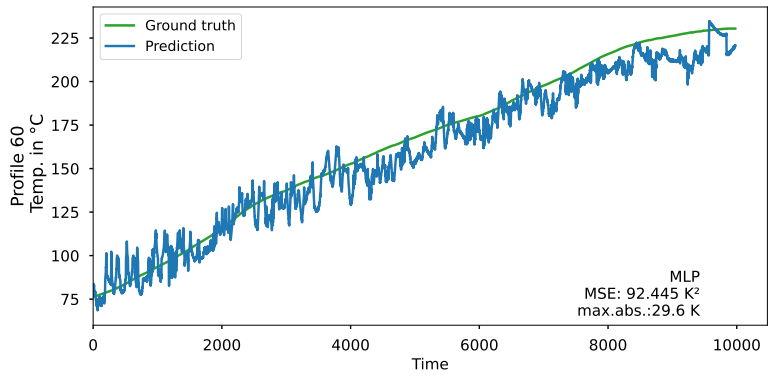


Figure 17. Estimating the rotor temperature using the MLP model for profile ID 60.

prediction accuracy.

Figure 15, shows a smooth curve with a peak early in the series, followed by a gradual and consistent decline. The temperature starts at around 180°C, peaks at about 240°C, and then gradually decreases to around 80°C. The prediction closely follows the ground truth but includes some variations, particularly in the form of spikes and minor deviations from the actual temperature curve. The model tends to underestimate the rotor temperature until the 19,000th time step before following it more closely. The MSE is relatively high at 64.5 K².

In Figure 16, shows a gradual downward trend for the first half of the series, which then transitions into a steady upward trend as the iterations progress. The model predicts this trend achieving a low MSE of only 10.8 K². The model slightly overestimates the temperature around decline in the middle portion.

In Figure 17, the ground truth line exhibits a smooth and steady increase in temperature, starting from around 75°C and gradually rising to approximately 225°C. The prediction line, although generally tracking the upward trend, shows a high degree of volatility and numerous spikes. The MSE of 92.4 K² is relatively high with a max. abs. error of 29.6 K.

The simple MLP model can effectively predict the rotor temperature, achieving the lowest MSE with profile 45 at just 10.8 MSE and a 13.0 maximum absolute error. In Figure 14, the MLP model starts at the same temperature as the ground truth and continues to closely track the green line. The graphs demonstrate the MLP model's capability to mimic the overall temperature trend of the profile accurately, though it struggles to replicate the smoothness of the actual temperature changes, which is evidenced by the frequent spikes and variability in the predicted temperatures. The model is generally effective in

following the major trends. However, there is room for improvement in handling the finer details and reducing peak errors.

Furthermore, we conducted experiments where the stator temperature was set as a target feature and removed as an input. This approach enabled us to evaluate the model's performance under different input configurations and to examine the impact of including or excluding the stator temperature on the accuracy of rotor temperature estimation. Figures 18, 19, 20, and 21 illustrate the performance of the model when both the rotor and stator temperatures are targets. The left plot in each figure shows predictions for the rotor temperature, while the right plot shows predictions for the stator temperature.

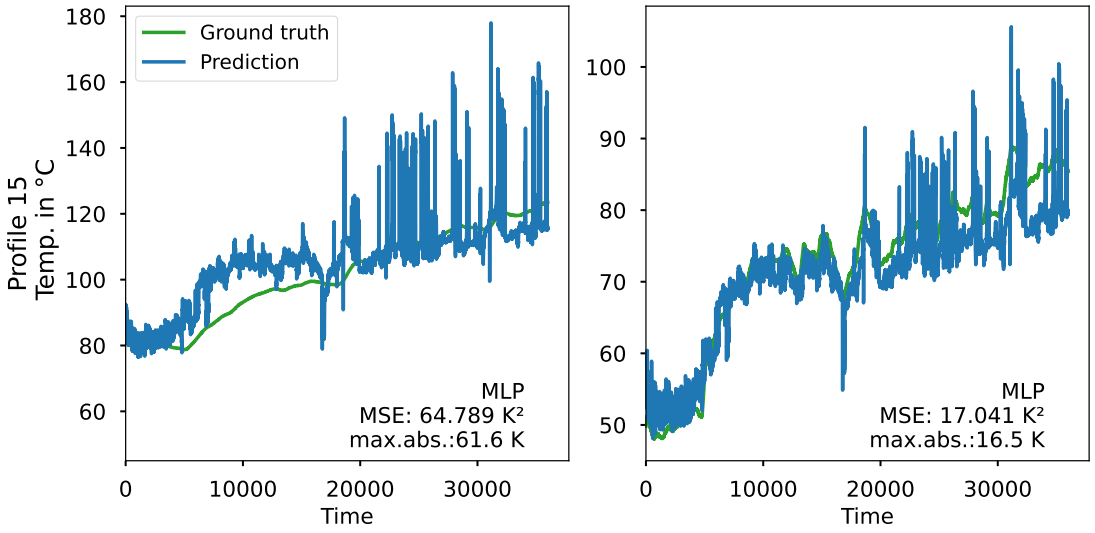


Figure 18. Estimating the rotor and stator temperature using the MLP model for profile ID 15.

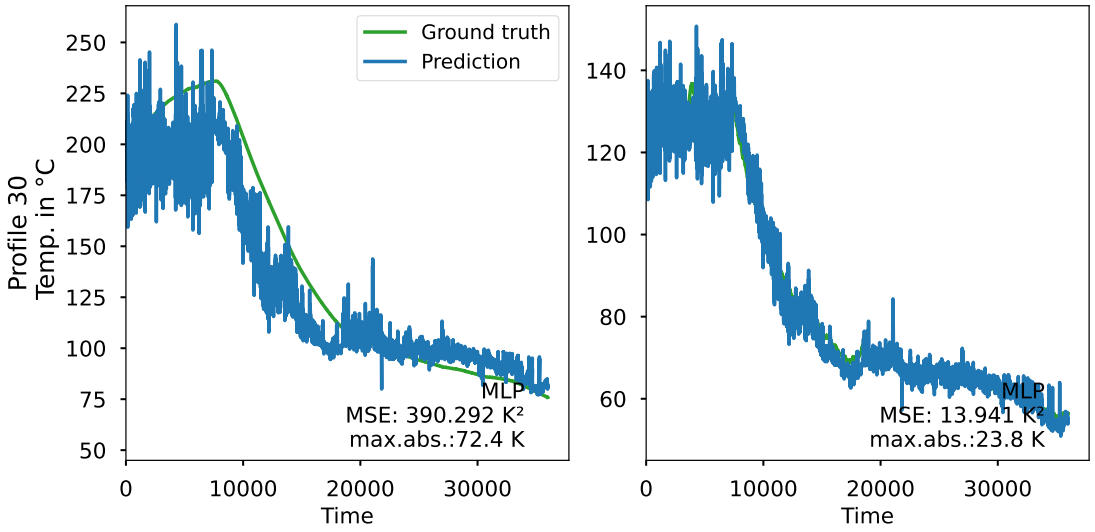


Figure 19. Estimating the rotor and stator temperature using the MLP model for profile ID 30.

In Figure 18, the rotor temperature predictions exhibit significantly higher volatility and errors compared to the stator. There is a gradual increase followed by a more stable trend, but the predictions feature drastic overestimations at several points. For the rotor, the maximum absolute error reaches an unacceptable 61.6 K, with the MSE similarly high at 64.8 K².

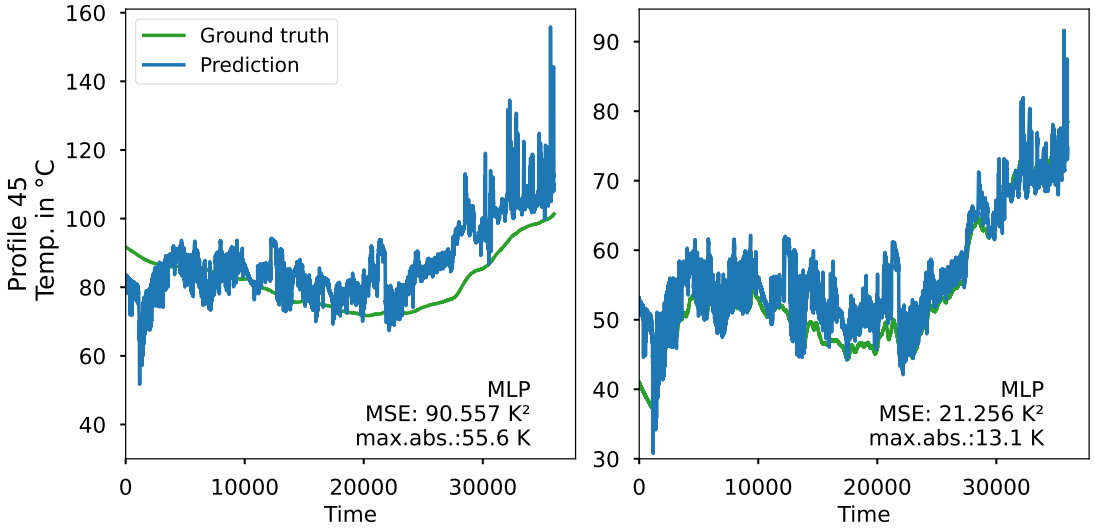


Figure 20. Estimating the rotor and stator temperature using the MLP model for profile ID 45.

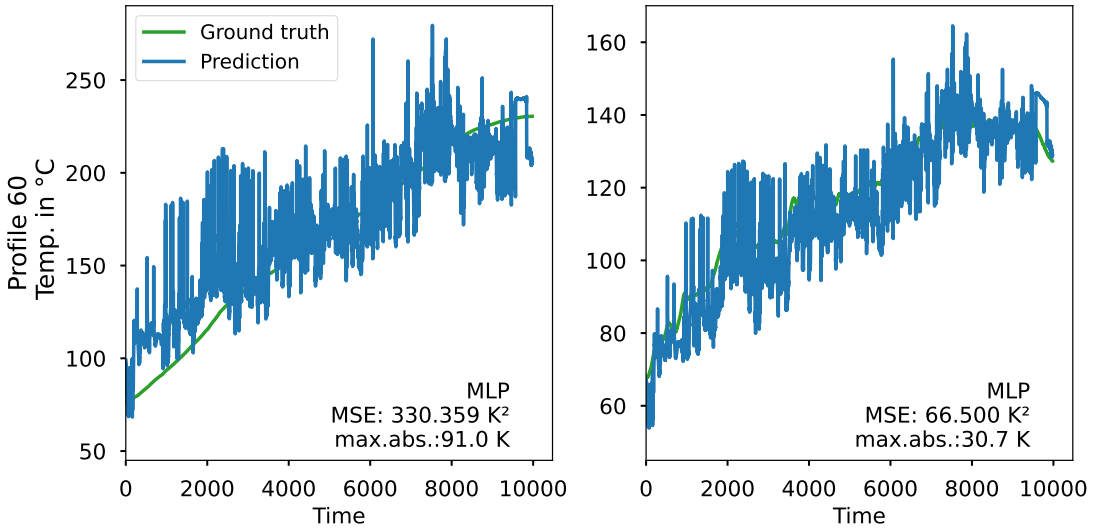


Figure 21. Estimating the rotor and stator temperature using the MLP model for profile ID 60.

Figure 19 displays dramatic fluctuations in the predicted temperatures with extreme peaks that significantly overshoot the ground truth, especially during the initial phase. The ground truth shows a peak early on followed by a steady decline. While the stator temperature predictions remain within acceptable ranges, the MSE for the rotor temperature estimation reaches a staggering 390.3 K², and the maximum absolute error climbs to 72.4 K.

Figure 20 presents another clear example of how the model mimics the stator temperature estimation and incorrectly applies it to the rotor. The ground truth for the rotor temperature is much smoother, while the model inaccurately applies the more jagged stator estimation to the rotor. The rotor temperature starts at around 90°C, dips to 70°C around the 22,000th iteration, and then rises again to 100°C. The stator experiences more sudden temperature rises, starting from 40°C and quickly escalating to 80°C.

In Figure 21, the ground truth shows a gradual upward trend followed by stabilization and a sharp peak

towards the end. The rotor temperature predictions exhibit significant volatility, with the highest maximum absolute error of the four profiles at 91.0 K. The stator temperature estimation remains manageable but still high, with an MSE of 66.5 K² and a maximum absolute error of 30.7 K.

When we designated the stator temperature as a target rather than an input, the MLP model became more adept at estimating the stator temperature compared to the rotor temperature. Interestingly, the rotor temperature predictions closely mirrored those for the stator temperature. However, as shown in Figure 19, there is a significant disparity between the accuracy of predictions for the rotor and stator temperatures. Specifically, the Mean Squared Error for the stator temperature was approximately 14 K, while the MSE for the rotor temperature soared to 390 K. Figure 21 highlights the largest maximum absolute temperature difference in rotor temperature estimation at 91 K. Concurrently, the worst stator predictions had a more manageable MSE of 66.5 K² and a maximum absolute temperature of 30.7 K.

By allowing the model to predict rotor temperature independently of stator temperature, Volvo could potentially eliminate the need for a stator sensor in their electrical machines, thereby reducing costs. To further improve these estimations, we optimized the training of the MLP model with new measurement data that includes motor housing temperature and coolant temperatures. Figures 22, 23, 24, and 25 illustrate the performance of the model when both the rotor and stator temperatures are targets.

