CHALMERS
UNIVERSITY OF TECHNOLOGY

Virtual sensor - AI model training using VOLVO Brake temperatures

Armin Alami Alamdari
Birtukan Fozzati

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's Thesis 2025

Virtual sensor - AI model training using VOLVO Brake temperatures

Armin Alami Alamdari
Birtukan Fozzati

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mechanics and Maritime Sciences
Chalmers University of Technology
Gothenburg, Sweden 2025

Virtual sensor - AI model training using VOLVO Brake temperatures
Armin Alami Alamdari
Birtukan Fozzati

© Armin Alami Alamdari, Birtukan Fozzati, 2025.

Supervisor at Volvo GTT: Martin K Petersson
Supervisor at Chalmers: Tore Vernersson, Department of Mechanics and Maritime Sciences
Examiner: Roger Lundén, Department of Mechanics and Maritime Sciences

Master's Thesis 2025
Department of Mechanics and Maritime Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: VOLVO CAD rendering of ventilated brake disc and brake pads

Typeset in LaTeX
Gothenburg, Sweden 2025

Armin Alami Alamdari
Birtukan Fozzati
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

Accurate prediction of brake disc temperatures in heavy-duty vehicles is essential for ensuring safety, reducing wear and improving braking performance. Excessive heat buildup in the disc can lead to brake fade, accelerated material degradation and increased emissions of harmful wear particles. This thesis focuses on predicting brake disc temperatures using time-series data collected from controlled dynamometer tests. The dataset includes braking signals such as torque, pressure and speed, recorded at high frequency under a wide range of operating conditions. Various machine learning models, including neural networks, were developed to predict brake disc temperatures during individual braking events. This work serves as a foundation for future efforts to extend temperature prediction models to real-world field data and ultimately support the development of intelligent thermal monitoring systems that can reduce brake wear, improve safety and help meet upcoming Euro 7 regulations on particle emissions from braking systems.

KEYWORDS: Brake Disc Temperature, Machine Learning, Gated Recurrent Unit, Thermal Modelling, Heavy-Duty Vehicles, Real-Time Monitoring

Preface

This report presents the outcome of our master's thesis project which is a collaboration between the Department of Mechanics and Maritime Sciences at Chalmers University of Technology and Volvo Group Trucks Technology, during the spring of 2025.

Acknowledgements

We would like to express our profound gratitude and appreciation to the individuals who have provided invaluable assistance, guidance and support throughout the completion of our master's thesis.

We are especially thankful to our supervisors. Martin Petersson at Volvo Group Trucks Technology played a central role in shaping the structure and direction of this work through his consistent involvement, technical expertise and ongoing support. Equally, we are deeply grateful to Associate Professor Tore Vernersson at Chalmers for his expert guidance, thoughtful feedback and dedicated commitment throughout the thesis process. We would also like to acknowledge and thank our examiner Professor Roger Lundén.

Our sincere thanks go to Zafer Yüce and Vineet Kothari, consultants at Volvo GTT, for their valuable input, constructive suggestions and continued willingness to discuss both technical and strategic aspects of the project. A special mention to Marilou Bagge of Volvo GTT, whose contributions and support were instrumental during the thesis.

We also acknowledge Volvo Group for providing the resources and an encouraging environment that enabled the successful execution of this project.

Use disclaimer regarding AI language tools

We take full responsibility for the content of this thesis work and we are able to justify all the choices made. With that being said, AI was strictly used for correcting spelling, grammar and syntax mistakes.

Armin Alami Alamdari and Birtukan Fozzati, Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AI	Artificial Intelligence
Adam	Adam Moment Estimation
AdamW	Adaptive Moment Estimation with decoupled Weight decay
AMC	Aluminum Matrix Composite
ANN	Artificial Neural Network
CAD	Computer Aided Design
CNN	Convolutional Neural Network
Dyno lab	Dynamometer Laboratory
FE	Finite Element
FFNN	Feedforward Neural Network
GPR	Gaussian Process Regression
GRU	Gated Recurrent Unit
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
MSE	Mean Squared Error
PDE	Partial Differential Equation
PINN	Physics Informed Neural Network
PM	Particle Matter
ReLu	Rectified Linear Unit
RL	Reinforcement Learning
RMSprop	Root Mean Square Propagation
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
R^2	Coefficient of Determination
SGD	Stochastic Gradient Descent
SVM	Support Vector Machines
wMAPE	weighted Mean Absolute Percentage Error

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background

Volvo Trucks is committed to advancing energy efficiency and reducing the environmental impact of heavy-duty vehicles. A particular focus is placed on minimizing both fossil fuel usage and brake particle emissions, in line with evolving European legislation such as Euro 7, which introduces strict limits on non-exhaust emissions including particulate matter (PM2.5) from braking systems.

Friction-based braking systems, though essential for vehicle safety, are a primary source of wear particles due to mechanical contact between brake pads and discs. In heavy-duty disc brake systems, thermal energy generated during braking causes gradual wear and the release of fine particles. These emissions pose environmental and public health concerns, especially in urban and densely trafficked areas.

As heavy-duty vehicles increasingly incorporate electrification and regenerative braking, the challenge becomes optimizing the interplay between regenerative and traditional friction braking. This optimization is critical not only for efficiency and brake wear reduction but also for thermal safety, particularly in demanding conditions such as downhill braking or emergency stops.

To address these challenges, Volvo Trucks has commissioned this thesis project to investigate the use of machine learning (ML) models, particularly neural networks, for the prediction of brake disc temperatures. The aim is to improve understanding and control of thermal behavior under real-world braking conditions. The project leverages time-series data from Volvo's dynamometer laboratory (dyno lab) and field measurements, with the ultimate goal of supporting efficent and more sustainable braking strategies in future vehicles.

1.2 Purpose

This thesis aims to answer the following research question:

- *How accurately can an AI model predict brake disc temperatures in Volvo heavy-duty vehicles based on experimental data?*

1.3 Goals

The goal of this thesis is to develop AI models that predict brake disc temperatures in Volvo heavy-duty vehicles using experimental data from dynamometer testing. These models serve as a foundation for future work on early detection of thermal overload and more efficient brake system development.

The project focuses on leveraging neural networks trained on time-series sensor data to learn temperature dynamic behaviour under varying braking conditions. By improving prediction accuracy, the models may contribute to better brake system design, support vehicle electrification strategies, and help reduce brake wear particle emissions in line with Euro 7 regulations. The intended outcome is a set of AI models capable of accurately estimating brake disc temperatures from experimental input data. These models are designed to support intelligent brake system development and possible integration into virtual validation workflows.

1.4 Limitations and Demarcations

Several limitations must be considered for the scope of this thesis:

- The dataset used originates from a specific disc–pad combination tested in the Volvo dynamometer laboratory. The model's generalizability to other brake configurations or material types is not assessed.
- The data employed is from temperature sensors embedded within the disc, that do not measure the surface temperature. This introduces an offset between the model's predictions and the higher temperature that is expected at the friction interface.
- Convective cooling is for dyno lab conditions and this could be different to real-world airflow around the disc. This leads to different cooling rates compared to on-road conditions, potentially affecting the accuracy and generalizability of the temperature predictions.
- The data only includes active braking events, i.e when braking is performed, without separate sequences for post-braking cooling phases. This means that the model does not explicitly learn the cooling dynamics outside of braking. The reason is that there are no measured data for the time between active braking events.

2
Theory

2.1 Braking Systems in Heavy-Duty Vehicles

Braking systems are critical to vehicle safety and operational efficiency, particularly in heavy-duty applications where vehicle mass and kinetic energy levels are significantly higher than in passenger transport. The design and selection of appropriate braking technologies are governed by constraints related to heat dissipation capacity, durability under prolonged use and integration with auxiliary safety systems. Broadly, braking systems in heavy-duty vehicles can be classified into three main categories: service brakes, auxiliary brakes and parking/emergency brakes.

2.1.1 Service Brakes

Service brakes are the primary braking system, responsible for maintaining vehicle speed downhill, for routine deceleration and stopping of the vehicle. Two dominant technologies are used in this domain: drum brakes and disc brakes.