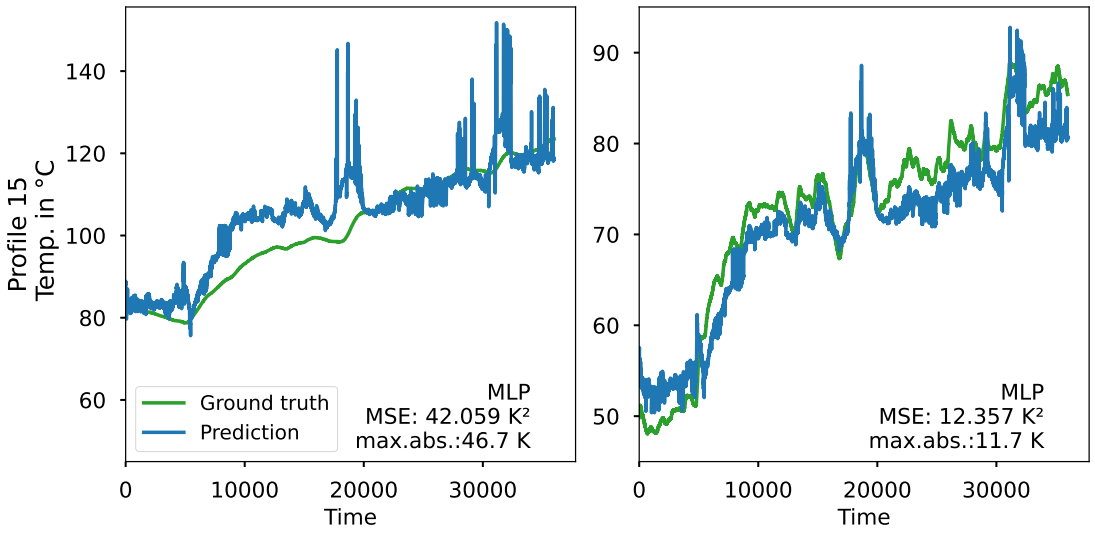


Figure 22. Estimating the rotor and stator temperature using the MLP model for profile ID 15.

Adding the motor housing temperature has improved the estimation of the rotor temperature resulting in less grainy, and overall, more accurate predictions, such as in Figures 22 and 23. In Figures 24 and 25, the less noisy prediction has slightly increased the MSE of the stator temperature, but it has improved the estimation of the rotor temperature. The MSE and Max Abs for estimating rotor temperature were compared with the existing model, as presented in Table 8. Finally, the model has a file size of 10346 bytes for the 1862 parameters. It has an average prediction time of 0.000193 seconds per row of input.

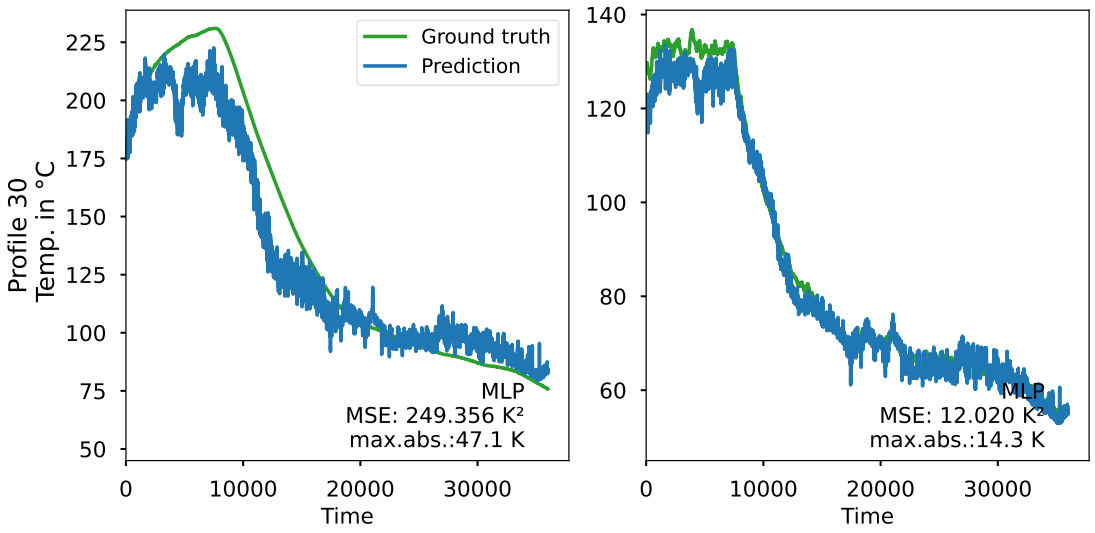


Figure 23. Estimating the rotor and stator temperature using the MLP model for profile ID 30.

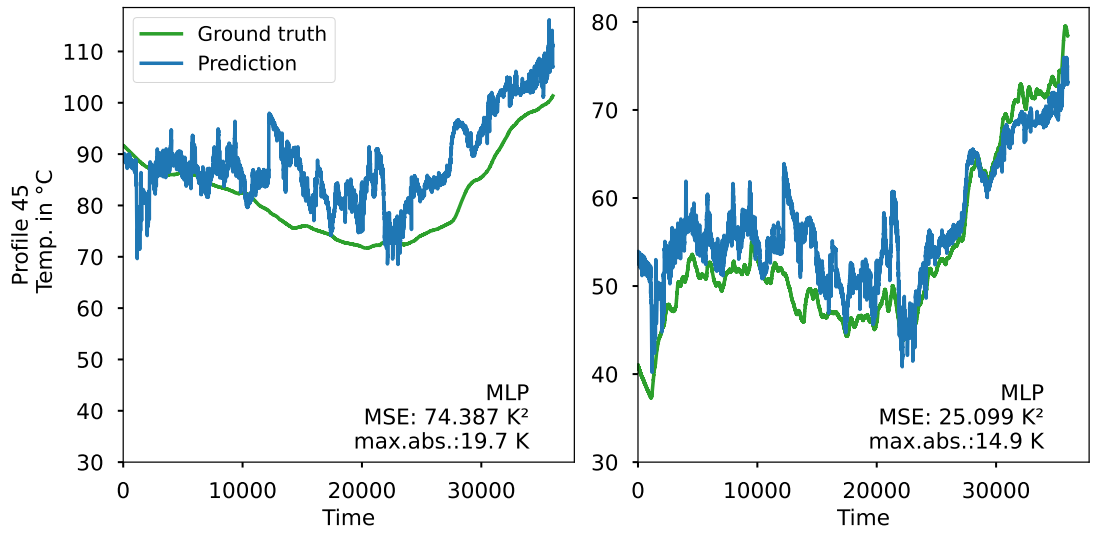


Figure 24. Estimating the rotor and stator temperature using the MLP model for profile ID 45.

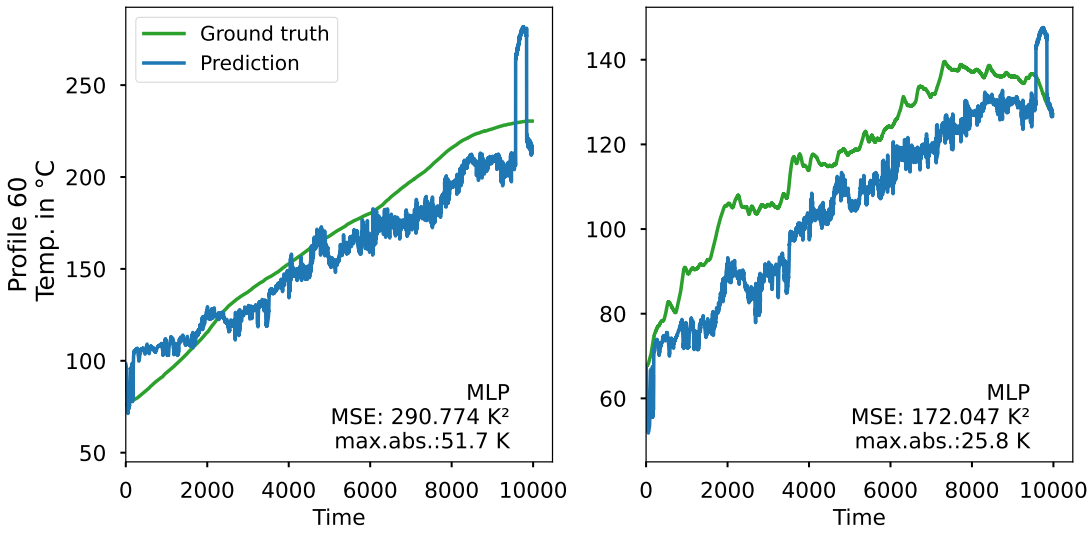


Figure 25. Estimating the rotor and stator temperature using the MLP model for profile ID 60.

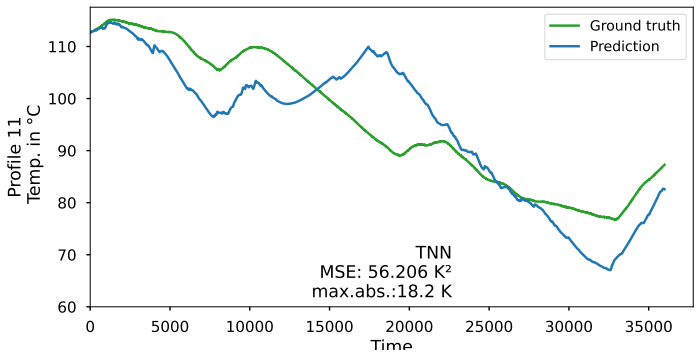


Figure 26. Estimating the rotor temperature using the TNN model for profile ID 11.

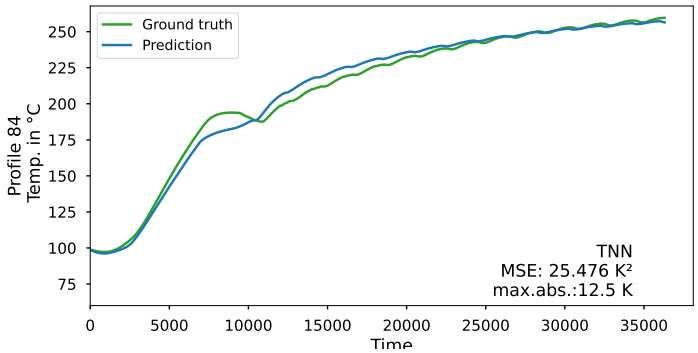


Figure 27. Estimating the rotor temperature using the TNN model for profile ID 84.

5.2 TNN Model's Results

The estimating of the rotor temperature results from Experiment 3 for TNN model, as detailed in Section 3.4.2, demonstrated better performance compared to other experiments across the test profiles. Figures 26 and 27, depict the plotted rotor temperature estimates for profile IDs 11 and 84, respectively. In each figure, the green line represents the actual values, and the blue line corresponds to our TNN estimation model.

For Figure 26, the model starts relatively well-aligned with the actual temperatures, capturing the trends though not the exact peaks and troughs. Over time, particularly after around 20,000 iterations, the prediction deviates significantly, notably underestimating the temperature as the actual values drop steeply. This is quantified by a MSE of 56.2 K² and a maximum absolute difference of 18.2 K, indicating that the model struggles to accurately capture the sharper changes and overall decline in rotor temperature in this profile.

Figure 27, on the other hand, shows a more closely aligned prediction across the whole range of iterations. The model follows the rising trend in rotor temperatures quite accurately up until it starts plateauing around 25,000 iterations. Although minor deviations are visible, they are less pronounced than in Figure 26. This is reflected in the MSE of 25.5 K² and a maximum absolute difference of 12.5 K.

Overall, the TNN model consistently tracks the fluctuations of the ground truth data more closely while maintaining a tighter fit. Additionally, the TNN offers the advantage of predicting outcomes with specified initial conditions. These effects and benefits are evident in Figure 26, where the TNN model closely aligns

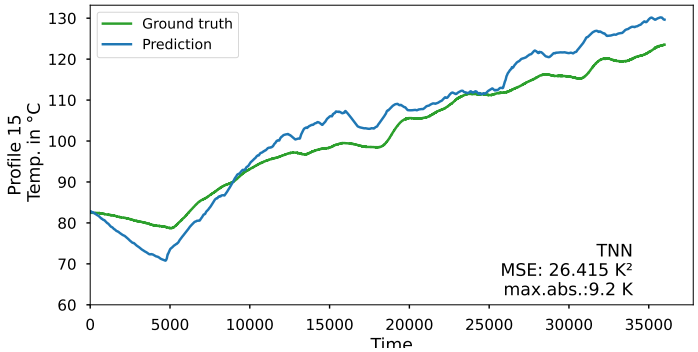


Figure 28. Estimating the rotor temperature using the TNN model for profile ID 15.

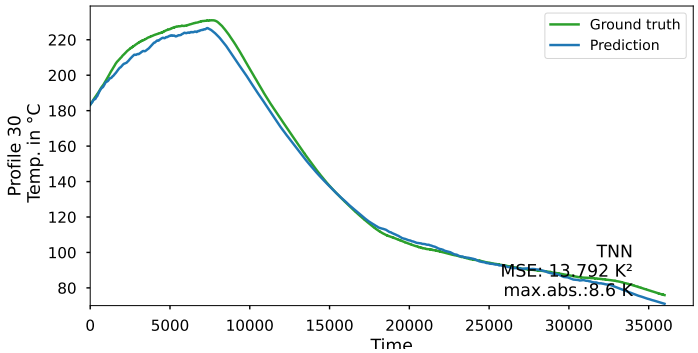


Figure 29. Estimating the rotor temperature using the TNN model for profile ID 30.

with the actual temperature from the outset.

Upon identifying the model in Experiment 3 as the most effective, the TNN model was further trained by excluding each of the selected profile IDs (15, 30, 45, 60) for testing purposes, in line with Subsection 3.2.1, we maintained consistency by training the model with the same input features throughout, with the exception of the Motor Housing Temperature feature, which was excluded due to its absence in the previous measurements. These test profile IDs were strategically chosen to represent different measurement scenarios as depicted in Figures 28, 29, 30 and 31. The concatenated dataset encompassed a total of 81 profiles out of 85, with four profiles reserved for testing.

Referring to Figures 28, 29, 30, and 31, the evaluation of the model's prediction for rotor temperatures, represented by the blue line, illustrates its varying degrees of alignment with the actual temperatures indicated by the green line.

In Figure 28, initially, the model experiences a significant deviation from the ground truth, undershooting the actual temperatures by a notable margin around 5,000 iterations. Subsequently, the model demonstrates a strong recovery, closely following the true temperature trajectory, though slightly lagging. The MSE for this profile is 26.4 K^2, and MAX ABS is relatively modest at 9.2 K.

In Figure 29, the model's robust performance throughout the temperature profile. It effectively captures both the ascent and descent phases of the temperature curve, adhering closely to the ground truth with minimal deviations. This indicates the model's proficiency in managing smooth, bell-shaped temperature variations, evidenced by an MSE of 13.8 K^2 and a MAX ABS of 8.6 K.

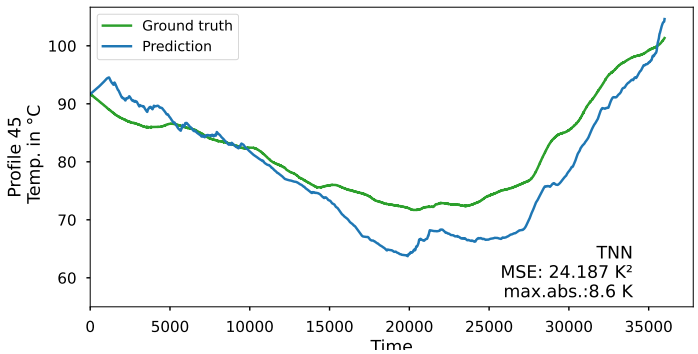


Figure 30. Estimating the rotor temperature using the TNN model for profile ID 45.

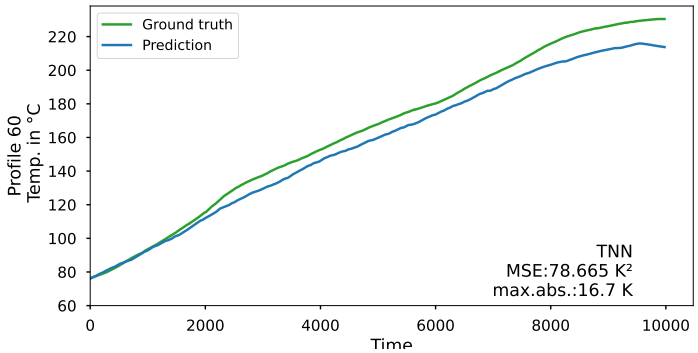


Figure 31. Estimating the rotor temperature using the TNN model for profile ID 60.

Figure 30 depicts the TNN model's blue line following a gradual decline before predicting a significant bump, leading to deviations from the target between 5000 and 20000. Initially, the prediction aligns well with the ground truth during the temperature decline. Nevertheless, as temperatures bottom out and begin to rise, the model fails to capture the steep upward trend, resulting in significant deviations. This contributes to an MSE of 24.2 K² and a MAX ABS of 8.6 K.

In Figure 31, profile 60 illustrates a consistent temperature increase. The TNN blue line consistently underestimates the temperature. Initially, the model accurately tracks the rising trend but begins to underestimate temperatures. Near the end, the model incorrectly predicts a sharper increase than what is observed in the ground truth. This profile records the highest MSE at 78.7 K² and a MAX ABS of 16.7 K, underscoring the model's challenges in handling extended rising temperature trends.

Across these test profiles, the model exhibits varying levels of accuracy, generally excelling in scenarios with smooth temperature changes as observed in Figure 29. However, it encounters difficulties in profiles with abrupt transitions or very steep gradients, particularly evident in Figure 31.

Moreover, we conducted experiments where the stator temperature was excluded as an input feature but considered as an output variable, as illustrated in Figures 32, 33, 34, and 35. This enabled us to evaluate the model's performance under different input configurations and examine the impact of including or excluding stator temperature on the accuracy of rotor temperature estimation.

From Figure 32, without the stator temperature as an input, the model exhibits an initial struggle to accurately predict the rotor temperature, with a significant underestimation early in the iterations. Despite this, it gradually improves alignment with the ground truth as iterations progress. The MSE here is

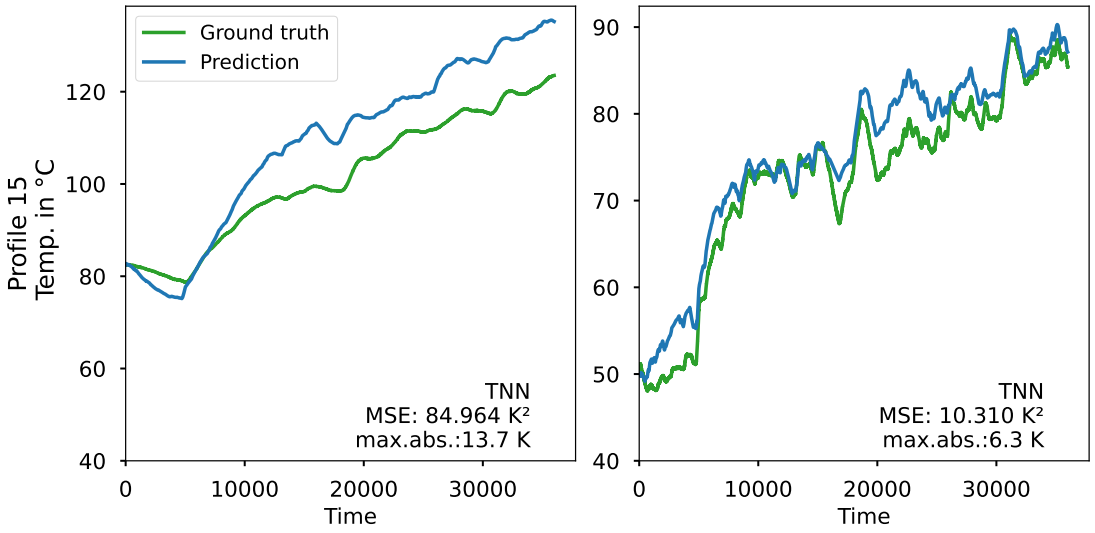


Figure 32. Estimating the rotor and stator temperature using the TNN model for profile ID 15.

considerably high at 85 K² with a MAX ABS of 13.7 K, highlighting substantial challenges in initial prediction accuracy under this input configuration.

In Figure 33, the prediction without stator temperature input shows a closer following of the actual temperature. The MSE is improved at 19.5 K², with less pronounced deviations than in Figure 32.

Figure 34 reveals moderate predictive accuracy, with the model aligning well with the ground truth throughout the temperature fluctuations. The MSE is relatively low at 8 K², and the MAX ABS stands at 8.5 K, indicating a better model performance in capturing the temperature dynamics even without stator temperature data. This shows that the model can adequately handle mid-range temperature changes even without the stator temperature as a guiding input.

Finally, Figure 35 shows that, the absence of stator temperature input results in the model struggling significantly with higher temperature ranges, especially as the profile exhibits steeper temperature inclines. The MSE skyrockets to 156.2 K² with a MAX ABS of 17.4 K, underscoring significant difficulties in predicting high and rapidly changing temperatures accurately.

The comparative analysis of these Figures 32, 33, 34, and 35 illustrates that the exclusion of stator temperature as an input feature generally increases the prediction error, particularly in scenarios involving rapid temperature changes or at higher temperature levels. While the model can adapt and align with actual temperatures to some extent, the increased MSE and MAX ABS differences across most profiles suggest that including stator temperature data significantly enhances prediction accuracy and model reliability.

Despite not incorporating stator temperature as an input, the estimation of rotor temperature using the TNN model remains satisfactory. While stator temperature can be accurately measured using sensors for various calculations, the rotor temperature sensor is prohibitively expensive and prone to breakage in practical environments. The temperature disparity between the rotor and stator can be significant in some machines, necessitating precise rotor temperature estimation independent of stator temperature readings to avoid erroneous parameter estimations. Additionally, machinery might require protection with current limitations at higher rotor temperatures, even if stator temperatures are not excessively high.

By enabling the model to estimate rotor temperature without relying on stator temperature, Volvo Cars can reduce costs by eliminating the need for a stator sensor in their electrical machines. Therefore, we endeavored to enhance these estimation results by optimizing the training of the TNN model using new

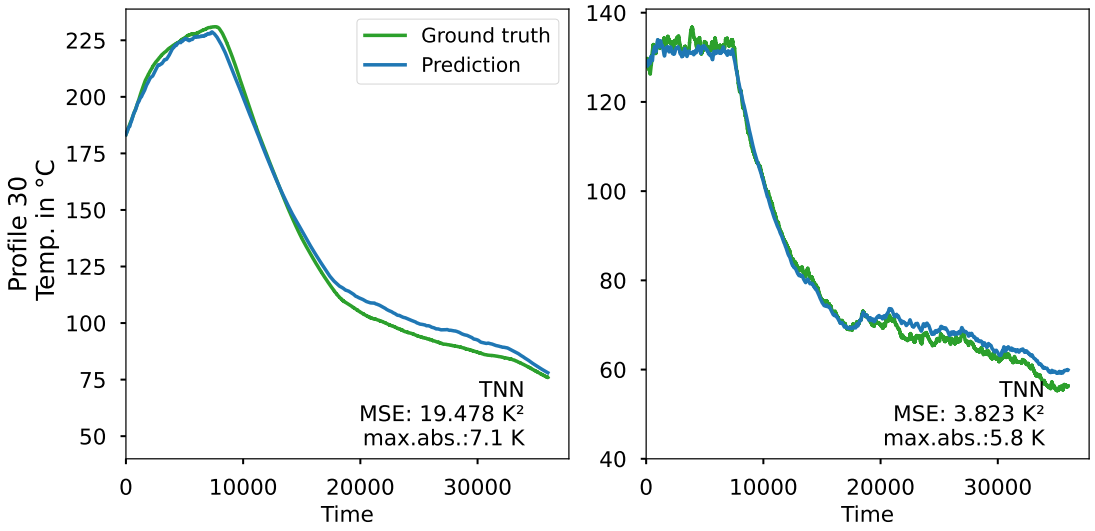


Figure 33. Estimating the rotor and stator temperature using the TNN model for profile ID 30.

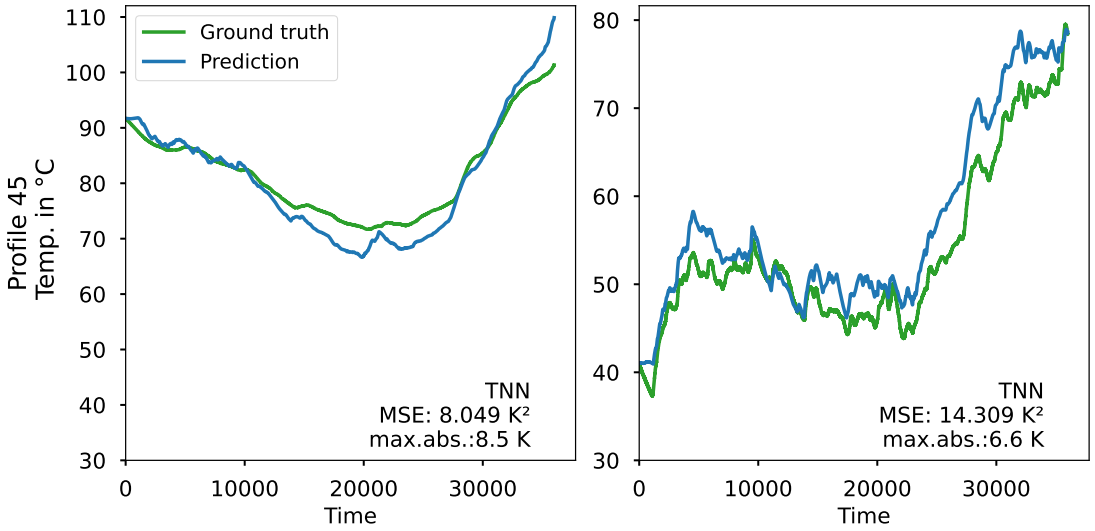


Figure 34. Estimating the rotor and stator temperature using the TNN model for profile ID 45.

measurement data, which includes motor housing temperature and coolant in and out. This optimization has shown improvements in the accuracy of rotor temperature estimation, as demonstrated in Figures 36, 37,38, and 39.

The MSE and Max Abs for estimating rotor temperature using TNN were compared with the existing models, as presented in Table 8. Finally, the TNN model has a file size of 6950 bytes for a in total 810 parameters. It has an average prediction time of 0.000176 seconds per row of input.