Drum brakes have been widely employed in heavy-duty vehicles due to their mechanical robustness, low manufacturing cost, and resistance to harsh conditions [1]. These systems function by pressing brake shoes against the interior surface of a rotating drum. Although their enclosed geometry restricts airflow and limits heat dissipation, leading to elevated temperatures, brake fade, and increased wear under sustained or high-energy braking, it also offers protection against water, dust and debris. This makes drum brakes particularly suitable for off-road or dusty environments where particulate contamination can compromise more exposed systems.

Disc brakes, in contrast are now the dominant technology in modern heavy-duty braking systems, particularly in European and high-performance commercial vehicles [1]. Disc brakes apply clamping forces via calipers and pads onto a rotating disc exposed to ambient air, which facilitates superior convective and radiative heat transfer to the surrounding. Figure 2.1 shows a CAD rendering of a ventilated brake disc and two brake pads. Their open structure enables faster thermal cycling, more uniform temperature distribution, and shorter recovery times following peak loading. This is especially advantageous during downhill braking, repeated urban stop cycles, or emergency braking scenarios. Many systems employ ventilated discs to further enhance cooling through increased surface area and internal airflow. Both drum and disc brakes in heavy-duty vehicles are typically actuated pneumatically rather than hydraulically.

Figure 2.1: Volvo CAD rendering of ventilated brake disc and brake pads

2.1.2 Auxiliary and Parking/Emergency Brakes

In heavy-duty vehicles, auxiliary brakes play an important role in reducing the thermal load on the service brakes. These systems are not designed to bring the vehicle to a full stop, but to provide sustained deceleration, especially on long downhill descents. The two main categories of auxiliary braking are engine brakes and retarders. Engine braking is commonly implemented via compression release mechanisms, which modify the engine's valve timing to create internal resistance [2]. This transforms the engine into an air compressor, converting kinetic energy into heat within the engine system rather than the braking surfaces. This approach offers effective deceleration without engaging friction-based service brakes.

Retarders provide continuous braking torque through additional mounted devices and are common in heavy-duty trucks [3]. Hydrodynamic retarders, often mounted to the driveline or integrated within the transmission, operate by shearing a working fluid between rotating and stationary vanes, producing resistance through fluid dynamics. Electromagnetic retarders use eddy current induction in a rotating disc to generate braking force, offering precise control with minimal wear, although at higher cost and weight [4]. The consumed kinetic energy is converted into heat and dissipated through the retarder's cooling system.

Parking and emergency brake systems help immobilizing the vehicle when stationary and provide a backup in the event of service brake failure [5]. These systems typically use spring-actuated, pneumatically released mechanisms mounted on the

rear axles. When air pressure is lost in the pneumatic service brake system, the spring force automatically engages the brakes as a back-up in emergency situations. Activation is mechanical or electronic, depending on the system design. Modern trucks increasingly incorporate electronic parking brake systems, which enhances electrification, a common theme in the industry [6]. These systems support advanced features such as automatic application during engine shutdown, hill-start assistance and diagnostic capabilities, contributing to improved safety.

2.2 Loading and Temperatures in Disc Brakes

Disc brakes in heavy-duty vehicles experience complex thermal and mechanical loads during operation. This section outlines the fundamental mechanisms of heat generation, material wear and cooling.

2.2.1 Heat Generation and wear

When a vehicle decelerates, its kinetic energy is converted into heat by the brake system. Assuming no regenerative braking or energy losses, the total brake energy generated by the service brakes during a stop is:

$$E_b = \frac{1}{2}m(V_1^2 - V_2^2) \tag{2.1}$$

where m is the vehicle mass, and V_1, V_2 are the initial and final speeds. If the braking event lasts for a duration t_b, the average braking power becomes:

$$P_b = \frac{E_b}{t_b} \tag{2.2}$$

Assuming constant deceleration, the instantaneous braking power as a function of time becomes:

$$P_b(t) = ma(V_1 - at) \tag{2.3}$$

where $a = \frac{V_1 - V_2}{t_b}$ is the uniform deceleration during the stop. This expression captures the linear decrease in power dissipation over the braking interval.

Frictional heat generation:

To determine the heat flux on both the pad and the disc under the assumptions of uniform pressure distribution and uniform wear according to [1], one should begin with the heat generation over an infinitesimal area $dA = \phi_0 r\, dr$, as illustrated in Figure 2.2. The heat generation rate is given by:

$$\mathrm{d}\dot{E} = \mathrm{d}P = V\, \mathrm{d}F_t = \omega \mu p \phi_0 r^2\, \mathrm{d}r \tag{2.4}$$

Furthermore, the total heat generated is distributed between the pad and the disc according to:

$$\mathrm{d}\dot{E} = \mathrm{d}\dot{E}_{\mathrm{pad}} + \mathrm{d}\dot{E}_{\mathrm{disc}} = (1-\sigma)\mathrm{d}P + \sigma\mathrm{d}P \tag{2.5}$$

where σ is the heat partitioning factor (dimensionless) that represents the fraction of the total heat dissipated into the disc.

The heat flux into the pad, over the area $S_{\mathrm{pad}} = dA = \phi_0 r dr$, is determined as:

$$q_{\mathrm{pad}}(r,t) = \frac{\mathrm{d}\dot{E}_{\mathrm{pad}}}{\mathrm{d}S_{\mathrm{pad}}} = (1-\sigma)\mu p r \omega(t) \tag{2.6}$$

Similarly, the heat flux into the disc, over the area $S_{\mathrm{disc}} = dA = 2\pi r dr$, is given by:

$$q_{\mathrm{disc}}(r,t) = \frac{\mathrm{d}\dot{E}_{\mathrm{disc}}}{\mathrm{d}S_{\mathrm{disc}}} = \frac{\phi_0}{2\pi}\sigma\mu p r \omega(t) \tag{2.7}$$

The boundary conditions governing braking scenarios with constant deceleration have been detailed in previous studies, including those by Talati and Jalalifar [7] and Mazidi et al. [8].

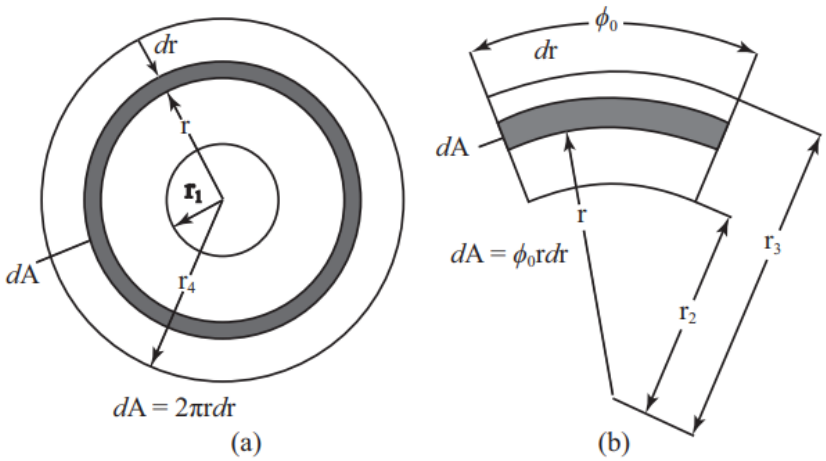


Figure 2.2: Contact surface element of the disc (a) and the pad (b) [8].

Archard's law [9] describes the volume of material worn away due to sliding contact between surfaces. The equation is given by:

$$V = \frac{KWL}{H} \tag{2.8}$$

where:

- V is the wear volume (m^3),
- K is the dimensionless wear coefficient,
- W is the normal load (N),
- L is the sliding distance (m),
- H is the hardness of the softer material (Pa).

This law states that the wear volume is directly proportional to the applied load and sliding distance while being inversely proportional to the hardness of the material. It is widely used in tribology to estimate wear rates in mechanical systems.

2.2.2 Uniform wear vs uniform pressure

The pressure distribution between the brake pad and disc can be modeled under two primary assumptions: uniform pressure and uniform wear. Uniform pressure assumes a constant contact pressure across the entire friction surface. If valid, it's typically for new brake pads or during short braking durations.

Uniform wear, on the other hand, reflects the pressure profile after prolonged use, assuming the product of pressure and sliding velocity is constant across the radius [1]. This leads to a pressure distribution that is inversely proportional to the radius and more accurately captures long-term wear behavior, as shown in Figure 2.3. The choice between these assumptions significantly affects analytical models of braking, particularly for calculating heat generation, wear rates, and thermal stresses.