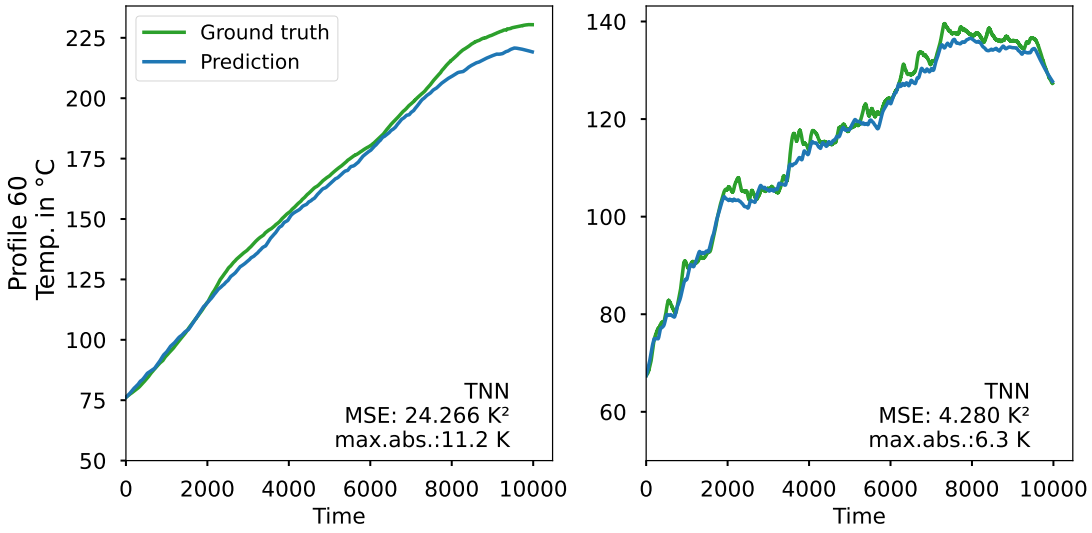


Figure 35. Estimating the rotor and stator temperature using the TNN model for profile ID 60.

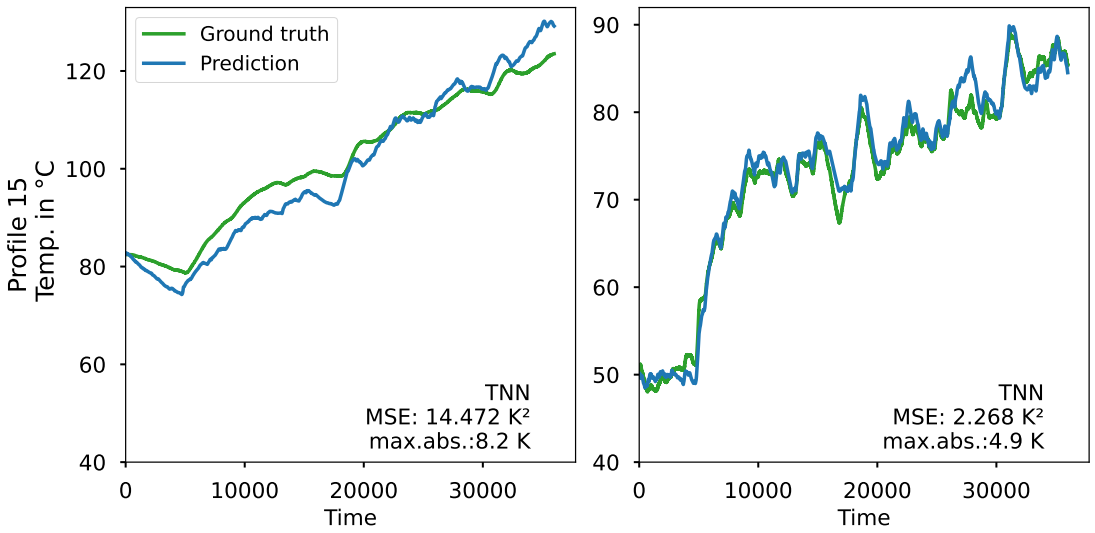


Figure 36. Improving the estimating the rotor and stator temperature using the TNN model for profile ID 15.

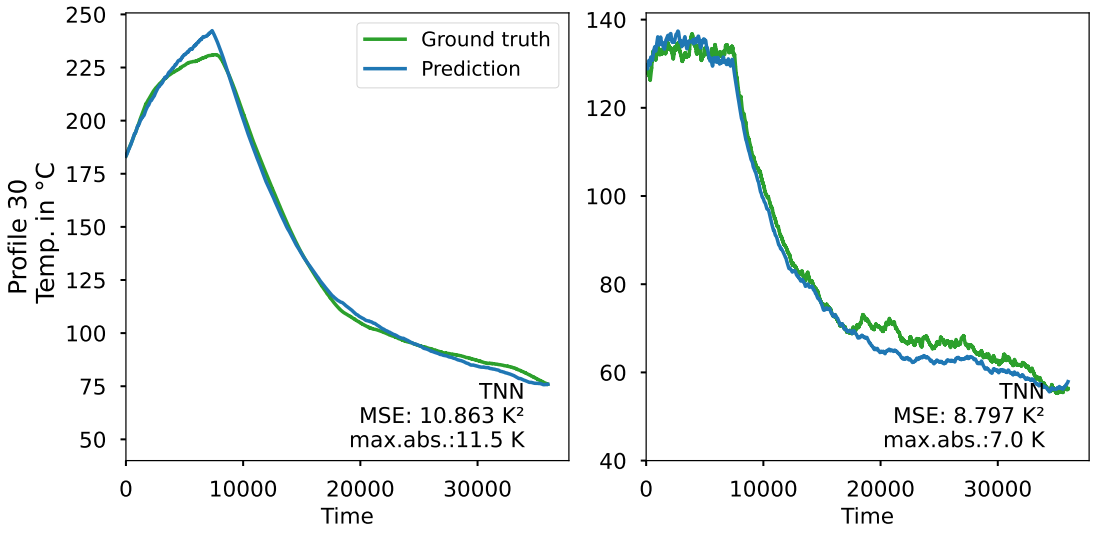


Figure 37. Improving the estimating the rotor and stator temperature using the TNN model for profile ID 30.

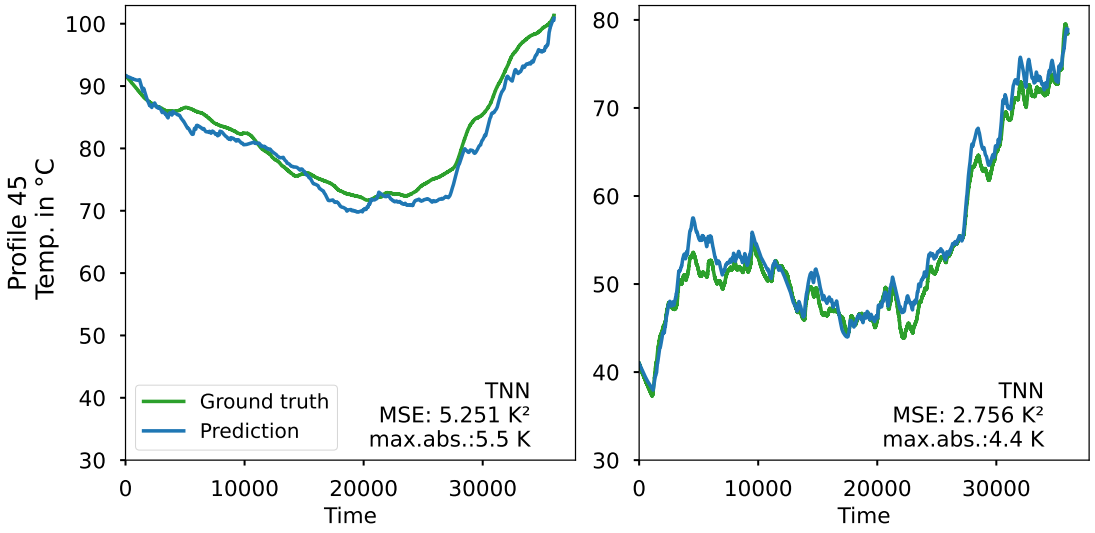


Figure 38. Improving the estimating the rotor and stator temperature using the TNN model for profile ID 45.

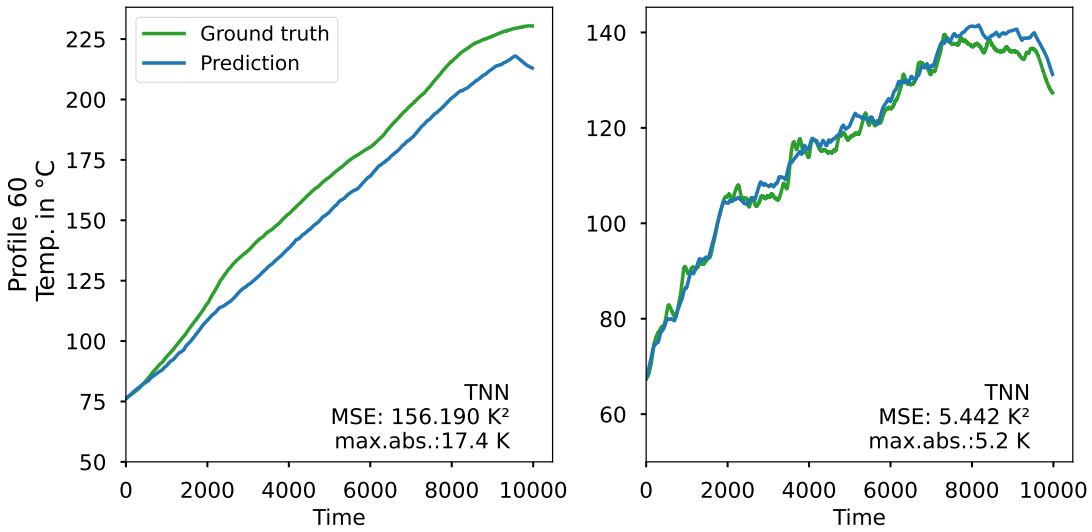


Figure 39. Improving the estimating the rotor and stator temperature using the TNN model for profile ID 60.

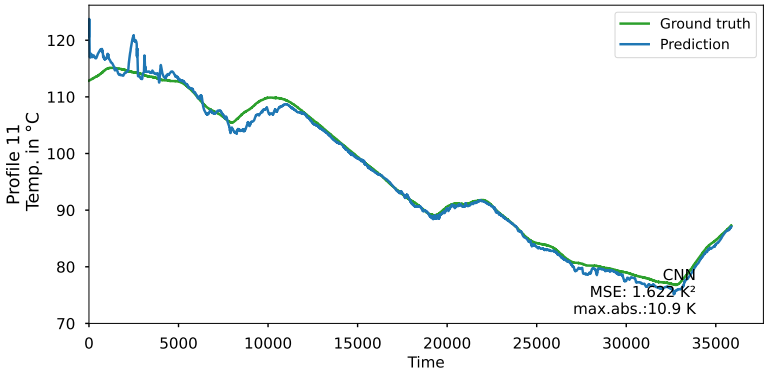


Figure 40. Estimating the rotor and stator temperature using the 1-D CNN model for profile ID 11.

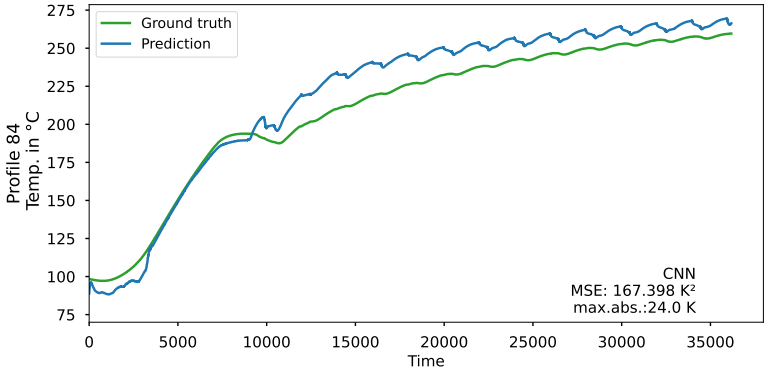


Figure 41. Estimating the rotor and stator temperature using the 1-D CNN model for profile ID 84.

5.3 1-D CNN Model's Results

The experimental results in Section 3.4.3, where we varied the batch sizes and step sizes, showed the best results with the parameters used in Experiment 2 as shown in Table 7 . The results are presented in Figures 40 and 41, where the rotor temperature estimates for profile IDs 11 and 84 are plotted, respectively. In each figure, the green line represents the actual values, and the blue line corresponds to our 1-D CNN estimation model.

In Figure 40, the model initially predicts the temperature to be $10°C$ lower than the actual temperature, contributing to the high maximum absolute error for this profile. After a shaky start, the model's predictions begin to track the ground truth very closely starting from the 5000th time step. The model cleanly follows the downward slope with an MSE of only $1.6°C$.

In Figure 41, the model follows the sharp temperature rise until the 7000th time step. Then, a gap opens between the ground truth and the predictions. The green line exhibits a saw-tooth pattern, which the model attempts to imitate. Over time, the blue and green lines appear to be converging. However, the MSE for this test profile reaches $167.4°C$.

The same experiments conducted for the MLP and TNN were also performed for the 1-D CNN, focusing on profiles 15, 30, 45, and 60. These experiments aimed to evaluate the 1-D CNN's performance when using the stator temperature as an input feature as shown in Figures 42, 43, 44 and 45.

Figure 42 shows a disparity of 7 degrees between the ground truth and the prediction at the start. They

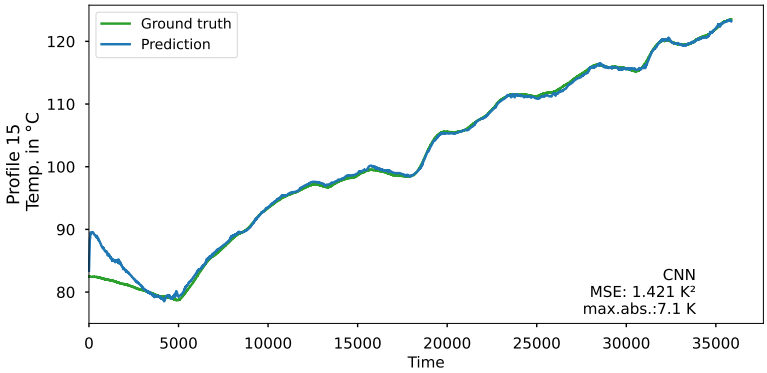


Figure 42. Estimating the rotor and stator temperature using the 1-D CNN model for profile ID 15.

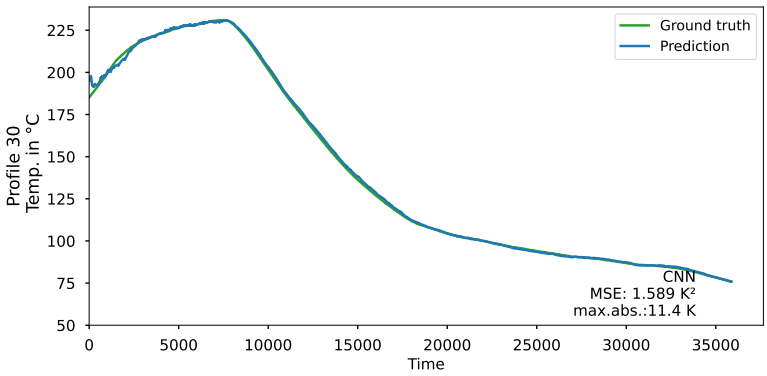


Figure 43. Estimating the rotor and stator temperature using the 1-D CNN model for profile ID 30.

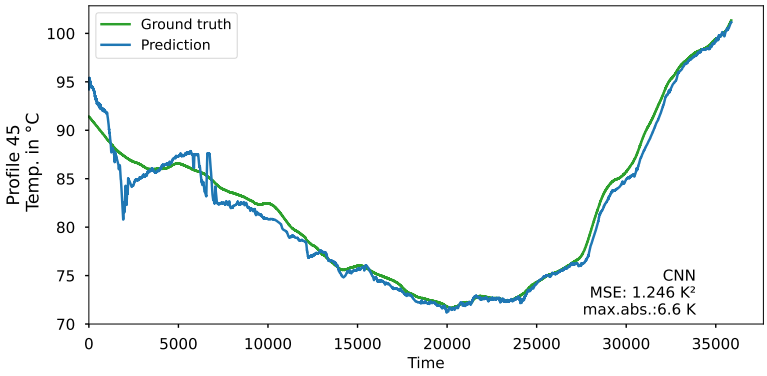


Figure 44. Estimating the rotor and stator temperature using the 1-D CNN model for profile ID 45.

converge at around the 5000th time step before climbing together. The model's predictions are exceptionally close to the ground truth.

In Figure 43, the model once again exhibits a disparity between the initial predicted temperature and the ground truth of 11.4°C. However, the two lines quickly converge after 1000 time steps. They rise to above 220°C until the 7000th time step before falling steeply until the end. The 1-D CNN shows no issues with an MSE of only 1.6°C.

In Figure 44, there are some abrupt spikes at around 2500 and 7000 time steps that the 1-D CNN appears

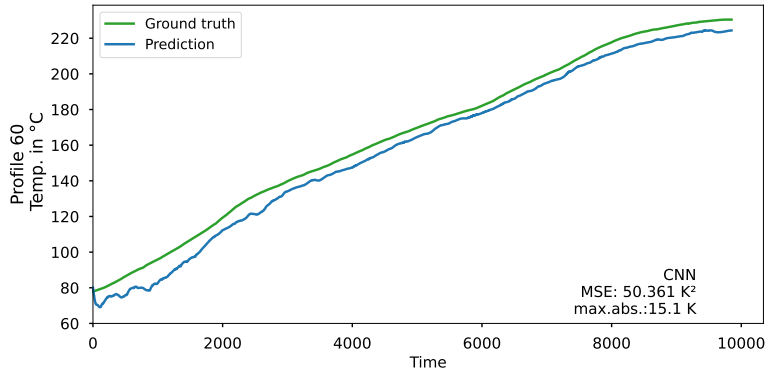


Figure 45. Estimating the rotor and stator temperature using the 1-D CNN model for profile ID 60.

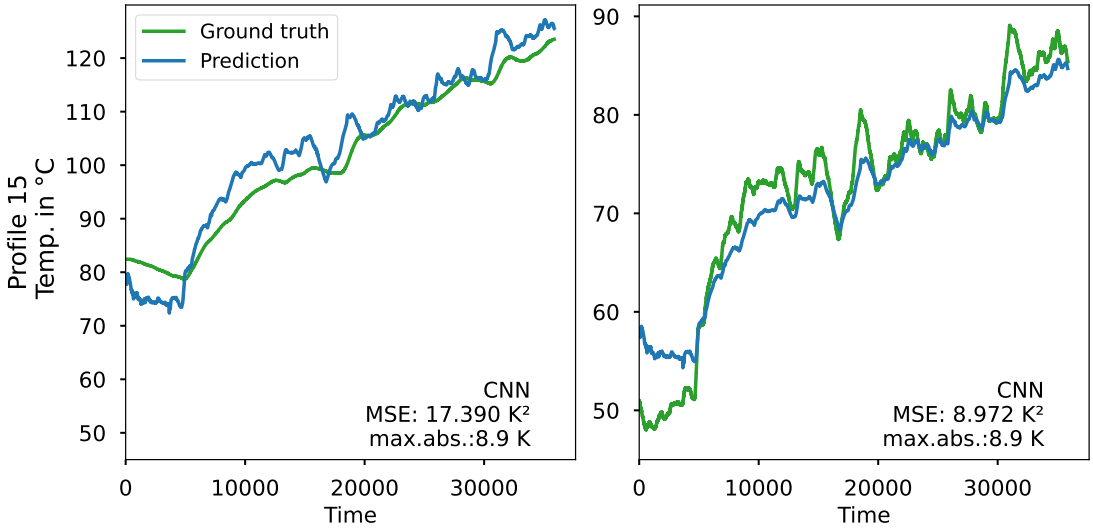


Figure 46. Estimating the rotor and stator temperature using the 1-D CNN model for profile ID 15.

to have smoothed out. These spikes resulted in a maximum absolute error of 6.6°C. There were some large spikes at the beginning, but the ground truth starts to stabilize after the 8000th time step. The model predicts the ground truth very well up to the 20000th time step, and then up again. The lowest MSE was achieved here, at only 1.2°C.

In Figure 45, the model's predictions closely follow the trend of the ground truth. Here, the green line is a simple slope. However, the model consistently predicts 10°C higher than the green line, resulting in a high MSE of 50.4°C. Apart from a high MSE on profile 60, the results from these experiments were among the best. Profile 45 had the lowest MSE, at only 1.2°C.

In Figure 46, the model struggles initially with estimating the stator temperature, overestimating it by around 7 K. After the 5000th time step, the model accurately predicts spikes in the stator temperature but consistently underestimates the largest spikes by around 5 K. The rotor temperature, on the other hand, is much smoother. However, the model predicts sharp spikes from the stator temperature and inaccurately applies these to the rotor temperature estimates. The MSE of the stator estimation is half that of the rotor estimation at 8.9 K².

In Figure 47, the model's predictions appear to be a combination of the stator and rotor temperature

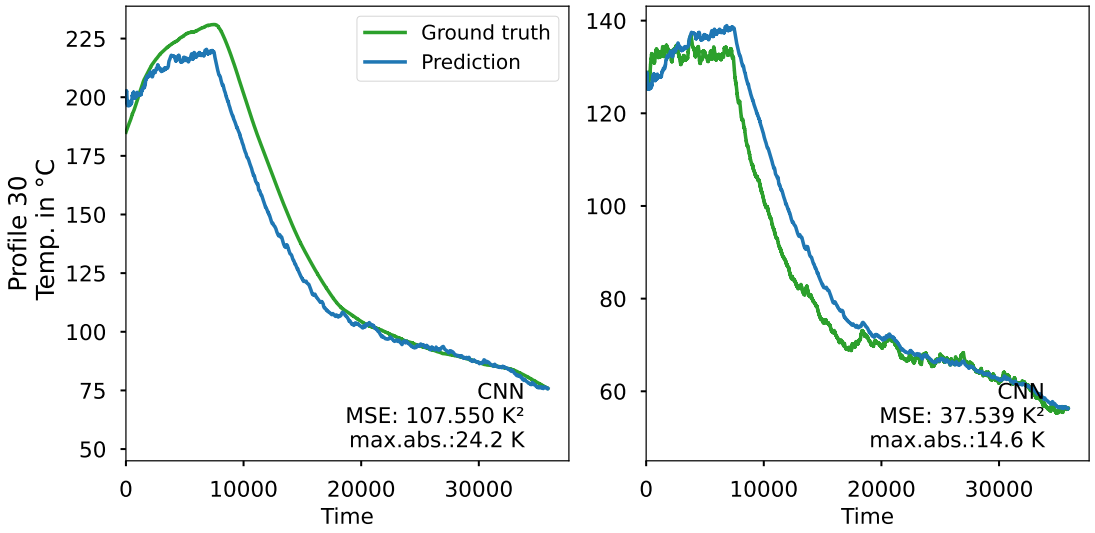


Figure 47. Estimating the rotor and stator temperature using the 1-D CNN model for profile ID 30.

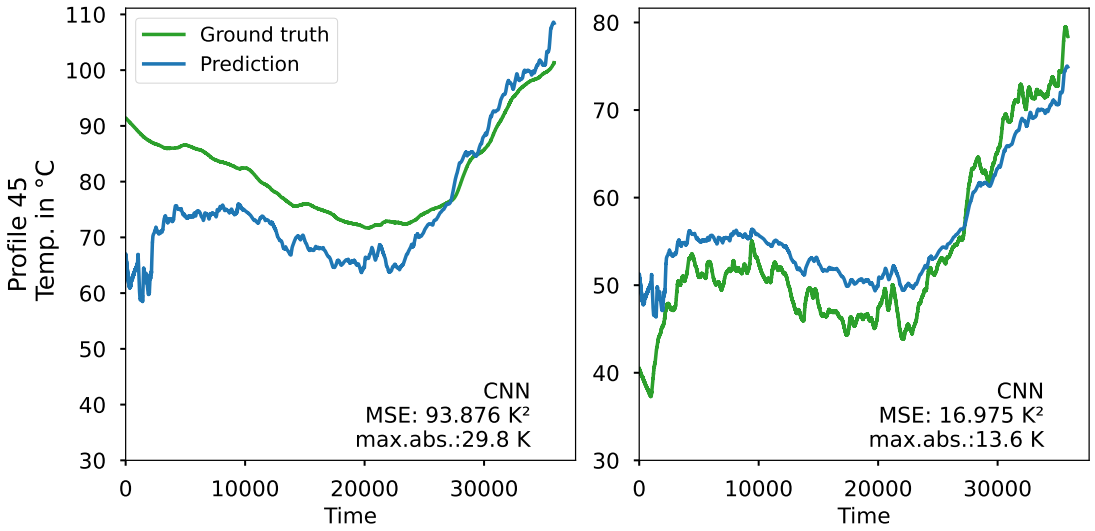


Figure 48. Estimating the rotor and stator temperature using the 1-D CNN model for profile ID 45.

estimates. The rotor temperature continues to rise, reaching around 230 K at the 7000th time step. The actual stator temperature seems to peak at around 130 K, exhibiting small spikes around this plateau. As a result, the model predicts rising small spikes for both the rotor and stator. Following the 7000th time step, the temperature falls steeply. The model consistently overestimates the stator temperature while constantly underestimating the rotor temperature. The ground truth and predictions converge again around the 20,000th time step.

In Figure 48, there is a large gap between the ground truth temperature of the rotor and the predicted temperature, leading to a high maximum absolute error of 29.8 K. The rotor temperature traces a smoother V-shape while the predictions are more rugged, similar to that of the stator, which has a smaller temperature drop than the rotor's notable V-shape. The predictions and ground truth are closest at around the 27,000th time step.

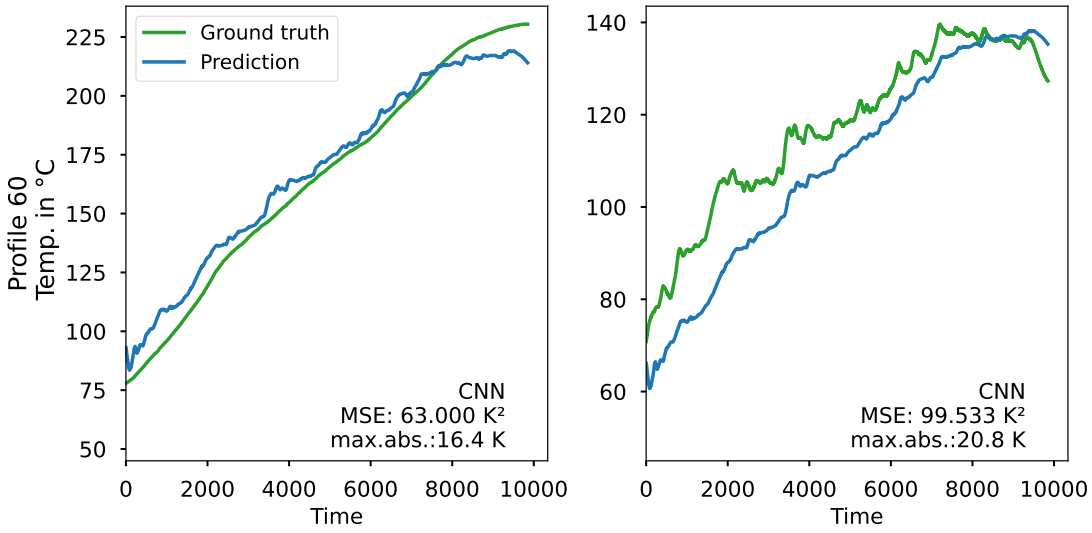


Figure 49. Estimating the rotor and stator temperature using the 1-D CNN model for profile ID 60.