No brake is ever in a truly steady-state operational condition all the time and despite best efforts of brake designers, some radial variation of the friction material at the friction interface will occur [10]. The effect is that the pad will tilt slightly and the pressure distribution will no longer be uniform, but under normal usage this will not be excessive.

This non-uniform pressure distribution alters how heat is introduced into the disc. Regions experiencing higher contact pressure generate more frictional heat, particularly when sliding speeds are also elevated. This leads to uneven heating across the disc surface, which introduces uncertainty in the temperature measurements used for modelling, especially since sensors only capture temperature at specific locations.

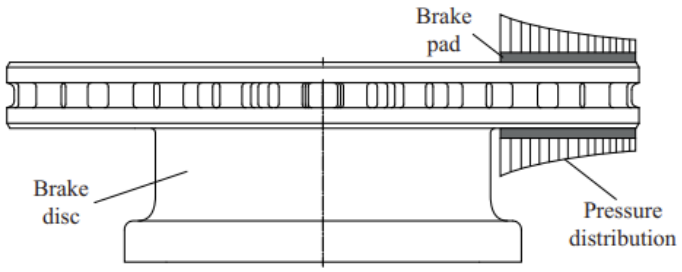


Figure 2.3: Pressure distribution on the disc for uniform wear [1].

2.2.3 Brake Disc Cooling

The heat created from braking must be effectively dissipated to prevent thermal fade, material degradation, or uneven wear. The primary cooling mechanisms involved are conduction, convection and radiation, each playing a distinct role in the thermal cycle of the disc.

Conduction governs the internal redistribution of heat within the brake disc. Upon pad contact, heat is first concentrated near the friction surface and then spreads into the rotor volume [11]. While this process does not remove heat from the braking

system, it plays a crucial role in reducing local temperature peaks and managing thermal gradients across the disc. The efficiency of this mechanism depends on the thermal conductivity of the disc material and the rotor geometry and can be explained using Fourier's law on heat transfer.

The axisymmetric thermal heat transfer equation is written as:

$$\rho c \frac{\partial T}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}\left(r k_r \frac{\partial T}{\partial r}\right) + \frac{\partial}{\partial z}\left(k_z \frac{\partial T}{\partial z}\right) \tag{2.9}$$

where ρ, c, k_r, and k_z are the density, the specific heat, and the thermal conductivity in the radial (r) and depth (z) directions of the material, respectively. The boundary and initial conditions used are:

$$T = T^* \quad \text{on } \Gamma_0 \tag{2.10}$$

$$q_n = q_c - q_r \text{ on } \Gamma_1 \tag{2.11}$$

$$q_n = q_n^* \quad \text{on } \Gamma_2 \tag{2.12}$$

$$T = T_0 \quad \text{at time } t = 0 \tag{2.13}$$

Here, T^* is the prescribed temperature, q_c is cooling by convection, q_r is cooling by radiation. q_n^* is the heat flux due to friction at each contact interface, T_∞ is the ambient temperature, and T_0 is the initial temperature. The boundaries Γ_0, Γ_1, and Γ_2 are where temperature, combined convection and radiation, and frictional heat flux are imposed, respectively.

Convection is the dominant mode of heat dissipation from discs and drums [11]. It is typically modeled using Newton's Law of Cooling:

$$q_\mathrm{c} = h(T - T_\infty) \tag{2.14}$$

where h is the convective heat transfer coefficient, T is the disc surface temperature, and T_∞ is the ambient air temperature. This expression quantifies the rate of heat transfer from the disc to the surrounding air as a function of the temperature difference and the efficiency of the convective process.

In the case of disc brakes, as the disc rotates, it transfers heat to the surrounding air through external surface exchange and internal airflow within ventilated vanes [11]. This form of forced convection is strongly influenced by vehicle speed, disc openness, and vane geometry. Ventilated discs, which are widely used in heavy-duty braking systems, incorporate internal channels that increase surface area and promote centrifugal airflow during rotation, significantly enhancing cooling performance.

According to classical convective heat transfer theory [12], the heat transfer coefficient scales approximately with the square root of flow velocity, which in the context of vehicle braking corresponds to wheel speed. Therefore, higher driving speeds typically result in more efficient convective cooling. When the vehicle is stationary after braking, however, the system relies on natural convection driven by

buoyancy rather than airflow, reducing cooling rate substantially [11].

Radiation contributes a smaller share of the total heat dissipation at moderate temperatures, but becomes increasingly relevant at elevated thermal loads [11]. Radiative losses scale with the fourth power of absolute temperature and are particularly significant in systems operating above 500°C, which can occur during repeated emergency braking or long downhill descents. Although often overlooked in engineering estimations, radiative cooling is a non-negligible contributor. According to Day [11], "even at the relatively low temperature of 100°C, its contribution to brake cooling is similar to that of convective cooling by natural convection".

Radiative cooling is defined as:

$$q_{\mathrm{r}} = \sigma\epsilon(T^4 - T_\infty^4) \tag{2.15}$$

where q [W/m^2] is the heat flux transferred by radiation, σ is the Stefan-Boltzman constant and ϵ the emissivity coefficient, T (in [K]) is the average disc surface temperature and T_∞ (in [K]) the ambient air temperature.

2.3 Limitations of brake systems in Heavy Duty Vehicles

One of the critical challenges in heavy-duty truck braking is the risk of braking system failure after excessive use. This phenomenon, commonly observed in long downhill descents or during frequent braking cycles in urban driving, is primarily caused by a combination of air brake system limitations and thermal effects on the braking components.

Most heavy-duty trucks, including Volvo Trucks, rely on compressed air brake systems rather than hydraulic systems, as used in passenger vehicles. While air brakes provide high braking force and reliability, they require a continuous supply of compressed air to function [13]. If a truck is subjected to repeated or prolonged braking, the air reservoir may deplete faster than the compressor can replenish it. As a result, air pressure in the brake lines drops, leading to delayed braking response. In extreme cases it could lead to brake system failure when pressure falls below the operational threshold. When air pressure reaches a critical level, the brakes may no longer engage properly or could remain locked due to insufficient force to release them. This situation is particularly hazardous in long-haul transport and mountainous terrain, where continuous braking may be necessary.

In addition to air supply limitations, excessive braking generates significant heat, which can lead to thermally induced brake fade. As the brake pads and disc or drum brake linings heat up due to continuous friction, their coefficient of friction decreases, reducing braking efficiency [14]. As a result, brake components may reach temperatures where they no longer provide sufficient stopping power, leading to dangerously extended braking distances. In severe cases, brake fade may cause a complete loss

of braking force despite the system being fully pressurized.

When a truck experiences both low air pressure and brake fade simultaneously, the ability to decelerate safely is significantly compromised. This highlights the importance of proper brake management strategies and the integration of advanced braking technologies to prevent overheating and ensure consistent braking performance [15]. Given the growing emphasis on safety in heavy-duty transportation, modern truck braking systems continue to evolve, incorporating electronic braking controls, real-time temperature monitoring, and predictive maintenance features to prevent such failures.

2.4 Materials for brake discs

The brake disc is a mechanical component designed to reduce a vehicle's velocity or bring it to a complete stop by decelerating the motion of its wheels. Braking occurs when the brake pads clamp onto the rotating disc, applying a force that generates friction between the pads and the disc surface [1]. This friction slows down or halts the rotation of the wheel, thereby producing braking power. The effectiveness of this braking action largely depends on the coefficient of friction between the pad and disc surfaces; a higher coefficient results in greater braking power. This coefficient varies depending on the material composition of the disc.

Since the braking system is a fundamental part of any transportation system, the materials used in brake discs must possess several essential properties: high compressive strength, wear resistance, low density, good thermal conductivity and cost-effectiveness. Various materials have been studied and implemented, but the most commonly used include grey cast iron, titanium alloys, aluminum metal matrix composites (AMCs), steel and carbon-ceramic composites. Brief descriptions of these materials are shown in Table 2.1.