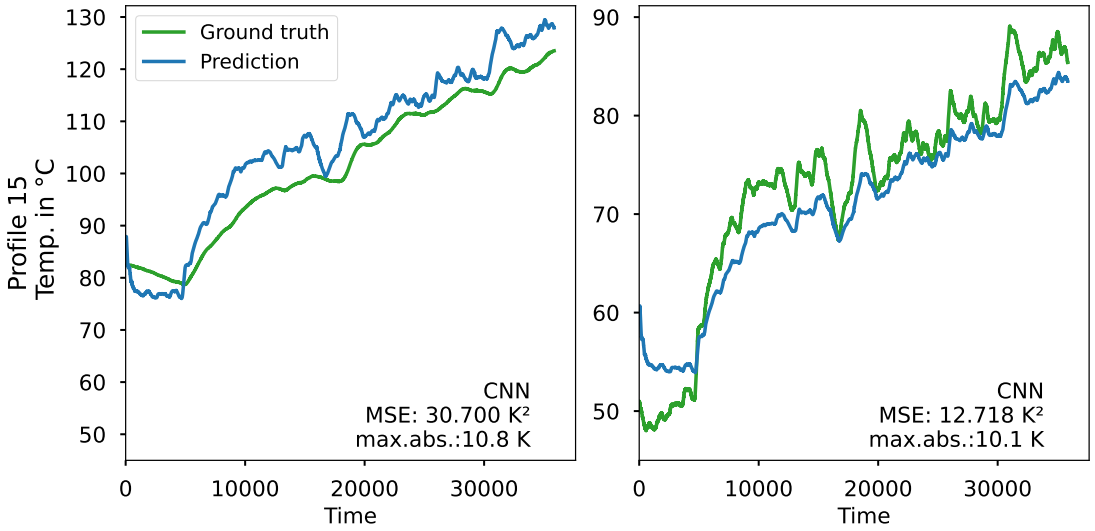


Figure 50. Estimating the rotor and stator temperature using the 1-D CNN model for profile ID 15.

In Figure 49, profile 60 depicts a constant slope which the model either overestimates for the rotor or underestimates for the stator. The model struggles slightly to predict some of the spikes in the stator temperature estimates. The predicted spikes are more subdued than those in the ground truth.

Additionally, we carried out trials where the stator temperature was omitted as an input but treated as an output, as shown in Figures 46, 47, 48, and 49. This allowed us to assess the model's effectiveness under various input setups and analyze how the inclusion or exclusion of stator temperature affects the accuracy of rotor temperature prediction.

In Figure 50, there is a sharp drop around the 5000th iteration, followed by a steady rise. Both the ground truth and the prediction show this pattern, although the prediction starts with a steeper initial drop and slightly lags behind the actual values throughout. The maximum absolute error observed for both the stator and the rotor temperature is 10 K.

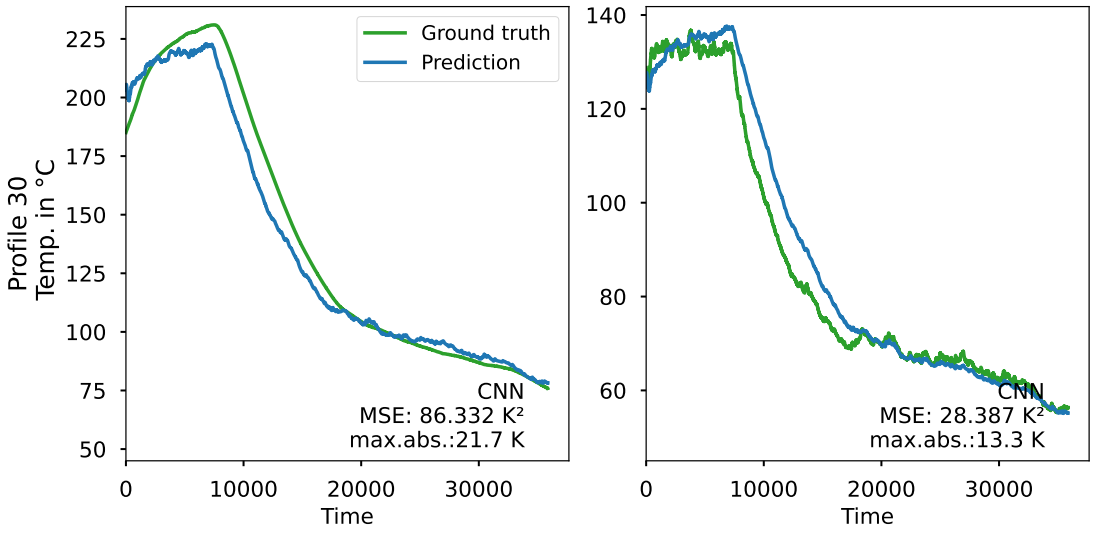


Figure 51. Estimating the rotor and stator temperature using the 1-D CNN model for profile ID 30.

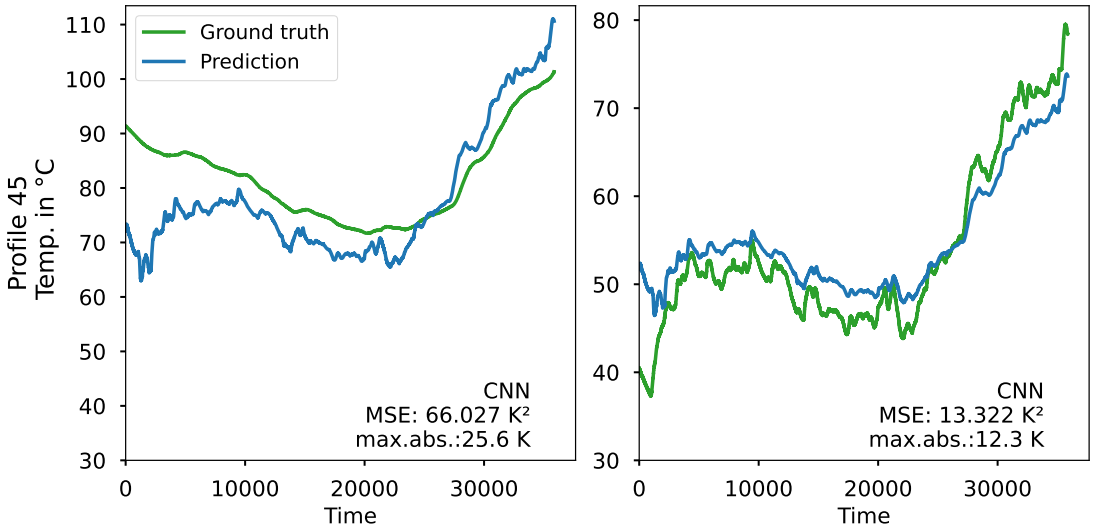


Figure 52. Estimating the rotor and stator temperature using the 1-D CNN model for profile ID 45.

In Figure 51, the MSE for the rotor temperature estimation reaches 86.3 K², indicating a relatively high average squared difference between the predicted values and the actual data. The model tends to underpredict the rotor temperature and overpredict the stator temperature. The ground truth shows a very sharp and almost linear decline, while the prediction initially follows the ground truth closely but then diverges slightly around the middle and tends to flatten as it approaches the 18,000th time step.

In Figure 52, there is a gradual decrease in temperature followed by a significant rise starting around the 20,000th time step. The prediction follows the actual temperature trend loosely, with notable discrepancies especially during the rise where it initially underestimates the rotor temperature. The stator temperature prediction features more fluctuation, displaying a detailed zig-zag pattern with frequent ups and downs before a steep rise in temperature. The model's prediction captures the general trend but misses some of the finer fluctuations seen in the actual temperature. The stator temperature prediction has an MSE of 12.3 K², which is considerably lower than that of the rotor temperature estimate at 66.0K².

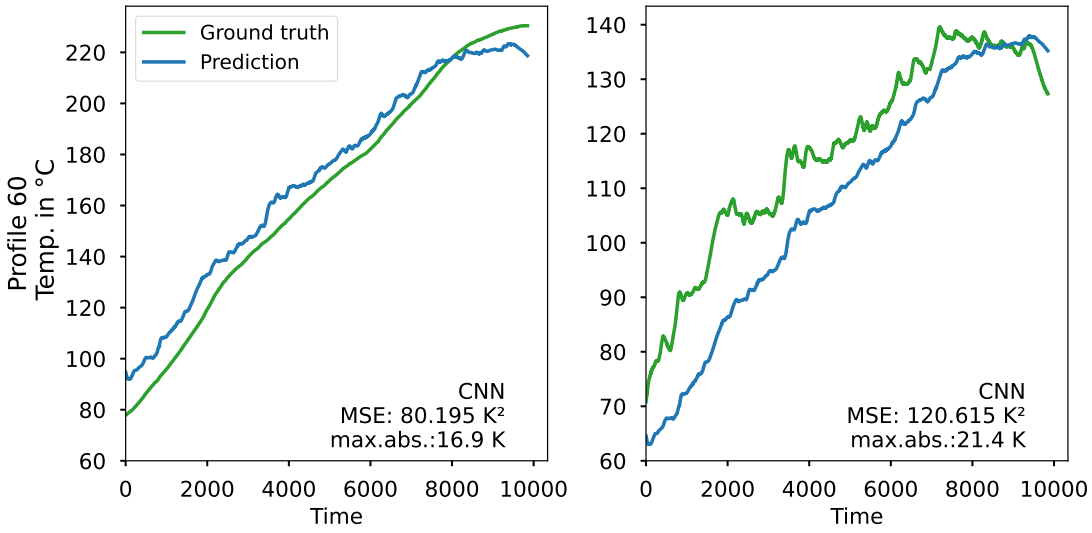


Figure 53. Estimating the rotor and stator temperature using the 1-D CNN model for profile ID 60.

In Figure 53, the model's predictions closely follow the actual temperature trend, with both lines showing a nearly consistent upward trajectory. Minor deviations occur between the predicted and actual values, especially near the peak around 8000 iterations. There are more pronounced fluctuations between the predicted and actual temperatures compared to the rotor temperature predictions, with the model capturing the general trend but with noticeable discrepancies at several points. The stator temperature prediction's MSE is significantly higher at 120.6 K², compared to the rotor's 80.2 K².

5.4 Comparison between Models

To further elucidate the comparative merits and limitations of the MLP, TNN, and 1-D CNN models, we explore a detailed analysis across several critical dimensions. This comparative framework aims to highlight the unique advantages and inherent drawbacks associated with each model, providing a comprehensive overview essential for informed decision-making.

Accuracy and Reliability: The TNN model, designed with integrated heat-transfer principles, exhibited superior accuracy in our tests, achieving the lowest mean squared errors, which highlights its reliability for critical temperature prediction tasks. The TNN shows good results both with and without the stator temperature, while the 1-D CNN performs the best with the stator and worse without it. There are many similarities between the rotor and stator temperatures, which suggests that the model may be learning to mimic the stator temperature pattern to predict the rotor temperature. The MSE and Max Abs for estimating rotor temperature using 1-D CNN were compared with the existing models, as presented in Table 8.

Computational Efficiency: In terms of computational efficiency, the TNN model was notable for its rapid processing capability, demonstrating the shortest training and inference times, making it particularly attractive for real-time applications where speed is paramount.

Table 8

MLP, 1-D CNN, and TNN model results.

Test Profile	MLP Model		1-D CNN Model		TNN Model	
	MSE	Max Abs	MSE	Max Abs	MSE	Max Abs
15	18.1	18.0	1.4	7.1	14.5	8.2
30	64.5	31.1	1.6	11.4	10.9	11.5
45	10.8	13.0	1.2	6.6	5.3	5.5
60	92.4	29.6	50.4	15.1	24.3	11.2

Ease of Implementation: The ease of implementation is a crucial factor in real-world applications, where the simpler architecture of the MLP model offers fewer integration challenges, thus facilitating smoother adoption within existing technological setups.

Adaptability: The TNN model's adaptability to rapidly changing operational conditions underscores its utility in dynamic environments where operational parameters frequently vary, offering significant advantages over more static model types.

This comprehensive analysis of model performance across various dimensions not only aids in understanding the relative strengths and weaknesses of each predictive model but also guides the selection process for normal driving scenarios, ensuring that the chosen model aligns with specific operational requirements and constraints. Therefore, the TNN was chosen for the tests with the normal driving scenario data.

5.5 Testing TNN with Normal Driving Scenario Data

Upon identifying the TNN as the most promising model, further testing was conducted on two additional profiles 87 and 88 which represent data simulating normal driving scenarios. This step was crucial in answering the second research question outlined in Subsection 1.2. The results are shown in Table 9 with additional evaluation metrics, R^2 and RMSE.

Initially, the TNN was evaluated using the stator temperature as an input variable, as demonstrated in Figures 54 and 55. The model performance was then assessed with the stator temperature set as an output, shown in Figures 56 and 57. These experiments reveal a notable difference in prediction accuracy between the stator and rotor temperatures.

In Figures 54 and 55, the TNN closely aligns with the ground truth when predicting stator temperature, achieving a lower MSE. This suggests robustness in handling input features directly related to the stator's conditions. However, as observed in Figures 56 and 57, the MSE significantly increases when the stator is set as an output, especially noticeable in the prediction of rotor temperatures. For example, Figure 57 shows the TNN attaining an MSE of 22.2 K^2 for the rotor temperature, considerably higher than the stator temperature predictions. The TNN's consistent performance in predicting both stator and rotor temperatures, when the stator is used as an output, underscores the model's potential in effectively integrating key thermal dynamics within motor systems.

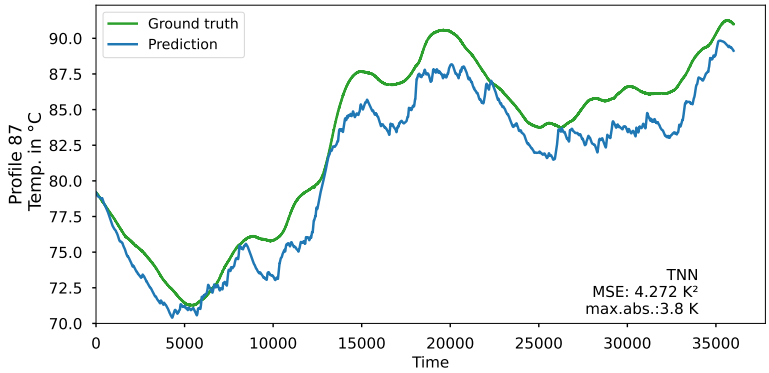


Figure 54. Estimating the rotor temperature using the TNN model for profile ID 87.

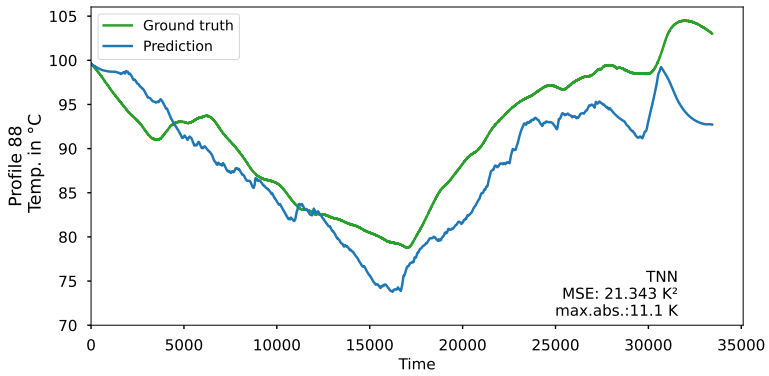


Figure 55. Estimating the rotor temperature using the TNN model for profile ID 88.

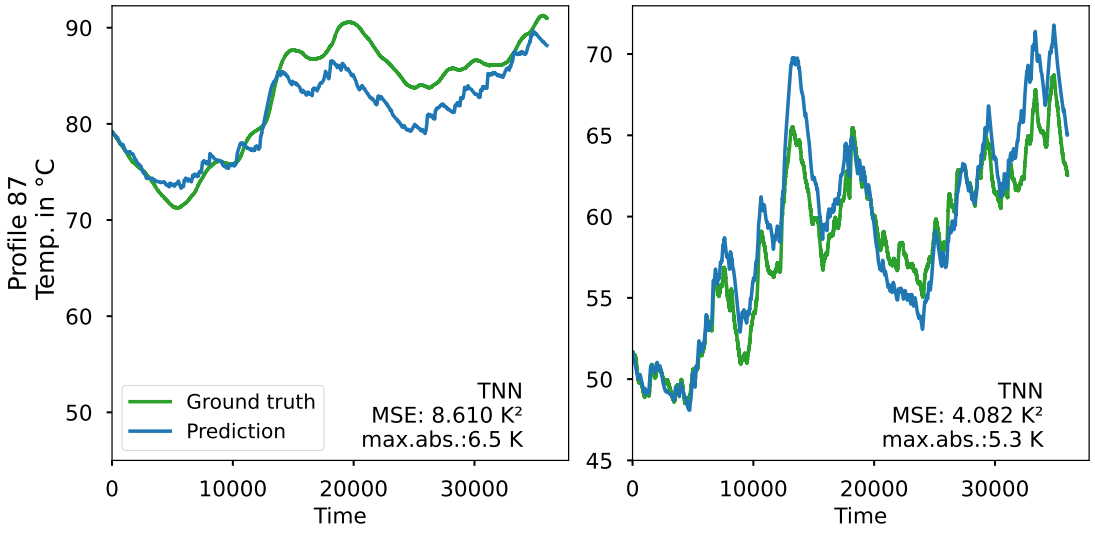


Figure 56. Estimating the rotor and stator temperature using the TNN model for profile ID 87.

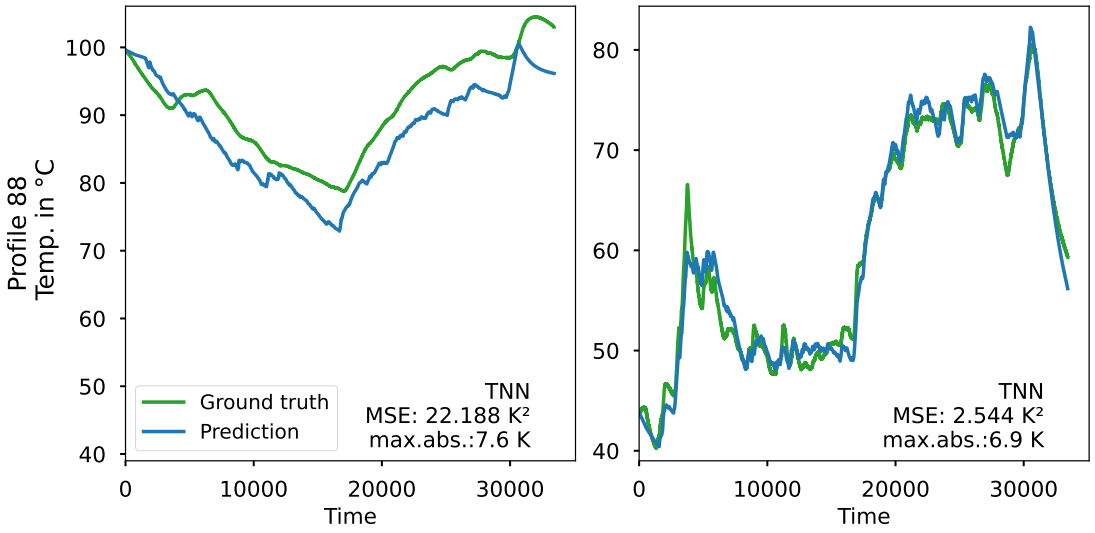


Figure 57. Estimating the rotor and stator temperature using the TNN model for profile ID 88.

Table 9

TNN model results with normal driving scenario data.

Test Profile	Without Stator				With Stator			
	MSE	Max Abs	RMSE	R^2	MSE	Max Abs	RMSE	R^2
87	8.6	6.5	2.9	0.758	4.3	3.8	2.07	0.88
88	22.2	7.6	4.7	0.578	21.3	11.1	4.6	0.59

5.6 Discussion

In revisiting our research question outlined in Subsection 1.2, we have achieved significant insights into the feasibility and effectiveness of employing ML models to accurately predict rotor temperature for IM. Our findings suggest that these models can indeed be trained effectively utilizing measured quantities such as rotor speed, slip speed, current, voltage, cooling, torque, and stator temperature. Moreover, the inclusion of motor housing temperature in/out has shown to enhance the accuracy of rotor temperature estimation.

An analysis of our experiments reveals that incorporating motor housing temperature in/out alongside other measured quantities can compensate for the absence of stator temperature as input. Through further optimization of model hyperparameters and additional training, it is plausible to improve the performance of rotor temperature estimation without relying on stator temperature input. This enhancement not only underscores the potential for cost reduction by eliminating the need for stator sensors but also suggests that ML models could potentially replace deterministic models solely based on stator temperature through rigorous statistical evaluation.

Furthermore, our investigation extends to the practical implementation of these models within an electrical inverter's software platform. While our focus has primarily been on model development and evaluation, a pertinent question remains: can these models be effectively embedded in a live system, respecting hardware and timing constraints? Due to the limited time frame of our thesis, we were unable to conduct comprehensive experiments to address this aspect. However, this represents a promising avenue for future research and application, which Volvo Cars company intends to explore further.

The baseline MLP model has proven capable to predict the rotor and the stator temperature, although, not

as effectively as the following two models. The MSE and maximum absolute error are within tolerable ranges, but would have to be improved to be viable in production use cases. Also, further filtering of the noisy predicted signals might be necessesary.

Regarding the TNN model's results, our experiments, particularly Experiment 3, showcased superior performance compared to other experiments across various test profiles. Results illustrating rotor temperature estimates for profile IDs 11, 15, 30, 45, 60 and 84 demonstrate that the TNN model consistently outperforms Volvo's current model, however, due to company privacy regulations, the comparison results are not shown in the figures. Notably, the TNN model exhibits a closer tracking of fluctuations in ground truth data while maintaining a tighter fit, thereby indicating its efficacy in predicting outcomes with specified initial conditions.

In contrast, the 1-D CNN's results, despite similar experimental setups, showed inferior performance compared to the TNN model, particularly when stator temperature was excluded as an input feature. The 1-D CNN model's reliance on stator temperature as an input suggests that it may be learning to mimic the stator temperature pattern to predict rotor temperature, thereby highlighting potential limitations.

The TNN model was identified as promising and further tested on profiles 87 and 88, representing normal driving scenarios. Results showed the TNN performing well when stator temperature was predicted, particularly in predicting rotor temperatures.

Overall, our findings underscore the promise of ML models. The TNN consistently performed better than other models, in accurately predicting rotor temperature for IM. Indicating its potential in integrating thermal dynamics within motor systems.

6 Conclusion

The primary limitation for effectively utilizing EM and achieving their maximum power density lies in their thermal constraints. These constraints must be carefully adhered to in order to prevent rotor demagnetization and internal short circuits in the stator resulting from overheating.

To ensure the efficient and effective use of deployed EM, it is imperative to monitor their internal temperatures, thereby enabling maximal utilization while avoiding overheating. Physical sensors can be installed within the stator windings of the machine to monitor temperature. However, limitations in accessibility and budgetary constraints often prevent these sensors from monitoring all areas of the stator, leading to uncertainties regarding the detection of hotspots. This challenge is exacerbated for the rotor due to its high-speed moving components. Therefore, it is essential to model rotor temperature in real-time to reduce uncertainties and optimize machine utilization.

Although computational methods such as computational fluid dynamics, heat equation finite element analysis, and high-order LPTN offer high accuracy, they are computationally intensive and cannot operate in real-time. Hence, our predictive models become necessary to address this need.

Aligned with the research questions posed, this study investigated whether a ML model could accurately predict rotor temperatures using measurable quantities such as rotor speed, slip speed, current, voltage, and cooling conditions without relying on stator temperature data. The approach taken involved developing and comparing three types of neural networks: MLP, TNN, and 1-D CNN. Each model was evaluated based on its accuracy and efficiency in predicting rotor temperatures. The findings demonstrated that while each model had its merits, the TNN provided a robust method for temperature prediction due to its integration of heat-transfer principles directly into its architecture, enabling more accurate and reliable predictions.

The analysis and results presented in this study have been based on a comprehensive dataset consisting of 2,876,126 data points and various operational scenarios represented by 87 profile IDs. Gathering this data entailed approximately 82 hours of meticulous effort. The model's training encompassed rotor temperatures ranging from 21.6 to 259.6°C, with 20°C representing idle room temperature and 250°C signifying the rotor's critical threshold.

The research has highlighted and shown that advanced ML models can enhance predictive accuracy and operational efficiency in real-time applications. Key findings from this study reveal that ML models are not only feasible but also offer significant improvements over traditional methods in terms of adaptability and predictive performance. These models successfully predicted rotor temperatures under various operational conditions without the need for direct stator temperature measurements, which is a common limitation in existing systems.

Overall, our findings underscore the promise of ML models. The TNN consistently performed better than other models, in accurately predicting rotor temperature for IM. Indicating its potential in integrating thermal dynamics within motor systems. However, further research is warranted to explore practical implementation challenges and to optimize model performance, potentially leading to significant advancements in the field.

This thesis contributes to the ongoing efforts in the automotive industry to enhance the reliability and efficiency of EM, aligning with broader sustainability and safety objectives. The advancements made here also indicate a promising direction for future research in the field of embedded systems and ML.

By demonstrating the capabilities of ML models, particularly TNN, MLP, and 1-D CNN, we aim to significantly advance predictive maintenance and control of EM, ultimately enhancing safety, efficiency, and cost-effectiveness in industrial operations.

6.1 Limitations

The model was only tested within the Volvo Cars dataset and not implemented in the inverter power module unit. Specifically, the focus was on estimating rotor temperature, as other parameters were deemed sufficiently accurate. While the research aims to enhance rotor temperature estimation, the ultimate goal is to achieve this without relying on stator temperature as input.

Training was conducted using a laptop with 32GB RAM and no GPU, which imposed resource constraints and hindered progress. Discrepancies existed between old and new measurements, with some features such as motor housing temperatures and cooling output being dropped during data concatenation.

The research focused on IM driven by three-phase alternating current, whereas previous studies utilized PMSM with a different target output parameter (permanent magnet temperature) (Kirchgässner et al., 2021) and (Kirchgässner et al., 2023). This divergence in motor types also affected input features, such as using liquid coolant temperature for training, which may not be applicable to IM. However, similarities exist with related studies, which investigated Brushless DC EMs for stator winding temperature prediction (Czerwinski et al., 2021). Despite these differences, the research draws inspiration from shared principles and methodologies, offering insight into tackling various motor-related challenges.

6.2 Future Work

Future work in this domain holds significant potential for further advancements and refinement. Firstly, Volvo Cars should assess the feasibility of replacing their current state observer model with more advanced thermal models, such as the TNN presented in this thesis. Such a transition could lead to improved accuracy and efficiency across various scenarios.