Material	Description
Grey Cast Iron	A metallic alloy containing 2–4.5% carbon. It is widely used due to its low cost, ease of manufacturing, and thermal stability. These characteristics make it one of the most popular materials for brake discs in the market.
Titanium Alloys	Titanium and its composites can reduce the weight of the disc by approximately 37% compared to cast iron of the same dimensions. Additionally, they offer high-temperature strength and superior corrosion resistance.
Aluminum Metal Matrix Composites (AMCs)	AMCs have lower density and higher thermal conductivity compared to conventional grey cast iron, potentially reducing the weight of brake systems by 50–60% [16]. However, they have three major drawbacks: 1. Segregation or inhomogeneity of SiC particles during solidification, due to density differences between aluminum and SiC; 2. Reduced ductility caused by the presence of SiC, which compromises product reliability; 3. Absence of graphite, leading to lower braking efficiency and increased adhesive wear.
Steel	While not particularly durable over long-term use, steel offers improved heat capacity and results in a relatively lighter brake system.
Carbon-Ceramic Composites	Produced by reinforcing silicon carbide with carbon filaments [1].The primary constituent is silicon carbide, which provides excellent thermal and mechanical load resistance. Despite their superior performance, the high production cost limits widespread use.

Table 2.1: Overview of materials used for brake discs and their key properties

Several material options are available in the market for brake discs; however, for this research, we focus on grey cast iron due to its favorable thermal and mechanical properties [1], as shown in Table 2.2. Cast irons are iron-carbon-silicon alloys, differing from steels by their higher carbon content. While steels typically contain between 0.008% and 2% carbon, cast irons have at least 2% carbon. Specifically, grey cast iron consists of approximately 2.5% to 4% carbon (C) and around 3% silicon (Si), with additional elements such as manganese influencing its microstructure.

The microstructure of grey cast iron varies depending on cooling rates and alloying elements [1]. Grey cast iron has graphite flakes, and the sizes ranges from 0.015 mm to 1 mm, embedded in a metallic matrix. While these graphite flakes enhance the material's thermal conductivity, they also contribute to mechanical weakness by acting as crack initiators, making the material more susceptible to fractures if subjected to tensile stress.

Grey cast iron is characterized by:

- Density: ρ [kg/m^3]
- Thermal conductivity: λ [W/mK]
- Specific heat capacity: c [J/kg/K]

Material	$\boldsymbol{\rho}$ [kg/m$^{\mathbf{3}}$]	$\boldsymbol{\lambda}$ [W/mK]	$\boldsymbol{c}$ [J/kg/K]
Grey cast iron	7250	54.0	500.0

Table 2.2: Material properties of grey cast iron [1].

2.5 Temperature Estimation Methods

The authors of [17] review several methodologies currently employed to estimate disc brake temperatures. One of the earliest approaches is the analytical method. A.A. Yevtushenko et al. [18] provide a detailed description of this technique in the context of railway vehicles, where temperature is modeled as a function of various parameters—such as initial conditions, material properties, and the geometric characteristics of the disc and pad—derived from fundamental physical laws and equations. While this method ensures physical consistency, it remains highly constrained by the specificity of the required inputs. In many scenarios, the spatial variability of physical quantities across the disc and pad surfaces, along with assumptions of constancy for some parameters, render analytical solutions either unattainable or significantly detached from real-world conditions.

Another widely adopted approach involves numerical simulation. As demonstrated by Zhuojun et al.[19], the integration of Computer Aided Design (CAD) models for both disc and pad, along with simulations of the surrounding airflow, enables a more comprehensive and accurate representation of the thermal behavior. This method not only enhances predictive accuracy but also provides deeper insight into the system's overall performance. However, it is also characterized by substantial computational demands and complexity in setup. Moreover, each distinct braking scenario necessitates a separate simulation, thereby increasing the time and resources required.

A third method discussed is experimental investigation. Panier et al. [20] conducted empirical studies on the development of hot spots in railway braking systems, using full-scale prototypes. Their findings underscore the high degree of accuracy and realism achievable through experimental methods. Nonetheless, these approaches are often cost-prohibitive due to the need for specialized equipment and large-scale test setups.

Finally, emerging techniques leveraging machine learning and artificial intelligence have been proposed as promising alternatives.

2.6 Machine Learning Overview

Machine learning (ML) is a subset of Artificial Intelligence (AI), a term coined for the first time by McCarthy and Minsky in 1956 [21]. In general, ML can be considered as a set of methods which can be used to solve a vast variety of real-world problems using computer systems. Machine learning focuses, in particular, on the development of algorithms which allows computer systems to learn from data and to make accurate predictions based on the given data [22].

2.6.1 Supervised Learning

In supervised learning, the machine is able to build knowledge about a specific task based on a series of examples, which represents the "past experience" [21]. In this group the model does not need manual adjustment or programming rules to solve the problem, in this case it is able to solve the problem itself. The model is generally trained on labeled data and uses this data to learn patterns and make predictions [22]. A clear and simple example is an email detector, where the system learns from a given dataset of emails, which will be labeled as spam or not, allowing the model to classify new unseen emails in these two categories.

2.6.2 Unsupervised Learning

In unsupervised learning, the data are not labeled. One of the main examples of this category is clustering [23]. For example, we have a set of measurements representing the length and dimension of petals of three different flowers. All we have is a set of measurements and we are asked to group the respective measurements for each flower. For this task we can use unsupervised learning techniques to be able to identify, automatically, three clusters from the given measurements. Another example where we use unsupervised learning is anomaly detection [22]. In this case the algorithm identifies outliers or unusual data in the dataset. This could be used for instance in fraud detection, where fraudulent activities happens or to monitor unusual transactions of money.

2.6.3 Reinforcement Learning

Reinforcement learning (RL) is a method which rewards or penalizes a model and it consists of teaching intelligent agents to take different actions to increase their reward [21]. In this case the system is continuously interacting and learning from the environment and gets feedback from it [23]. For example, one of the most famous applications of RL is robotics. In this specific case, the robot is able to learn how to navigate an environment, based on a series of trial and errors. The system will receive negative or positive feedback, such as rewards if it successfully completes the task or penalties if it makes mistakes. Over time, it will be able to learn an optimal solution to navigate the environment [22].

2.6.4 Deep Learning

Deep learning has gained significant attention due to its ability to automatically learn hierarchical representations of data through deep neural networks. As Goodfellow et al. [24] highlight, deep learning models, particularly convolutional neural networks (CNNs), have revolutionized fields such as computer vision, speech recognition and natural language processing. These models process data through multiple layers of interconnected nodes, with each layer capturing increasingly abstract features from the input, allowing them to achieve impressive accuracy in complex tasks.

Deep learning represents a prominent subfield of machine learning, characterized

by the use of artificial neural networks composed of multiple layers of interconnected computational units, commonly referred to as *neurons* [25]. These neurons are loosely inspired by their biological counterparts and are organized into layered structures that enable the modeling of complex, non-linear relationships within data.

Fundamentally, a neural network aims to approximate a function that maps a given set of inputs to corresponding outputs. Each neuron performs a mathematical transformation, typically a weighted sum followed by a non-linear activation, and passes the result to subsequent layers. The final output of the network emerges as a composition of these transformations, effectively forming a hierarchical representation of the input data.

Learning in a neural network involves adjusting the connection weights through an optimization process, most often via stochastic gradient descent or its variants. This process seeks to minimize a predefined loss function, which quantifies the difference between the predicted outputs and the ground truth labels. By iteratively refining the weights, the network learns to model the underlying data distribution.

The architecture of a neural network is highly adaptable and can be tailored to suit specific tasks. Variations may include increasing the number of layers (depth), expanding the number of neurons per layer (width) or modifying the connectivity patterns between neurons, see Figure 2.4. While deeper and wider networks generally offer greater capacity to learn complex functions, they also introduce challenges such as higher computational cost and a greater risk of overfitting. These trade-offs necessitate careful architectural design, regularization strategies, and empirical tuning to achieve optimal performance.