Moreover, it is crucial to evaluate the practical implementation of the developed models within the software inverter. This involves embedding the models and assessing their performance under real-time constraints, ensuring that the model size is acceptable and that computations meet timing criteria, especially regarding safety considerations. Techniques such as quantization or knowledge distillation may be necessary to optimize model size and computational efficiency without compromising accuracy.

Expanding the complexity of the models may require training and testing on larger datasets to ensure robustness and generalization across a wider range of temperatures and motor conditions. Additionally, measurements during the pre-heat phase and multiple revolutions of the motor can provide valuable insights into model performance under diverse operating conditions.

The exploration of advanced ML techniques and algorithms, such as other ML models and quantization methods, holds promise for further enhancing model performance and efficiency. Analyzing parameter distributions and employing techniques like Huffman encoding and trained quantization can significantly reduce memory footprint and inference time (Gajjala et al., 2020; Gholami et al., 2022; Han et al., 2015).

Furthermore, there is ample scope for fine-tuning hyperparameters, optimizing optimizers, and exploring regularization techniques to improve model accuracy and stability. The inclusion of all permutations of chosen input features in hyperparameter tuning can provide deeper insights into the impact of different feature combinations on model performance.

Expanding on future work, it is imperative to consider a wider range of motor states, including scenarios where the motor temperature is negative, to account for colder environments. Training the model with more diverse data will enhance its generalization across all temperature ranges and improve its ability to accurately represent real-world conditions. Additionally, incorporating data points with negative motor temperatures will allow the optimizer to update weights more effectively, leading to better model performance.

Furthermore, when evaluating the accuracy of safe torque estimations, comprehensive testing with varying degrees over extended periods is essential. This exhaustive testing ensures that the better-generalized model performs well across different parameter combinations and can accurately estimate safe torque under various conditions. Additionally, investigating the relationship between rotor temperature and safe torque is crucial, as it may influence safety calculations and monitoring systems.

A reliable estimation of rotor temperature not only aids in safe torque estimation but also has broader applications in safety calculations, monitoring, and data analysis. The ML approach's superior generalization capabilities make it a more reliable option for accurately estimating rotor temperature across diverse operating conditions.

However, it's important to consider the long-term performance of the model, particularly regarding motor degradation over time. While a model may demonstrate accuracy when trained on a new motor, its predictive capability may diminish over the motor's lifespan. Motor life expectancy varies greatly and can range from a few years to several decades, depending on various factors. Therefore, regular retraining of the model or maintaining different models corresponding to the motor's lifetime may be necessary to ensure continued accuracy and reliability.

This work lays the foundation for further exploration and improvement in every aspect of the process, including hardware, software, data, models, and desired outcomes. The framework established in this thesis opens doors for future research in microcontroller frameworks, alternative models, traditional methods, optimizers, hyperparameters, datasets, and data preparation techniques. It serves as a guide for researchers and practitioners to delve deeper into these areas, ultimately advancing the field of thermal modeling for induction motors. The practical insights gained from applying the TNN model to actual rotor temperature scenarios suggest new directions for enhancing the model's theoretical basis, particularly in handling variable operational environments.

References

Al-Gabalawy, M., Elmetwaly, A. H., Younis, R. A., & Omar, A. I. (2024). Temperature prediction for electric vehicles of permanent magnet synchronous motor using robust machine learning tools. *Journal of Ambient Intelligence and Humanized Computing*, *15*(1), 243–260.

Altares, E. (2003). *Elementary statistics : A modern approach' 2003 ed.* Rex Book Store. https://books.google.se/books?id=52_CgfJwWZQC

Czerwinski, D., Gęca, J., & Kolano, K. (2021). Machine learning for sensorless temperature estimation of a bldc motor. *Sensors*, *21*(14), 4655.

Gajjala, R. R., Banchhor, S., Abdelmoniem, A. M., Dutta, A., Canini, M., & Kalnis, P. (2020). Huffman coding based encoding techniques for fast distributed deep learning. *Proceedings of the 1st Workshop on Distributed Machine Learning*, 21–27.

Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., & Keutzer, K. (2022). A survey of quantization methods for efficient neural network inference. In *Low-power computer vision* (pp. 291–326). Chapman; Hall/CRC.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning* [http://www.deeplearningbook.org]. MIT Press.

Guo, H., Ding, Q., Song, Y., Tang, H., Wang, L., & Zhao, J. (2020). Predicting temperature of permanent magnet synchronous motor based on deep neural network. *Energies*, *13*(18), 4782.

Han, S., Mao, H., & Dally, W. J. (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*.

Hosseini, S., Shahbandegan, A., & Akilan, T. (2022). Deep neural network modeling for accurate electric motor temperature prediction. *2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 170–175.

Hughes, A., & Drury, B. (2019). *Electric motors and drives: Fundamentals, types and applications*. Newnes.

Hughes, R., Haidinger, T., Pei, X., & Vagg, C. (2023). Real-time temperature prediction of electric machines using machine learning with physically informed features. *Energy and AI*, *14*, 100288.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*.

Incropera, F., Bergman, T., DeWitt, D., Lavine, A., Seetharamu, K., & R, S. (2011). *Fundamentals of heat and mass transfer*. Wiley. https://books.google.se/books?id=g6SYoAEACAAJ

Iso 26262-1:2018 [[Pages 2 and 22.]]. (2018, December). https://www.iso.org/standard/68383.html

Kim, S.-H. (2017). *Electric motor control: Dc, ac, and bldc motors*. Elsevier.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1d convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, *151*, 107398. https://doi.org/https://doi.org/10.1016/j.ymssp.2020.107398

Kirchgässner, W., Wallscheid, O., & Böcker, J. (2020).
Estimating electric motor temperatures with deep residual machine learning.
IEEE Transactions on Power Electronics, *36*(7), 7480–7488.

Kirchgässner, W., Wallscheid, O., & Böcker, J. (2021). Data-driven permanent magnet temperature estimation in synchronous motors with supervised machine learning: A benchmark.
IEEE Transactions on Energy Conversion, *36*(3), 2059–2067.

Kirchgässner, W., Wallscheid, O., & Böcker, J. (2023). Thermal neural networks: Lumped-parameter thermal modeling with state-space machine learning.
Engineering Applications of Artificial Intelligence, *117*, 105537.

Kumar, R. (2019). *Machine learning quick reference: Quick and essential machine learning hacks for training smart data models*. Packt Publishing.
https://books.google.se/books?id=BmSGDwAAQBAJ

Lamb, H. (1883). Xiii. on electrical motions in a spherical conductor.
Philosophical Transactions of the Royal Society of London, (174), 519–549.

Madhavan, S., Devdatta PB, R., Konda, Y. R., Gundabattini, E., Mystkowski, A., Palka, R., Wardach, M., & Prajzendanc, P. (2023). Thermal management analyses of induction motor through the combination of air-cooling and an integrated water-cooling system.
Scientific Reports, *13*(1), 10125.

MATLAB. (2024). Deep Learning Toolbox - matlab [Accessed: 2024-04-12].
https://se.mathworks.com/products/deep-learning.html

Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 25).
Determination press San Francisco, CA, USA.

Odvářka, E., Brown, N. L., Mebarki, A., Shanel, M., Narayanan, S., & Ondrůšek, C. (2010).
Thermal modelling of water-cooled axial-flux permanent magnet machine.

Riddiough, R., & Thomas, D. (1998). *Statistics for higher mathematics*. Nelson.
https://books.google.se/books?id=qSaRjcX6oRoC

Šare, A., Krajačić, G., Pukšec, T., & Duić, N. (2015). The integration of renewable energy sources and electric vehicles into the power system of the dubrovnik region.
Energy, Sustainability and Society, *5*, 1–16.

Simoes, F. N. F. Q., Dastani, M., & van Ommen, T. (2024).
Causal entropy and information gain for measuring causal control.

Specht, A., & Böcker, J. (2010). Observer for the rotor temperature of ipmsm. *Proceedings of 14th International Power Electronics and Motion Control Conference EPE-PEMC 2010*, T4–12.

Specification, C. (1991). Bosch. *Robert Bosch GmbH, Postfach*, *50*, 15.

Talaei Khoei, T., Ould Slimane, H., & Kaabouch, N. (2023).
Deep learning: Systematic review, models, challenges, and research directions.
Neural Computing and Applications, *35*(31), 23103–23124.

Ullah, Z., Lodhi, B. A., & Hur, J. (2020). Detection and identification of demagnetization and bearing faults in pmsm using transfer learning-based vgg. *Energies*, *13*(15), 3834.

United Nations. (n.d.). *Sustainable development goal 9* [[Page 22.]]. https://sdgs.un.org/goals/goal9

Vittal, V., McCalley, J., Anderson, P., & Fouad, A. (2019). *Power system control and stability*. Wiley.
https://books.google.se/books?id=obCuDwAAQBAJ

Wallscheid, O., & Böcker, J. (2015). Global identification of a low-order lumped-parameter thermal network for permanent magnet synchronous motors.
IEEE Transactions on Energy Conversion, *31*(1), 354–365.

Wolff, S., Brönner, M., Held, M., & Lienkamp, M. (2020). Transforming automotive companies into sustainability leaders: A concept for managing current challenges.
Journal of Cleaner Production, *276*, 124179.

Yadav, P. K., Prabhakaran, M., et al. (2023). Temperature prediction of permanent magnet synchronous motor using ai techniques for effective traction control. *2023 First International Conference on Advances in Electrical, Electronics and Computational Intelligence (ICAEECI)*, 1–8.

Zhao, J., Mao, X., & Chen, L. (2019).
Speech emotion recognition using deep 1d & 2d cnn lstm networks.
Biomedical signal processing and control, *47*, 312–323.

Appendices

Appendix A

A.1 Concatenate Dataset

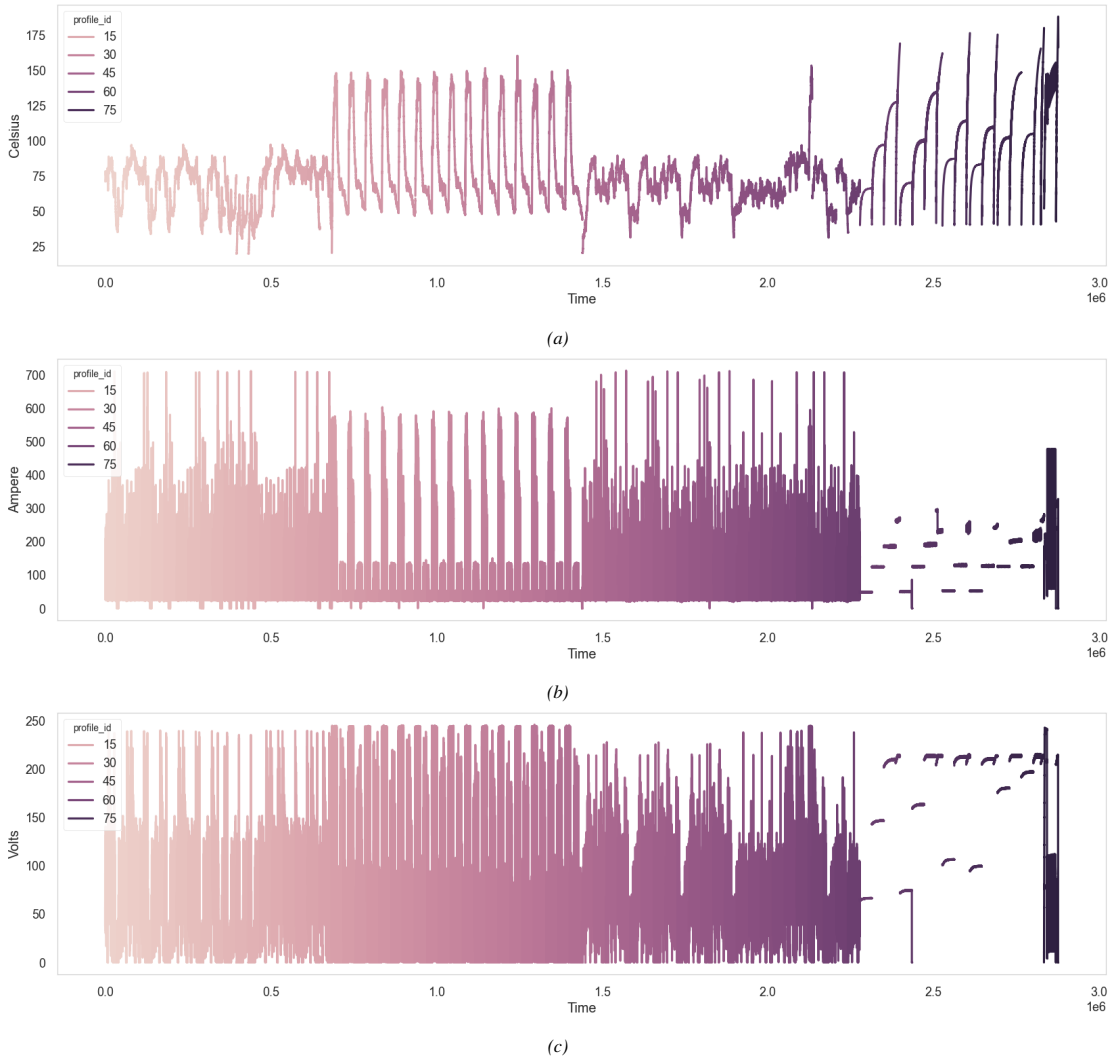


Figure A.1. A plot showing the (a) Coolant Temperature (b) Winding Temperature (c) Current (d) Voltage. Each time column represents 0.1 seconds.

A.2 Old Measurement Correlation

Figure A.2. A heatmap showing the correlation of the input features for the new measurements.