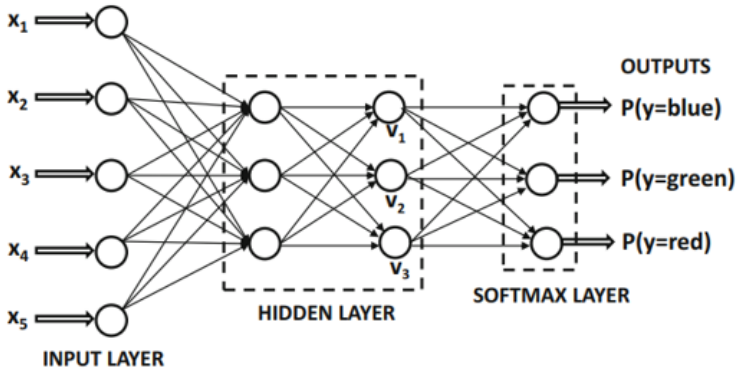


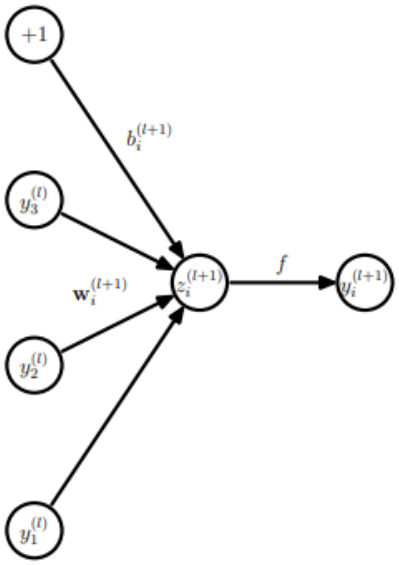
Figure 2.4: Fully Connected Neural Network [25].

2.6.5 Neural Networks

Neural networks are organized into layers of neurons, and the flow of information between layers defines the network's topology. The two principal types of neu-

ral architectures are *Feedforward Neural Networks* (FFNNs) and *Recurrent Neural Networks* (RNNs), which differ in the connectivity pattern among neurons. In feedforward networks, information flows unidirectionally from input to output without any feedback loops. In contrast, recurrent networks allow connections from output neurons back to inputs, enabling them to maintain internal state and model temporal dependencies.

Architecturally, neural networks can be further categorized based on their depth. A network with only one hidden layer is referred to as a shallow network, while those with two or more hidden layers are considered deep neural networks. As noted by Sazli [26], increasing depth or width enhances the network's capacity to model complex functions, but also introduces challenges such as vanishing gradients and overfitting.



(a) Standard network

Figure 2.5: Standard Network [27]

The forward propagation step for a neuron i in layer $l+1$, see Figure 2.5, is given by:

$$z_i^{(l+1)} = \mathbf{w}_i^{(l+1)} \cdot \mathbf{y}^l + b_i^{(l+1)},$$
$$y_i^{(l+1)} = f\left(z_i^{(l+1)}\right),$$

where $\mathbf{y}^l$ denotes the output vector from the previous layer l, $\mathbf{w}_i^{(l+1)}$ is the weight vector associated with neuron i in layer $l+1$, and $b_i^{(l+1)}$ is the corresponding bias term. The value $z_i^{(l+1)}$ represents the pre-activation value, while $y_i^{(l+1)}$ is the post-activation output. The function $f(\cdot)$ is a non-linear activation function, such as ReLU, sigmoid, or hyperbolic tangent. This forward pass is repeated layer by layer

until the network produces an output [27]. During training, the parameters $\mathbf{w}_i^{(l)}$ and $b_i^{(l)}$ are optimized through backpropagation and gradient-based methods to minimize a chosen loss function.

2.6.6 Cost Function in Deep Learning

In deep learning, the cost function, also referred to as the loss function, plays a central role in training models by providing a quantitative measure of error between the predicted outputs and the actual target values [24]. It acts as the objective function that the learning algorithm seeks to minimize throughout the training process. In a supervised learning setting, where the model is trained on a dataset of input-output pairs $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$, the prediction for a given input $\mathbf{x}^{(i)}$ is denoted by $\hat{\mathbf{y}}^{(i)} = f(\mathbf{x}^{(i)}; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents the set of model parameters. The cost function $\mathcal{J}(\boldsymbol{\theta})$ is typically defined as the average loss over all m training examples:

$$\mathcal{J}(\boldsymbol{\theta}) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\hat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)}), \tag{2.16}$$

where $\mathcal{L}$ denotes the individual loss function. The choice of $\mathcal{L}$ depends on the nature of the task. For regression tasks, the Mean Squared Error (MSE) is commonly used and is defined as:

$$\mathcal{L}(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2,$$

where $\hat{y}$ is the predicted value and y is the true target value. For classification tasks, the Cross-Entropy Loss is typically employed and is given by:

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_{k=1}^{K} y_k \log(\hat{y}_k),$$

where $\mathbf{y}$ is the true label vector, $\hat{\mathbf{y}}$ is the predicted probability distribution over K classes, and y_k and $\hat{y}_k$ are the true and predicted values for class k, respectively.

Minimizing the cost function with respect to the parameters $\boldsymbol{\theta}$ allows the model to iteratively improve its performance. This is typically achieved using optimization algorithms such as Stochastic Gradient Descent (SGD) or its variants (e.g., Adam, RMSprop). The gradients required for this optimization are computed via the backpropagation algorithm, which relies directly on the cost function to provide the necessary error signal. The cost function thus plays a fundamental role in guiding the learning process, influencing both convergence behavior and final model accuracy.

2.6.7 Activation Functions in Deep Learning

Activation functions are essential components of neural network architectures, enabling them to learn complex, non-linear patterns [24]. Without activation functions, even deep neural networks would reduce to a single linear transformation,

fundamentally limiting their expressive capacity. Formally, an activation function f is applied element-wise to the pre-activation value z of a neuron:

$$y = f(z), \tag{2.17}$$

where $z = \mathbf{w}^\top \mathbf{x} + b$, with $\mathbf{x} \in \mathbb{R}^n$ denoting the input vector, $\mathbf{w} \in \mathbb{R}^n$ the weight vector, and $b \in \mathbb{R}$ the bias term. Several activation functions are commonly used in practice, each with its own characteristics and trade-offs [28]. The sigmoid function is defined as:

$$f(z) = \frac{1}{1 + e^{-z}}.$$

It was historically popular in early neural networks, especially for binary classification tasks. However, it suffers from issues such as saturation and vanishing gradients during backpropagation. The hyperbolic tangent function, or tanh, is given by:

$$f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}.$$

This function outputs values in the range $(-1, 1)$, making it zero-centered. While it often performs better than the sigmoid function, it still suffers from the vanishing gradient problem. The rectified linear unit (ReLU) is widely used in deep networks and is defined as: follows [29]:

$$h^{(i)} = \max(\mathbf{w}^{(i)\top}\mathbf{x}, 0) = \begin{cases} \mathbf{w}^{(i)\top}\mathbf{x} & \text{if } \mathbf{w}^{(i)\top}\mathbf{x} > 0 \\ 0 & \text{otherwise.} \end{cases}$$

This activation function is computationally efficient, encourages sparse activation, and helps mitigate the vanishing gradient issue. To address the problem of inactive neurons in ReLU, the leaky ReLU variant was proposed by Andrew L. Maas et al. [29]. It is defined as:

$$h^{(i)} = \begin{cases} \mathbf{w}^{(i)\top}\mathbf{x} & \text{if } \mathbf{w}^{(i)\top}\mathbf{x} > 0 \\ 0.01\,\mathbf{w}^{(i)\top}\mathbf{x} & \text{otherwise.} \end{cases}$$

Leaky ReLU introduces a small, non-zero gradient when the unit is not active, thereby improving gradient flow and model robustness during training. Finally, the softmax function is typically applied in the output layer for multiclass classification problems. It transforms a vector of raw scores (logits) into a probability distribution over K classes [30]:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

This function ensures that each output is non-negative and that the outputs sum to one, making it particularly suitable for interpreting model predictions as class probabilities.

2.6.8 Backpropagation

Backpropagation is a key method for calculating gradients efficiently within artificial neural networks [24]. This process involves propagating information backward from the output layer through the preceding layers, allowing adjustments to be made to the model's parameters, such as weights and biases, based on the error observed. Fundamentally, backpropagation relies on the chain rule from calculus to determine the partial derivatives of a function composed of multiple nested functions. Given that neural networks consist of a series of interconnected layers, the overall error at the output can be broken down into derivatives corresponding to each layer's computations. By systematically propagating the error backward through the network, the algorithm calculates the gradients required to update parameters during training. Let $x \in \mathbb{R}$ be a real-valued input, and let f and g be differentiable functions such that:

$$y = g(x), \quad z = f(y) = f(g(x)).$$

Using the chain rule, the derivative of z with respect to x is:

$$\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx}.$$

In vector form, this can be generalized as:

$$\nabla_{\mathbf{x}} z = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right)^{\top} \nabla_{\mathbf{y}} z, \tag{2.18}$$

where $\mathbf{y} = g(\mathbf{x})$, and $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$ is the Jacobian matrix containing all partial derivatives of the output $\mathbf{y}$ with respect to the input $\mathbf{x}$.

The backpropagation algorithm iteratively applies this chain rule across each layer of the network. Starting from the output layer, it computes the gradient of the cost function with respect to each parameter by performing Jacobian-gradient products at every step of the computational graph. These gradients are then used by an optimization algorithm (e.g., stochastic gradient descent) to update the parameters and reduce the training error. This mechanism allows neural networks to learn complex functions by minimizing a loss function through successive updates, refining their predictions over time based on the gradients computed during backpropagation.

2.6.9 Gradient Descent and Optimization in Deep Learning

Optimization is at the heart of most deep learning algorithms. It involves adjusting the parameters of the model to reduce the value of the loss function defined earlier. This process is essential for enabling neural networks to learn patterns from data. Consider a differentiable function $y = f(x)$. The first derivative $f'(x)$ describes the slope of the function at a given point x, providing crucial information about how small changes in the input will affect the output. In the context of optimization, this gradient guides how to adjust x to reduce the value of $f(x)$. When the derivative equals zero, i.e., $f'(x) = 0$, no further improvement can be made through infinitesimal steps. These points are referred to as *critical points* or *stationary points*, as shown in Figure 2.6.

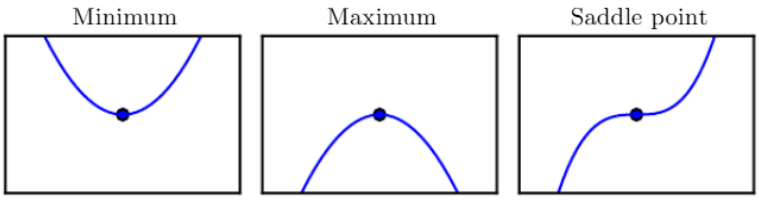


Figure 2.6: Critical points, Goodfellow et al. [24]

Critical points can take several forms. A local minimum is a point where $f(x)$ cannot be decreased further by small perturbations in the input. A local maximum is a point where $f(x)$ cannot be increased by small perturbations. A saddle point, on the other hand, is a critical point that is neither a minimum nor a maximum, but may exhibit characteristics of both depending on the direction of movement. In the ideal case, we aim to find the *global minimum*, which is the point at which $f(x)$ reaches its lowest possible value across the entire domain. However, the optimization landscape in deep learning is highly non-convex and often contains numerous local minima, saddle points, and flat regions, making exact global optimization intractable [24]. Therefore, practical algorithms settle for finding a parameter configuration that yields a sufficiently low cost, even if it is not the global minimum.

The most widely used optimization algorithm in deep learning is gradient descent, which iteratively updates the parameters in the direction opposite to the gradient of the cost function:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}_t), \tag{2.19}$$

where η is the learning rate, $\nabla_{\boldsymbol{\theta}} \mathcal{J}$ is the gradient of the cost function with respect to the parameters $\boldsymbol{\theta}$, and t denotes the iteration step. Modern variants of gradient descent improve its efficiency and convergence behavior. Building upon the basic concept of gradient-based optimization discussed earlier, one widely used advanced method is Adam, which stands for Adaptive Moment Estimation.

2.6.10 The Adam Optimizer

Adam is a first-order gradient-based optimization algorithm that combines the advantages of two other popular methods: momentum and RMSProp [31]. It has become one of the most widely used optimizers in deep learning due to its efficiency, robustness, and ability to handle sparse gradients and noisy objectives . Adam maintains two exponentially decaying moving averages for each parameter: one for the gradients (the first moment estimate) and one for the squared gradients (the second moment estimate). These moving averages are then bias-corrected to compensate for their initial values being set to zero.

- $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \eta = 10^{-8}$ (Defaults)

$m_0 \leftarrow 0$ (Initialize 1st moment vector)
$v_0 \leftarrow 0$ (Initialize 2nd moment vector)
$i \leftarrow 0$ (Initialize step)
while Θ_i not converged **do**
$i \leftarrow i+1$
$g_i \leftarrow \nabla_\Theta f_i(\Theta_{i-1})$ (Get gradients at step i)
$m_i \leftarrow \beta_1 \cdot m_{i-1} + (1-\beta_1) \cdot g_i$ (Update biased first moment estimate)
$v_i \leftarrow \beta_2 \cdot v_{i-1} + (1-\beta_2) \cdot g_i^2$ (Update biased second raw moment estimate)
$\hat{m}_i \leftarrow m_i / (1-\beta_1^i)$ (Compute bias-corrected first moment estimate)
$\hat{v}_i \leftarrow v_i / (1-\beta_2^i)$ (Compute bias-corrected second raw moment estimate)
$\Theta_i \leftarrow \Theta_{i-1} - \alpha \cdot \hat{m}_i / (\sqrt{\hat{v}_t} + \eta)$ (Update parameters)
end while
return Θ_i (resulting parameters)

Figure 2.7: Adam optimizer algorithm. [31]

The Adam optimization algorithm,see Figure 2.7 begins by initializing the first moment vector m_0, the second moment vector v_0, and the iteration counter i to zero. Here, m_t represents the exponentially decaying average of past gradients, also referred to as the first moment estimate, while v_t denotes the exponentially decaying average of past squared gradients, or the second moment estimate. At each iteration t, the gradient of the loss function with respect to the parameters is computed as

$$g_t = \nabla_\theta \mathcal{L}_t(\theta_{t-1}).$$

Subsequently, the biased first and second moment estimates are updated according to

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad \text{and} \quad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2,$$

where β_1 and β_2 are the exponential decay rates for the moment estimates. To correct for the bias introduced by initializing the moment vectors at zero, bias-corrected estimates are computed as

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \quad \text{and} \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t}.$$

Finally, the parameters are updated using the rule

$$\theta_t = \theta_{t-1} - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \eta},$$

where α is the learning rate and η is a small constant added to ensure numerical stability.

The parameter α represents the learning rate, a small scalar that determines the step size at each iteration while minimizing the loss function; a typical default value is 0.001. The parameter β_1 is the exponential decay rate for the first moment estimates (i.e., the moving average of gradients), commonly set to 0.9. Similarly, β_2 is the exponential decay rate for the second moment estimates (the moving average of

squared gradients), usually set to 0.999. The constant η is a small number, typically 10^{-8}, added to the denominator for numerical stability and to avoid division by zero. The variable m_t denotes the biased first moment estimate (the mean of gradients), while v_t represents the biased second raw moment estimate (the uncentered variance of gradients). The terms $\hat{m}_t$ and $\hat{v}_t$ refer to the bias-corrected first and second moment estimates, respectively. Finally, θ_t indicates the parameters (weights) being optimized during training.
Adam dynamically adjusts the learning rate for each parameter, making it well-suited for problems with non-stationary objectives and varying feature scales. Its fast convergence and relatively low sensitivity to hyperparameters make it a common default in deep learning frameworks.

2.6.11 AdamW Optimizer

AdamW is a variant of the Adam optimizer that introduces a more effective approach to applying weight decay during model training. While it builds on the adaptive learning features of Adam, AdamW separates the regularization process from the gradient-based update, addressing a known limitation in the original Adam formulation [32].

In standard Adam, weight decay (often used for regularization) is applied indirectly by adding an L2 penalty term to the gradient during the update step. However, this approach can cause interactions with Adam's adaptive learning rates, making the regularization effect inconsistent and potentially less effective.

AdamW modifies this behavior by decoupling the weight decay from the gradient update. Instead of modifying the gradient itself, AdamW directly subtracts a small portion of the model weights before performing the parameter update. This more principled approach ensures that regularization acts purely to constrain the magnitude of the weights, without interfering with the adaptive gradient scaling.

Due to this adjustment, AdamW often leads to improved generalization and more stable training, especially in deep learning models where overfitting is a concern. In this work, AdamW was used to train one of the GRU-based models to evaluate its impact on model performance and convergence behavior in comparison to the standard Adam optimizer.

2.6.12 Recurrent Neural Networks (RNNs)

Many machine learning tasks, such as language modelling, speech recognition, and time-series forecasting, involve sequential data where the order of inputs significantly affects the output [33]. Traditional feedforward neural networks are not well-suited for such problems, as they assume independence between inputs. Recurrent Neural Networks (RNNs) address this limitation by incorporating feedback connections, enabling them to maintain a form of memory over time. In a RNN, the hidden state at each time step t is a function of the input at that time x_t and the hidden state

from the previous time step h_{t-1}:

$$h_t = \phi(W_h h_{t-1} + W_x x_t + b) \tag{2.20}$$

where W_h and W_x are weight matrices, b is a bias vector, and ϕ is a non-linear activation function (typically tanh or ReLU). The output y_t at each step is computed as:

$$y_t = \psi(W_y h_t + b_y) \tag{2.21}$$

with ψ often being the softmax function for classification tasks. This process is computed iteratively:

$$h_0 = \mathbf{0} \tag{2.22}$$

$$\text{for } t = 1 \text{ to } T: \tag{2.23}$$

$$h_t = \phi(W_h h_{t-1} + W_x x_t + b) \tag{2.24}$$

$$y_t = \psi(W_y h_t + b_y) \tag{2.25}$$

While RNNs are theoretically capable of learning long-term dependencies, in practice they suffer from issues such as vanishing and exploding gradients, making it difficult to propagate information across many time steps. This limitation led to the development of more advanced architectures like the Long Short-Term Memory (LSTM) network.

2.6.13 Long Short-Term Memory (LSTM)

LSTM networks, introduced to overcome the limitations of standard RNNs, incorporate a memory cell that can preserve information across long sequences. Each LSTM cell is equipped with gates that regulate the flow of information. The forget gate, denoted as f_t, determines which portion of the previous cell state should be discarded. The input gate, i_t, decides what new information should be stored in the cell state. Finally, the output gate, o_t, controls the output based on the current cell state, as shown in Figure 2.8

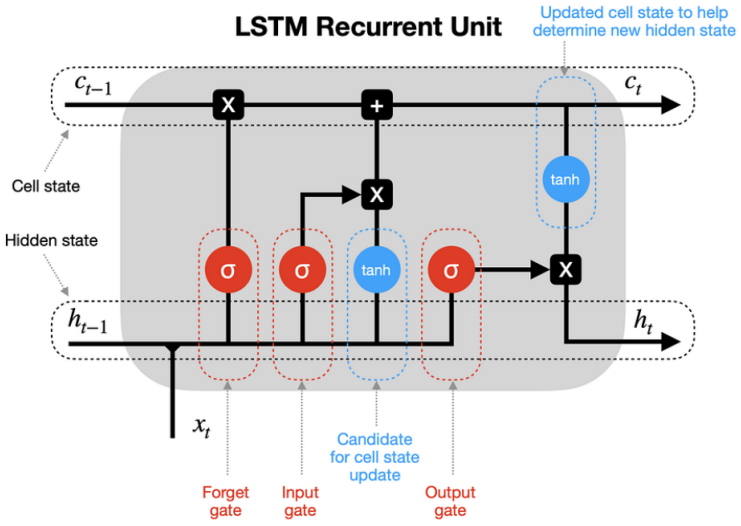


Figure 2.8: The LSTM architecture [33]

The equations governing the forward pass of an LSTM cell at time step t are as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.26}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.27}$$
$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{2.28}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{2.29}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.30}$$
$$h_t = o_t \odot \tanh(c_t) \tag{2.31}$$

The computation within an LSTM cell at time step t involves several components. First, the forget gate f_t determines which parts of the previous cell state c_{t-1} should be discarded. It is computed as $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$, where σ is the sigmoid function, which outputs values between 0 and 1, W_f is the weight matrix for the forget gate, h_{t-1} is the previous hidden state, x_t is the current input, and b_f is the bias term.

Next, the input gate i_t decides which new information should be added to the cell state, given by $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$. Simultaneously, a candidate cell state $\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$ is computed using a hyperbolic tangent activation function, capturing the potential new memory to be added. The new cell state c_t is then updated by combining the scaled previous cell state and the new candidate, as $c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$, where $\odot$ denotes element-wise multiplication.

The output gate o_t determines which parts of the cell state should influence the next hidden state. It is computed as $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$. Finally, the hidden state h_t, which also serves as the output of the LSTM cell, is obtained by applying the output gate to the cell state's non-linear transformation: $h_t = o_t \odot \tanh(c_t)$. Each of these gates and transformations allows the LSTM to effectively control the flow of information, enabling the model to retain or forget information as needed over long sequences.

The forget gate f_t determines the extent to which the previous memory c_{t-1} is retained. Simultaneously, the input gate i_t and the candidate cell state $\tilde{c}_t$ collaborate to decide the amount of new information to be incorporated. The updated cell state c_t is formed by integrating the retained memory with the new input. Finally, the output gate o_t selects which components of the cell state are used to generate the current output h_t. This gating mechanism allows LSTM networks to maintain long-term dependencies and selectively forget or update information, making them highly effective for sequence modeling tasks such as language modeling, speech recognition and time series prediction.

2.6.14 Physics-Informed Neural Networks (PINN)

A PINN is a particular kind of neural network that learns not only from data but also from the physical laws that describe a system itself. These physical laws are

often written as partial differential equations (PDEs), such as equations for heat transfer, motion or fluid flow [34].

In a regular neural network, the model is trained only to match observed data. In a PINN, the training process includes three blocks. A neural network takes a vector of variables from the equation as input and produces a corresponding output value. PINNs extend this by computing derivatives to evaluate the losses associated with the terms of the governing equations. A feedback mechanism is then used to minimize these losses according to specified learning rates, as shown in Figure 2.9.

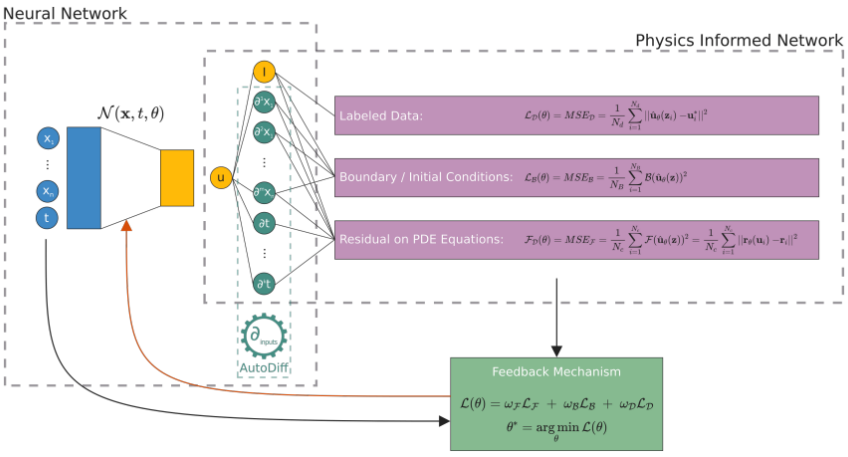


Figure 2.9: PINN building blocks [34]

This is done by adding a *physics loss* to the regular data loss. The total loss function looks like this:

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \lambda \mathcal{L}_{\text{physics}}, \tag{2.32}$$

where $\mathcal{L}_{\text{data}}$ represents the error between the network's prediction and the observed data, $\mathcal{L}_{\text{physics}}$ denotes the error resulting from substituting the prediction into the physical equation, and λ is a weighting factor used to balance the two terms. This way, the network is guided to produce results that not only fit the data but also follow the laws of physics. This is especially useful when we don't have much data, but we do know how the system behaves physically. In the context of disc brake temperature prediction, the governing physical model can be described by a simplified heat balance equation:

$$\frac{dT}{dt} = \alpha P - \beta(T - T_{\text{ambient}}), \tag{2.33}$$

where T is the brake temperature, P is the braking power, T_{ambient} is the ambient temperature, and α, β are heat transfer coefficients capturing heating and cooling dynamics, respectively. To integrate this equation into the PINN framework, the physics-based loss is computed as the mean squared residual of the discretized heat equation.

2.6.15 Gated Recurrent Unit

The Gated Recurrent Unit (GRU) is a type of RNN, introduced as a computationally efficient alternative to the LSTM network [35]. GRUs are particularly well-suited for time-series prediction tasks, such as forecasting brake disc temperature evolution based on sequential sensor input, due to their ability to model temporal dependencies while mitigating the vanishing gradient problem common in traditional RNNs.

GRUs use gating mechanisms to control the flow of information through the network [36]. Unlike LSTMs, which have separate input, output and forget gates, GRUs combine these into two primary gates: the update gate and the reset gate. The update gate determines how much of the previous hidden state should be retained, while the reset gate controls how much of the past information should be forgotten when computing the new candidate hidden state. This structure enables the GRU to adaptively capture long- and short-term dependencies without explicitly maintaining a memory cell, thereby reducing model complexity and training time.

In the context of brake temperature modeling, GRUs provide a balance between predictive performance and computational efficiency. Their relatively simple structure compared to LSTM allows for faster training and inference, which is advantageous when deploying models in resource-constrained environments or for real-time applications in vehicles.

2.6.16 Random Forest

Random Forest is a widely used ensemble learning technique that builds on the foundation of decision trees by combining the predictions of multiple trees to improve generalization and reduce overfitting [37]. It is particularly useful in tasks involving complex datasets with noisy or heterogeneous input features, such as predicting the temperature of brake discs using time-varying sensor data.

The method operates by training each tree on a randomly sampled subset of the training data (with replacement) and selecting a random subset of features at each decision point [38]. This dual-randomization process introduces variability among the individual trees, which helps to lower the model's variance without substantially increasing bias. Final predictions are obtained by averaging outputs in regression tasks or taking a majority vote in classification scenarios.

Applied to the problem of brake temperature forecasting, Random Forests offer several practical benefits. They are capable of capturing nonlinear interactions between input variables and are inherently resistant to overfitting, especially when compared to single decision trees. Additionally, their ability to estimate feature importance can provide valuable insights into which sensor readings having largest influence on temperature behavior. These qualities make Random Forest a reliable and interpretable model choice, particularly suitable for environments requiring dependable performance with relatively low computational cost.

2.6.17 One-Dimensional Convolutional Neural Networks (1D CNNs)

1D CNNs are a class of deep learning models designed for analyzing sequential data, where inputs vary over a single dimension such as time or position. These models are particularly effective at learning spatially or temporally local patterns, which makes them well-suited for time-series problems like predicting brake disc temperature based on sensor readings [39].

The core mechanism involves sliding convolutional filters over the input sequence to extract features at various levels of abstraction. Each convolution operation is followed by nonlinear transformations, and often pooling layers that reduce dimensionality while preserving critical information [40]. Because the filters are shared across the input, the model becomes efficient in both parameter count and computation, allowing for scalable learning on relatively long sequences.

In practical applications such as brake temperature modeling, 1D CNNs provide a strong compromise between accuracy and speed. Unlike traditional feature-based approaches, they can automatically identify meaningful patterns in raw input data, minimizing the need for manual preprocessing. Additionally, compared to recurrent models, 1D CNNs generally require less training time and are better suited for systems with limited computational resources, such as embedded platforms in automotive environments.

2.6.18 Machine Learning Models to Predict Heating During braking

The thermal behavior of braking systems is inherently complex, influenced by various factors such as conduction, convection, radiation, cooling conditions, and material properties. Accurately predicting the heat generated during braking is critical in both the automotive and railway sectors. Traditionally, numerical methods such as Finite Element (FE) modeling have been employed to simulate temperature evolution in brake components. While these methods provide detailed and reliable insights, they are often computationally intensive and require comprehensive physical modeling, which limits their practical application in real-time applications. In contrast, machine learning techniques offer a data-driven alternative for modeling complex thermal behaviors in a more computationally efficient way. Despite their growing popularity, the application of ML to predict thermal behavior in braking systems remains relatively underexplored in the literature.

One notable contribution is by Pavelčík et al.[17], who investigated different classical ML algorithms, including linear regression, Support Vector Machines (SVM) and Gaussian Process Regression (GPR), to predict railway brake disc temperatures. Using MATLAB R2020a´s Machine Learning Toolbox and its Regression Learner app, they trained models on a limited dataset comprising only 20 samples (15 for training and 5 for testing) from a single disc-pad configuration. Their results

demonstrated that nearly 99% of the variance in brake disc temperature could be explained by two primary features: vehicle velocity and braking force. Despite the limited dataset, this study illustrates the potential of even simple ML models in capturing key thermal dynamics.

Artificial Neural Networks (ANNs) have been applied to thermal prediction tasks. Ghadimi et al. [41] developed an ANN model to estimate the heat flux in locomotive brake discs based on measured temperature data. Their model was capable of providing real time estimations of the heat flux, thereby offering valuable insights into the thermal state of the braking system during operation.
More recently, PINNs have been proposed to combine data driven modeling with the governing physical laws of heat transfer. Unlike traditional ML models, PINNs incorporate physical constraints, typically in the form of PDE, into the training process by augmenting the loss function with physics based residuals. Cai et al. [42] applied PINNs to a variety of heat transfer problem, including scenarios with incomplete boundary conditions, and demonstrated that the model could accurately predict both temperature and velocity fields even in data scarce environments. The key innovation lies in the use of automatic differentiation to enforce compliance with the underlying physics across the entire domain, thus ensuring physically consistent predictions without the need for explicit numerical solvers like FE.

In addition to feedforward networks and PINNs, RNNs have been explored for time series modeling in braking systems. Grochevaia et al. [43] investigated the use of encoder-decoder architectures based on LSTM and GRU cells to predict brake squeal behavior, a complex vibrational phenomenon closely tied to braking dynamics. Using time series data of the normal force between the brake pad and disc, the LSTM model achieved a weighted Mean Absolute Percentage error (wMAPE) of 61.57% and a Mean Absolute Error (MAE) of 0.1647 N. The GRU model performed better in terms of wMAPE (21.77%) but slightly worse in MAE (0.1804 N). While this study study did not focus on thermal prediction, it underscores the potential of deep learning methods, particularly recurrent architectures, in modeling complex, time dependent phenomena in braking systems.

Overall, while current literature demonstrates promising results from classical ML models and early deep learning implementations, there is a significant potential for further research. In particular, the integration of physics based knowledge into neural architecture (e.g., PINNs) and the use of advanced recurrent models (e.g.,GRU or LSTM) could substantially improve the robustness, interpretability, and accuracy of temperature predictions in real world braking systems.

3
Method

This chapter outlines the method used to develop machine learning models for predicting brake disc surface temperature using time-series data from controlled dynamometer tests. A preprocessing pipeline was established and followed by all implementations in the project, designed to follow industry standards for handling time-series sensor data without introducing data leakage. The workflow includes dataset assembly, feature engineering, preprocessing, and model training, followed by division into two independent modeling strategies. The models were developed in Python using pandas [44], NumPy [45], scikit-learn [46] and the PyTorch framework [47]. Although beyond the formal scope of this thesis, a final evaluation on real-world field test data was performed to assess the generalizability of the best-performing model.

3.1 Data Introduction

The dataset used in this project was collected from 17 structured braking sections performed in a dynamometer laboratory. Each section simulates a distinct braking scenario defined by specific configurations of input variables. These configurations include predetermined values for features such as initial vehicle speed, brake torque, braking duration and initial disc inboard temperature (e.g., 100°C, 200°C, 300°C). While this setup is useful for controlled brake testing, it introduces a bias into the dataset, as many braking events begin from the same set of initial conditions. This lack of randomness, particularly in initial temperatures and initial speeds results in clustered data distributions and reduces the overall variability of the input space. This could limit the model's generalizability to real-world conditions where initial states are more diverse. Additionally, as mentioned in the limitations, the same brake disc and pad setup was used across all tests to ensure data consistency.

The complete dataset was consolidated into a single file with 9767 samples corresponding to 168 braking stops. After segmentation and filtering, an initial split of 134 stops for training and 34 for testing was used for the first model (Test 1). For subsequent models, a validation set was included: 80% of the stops (134) for training, 10% (17) for validation and 10% (17) for testing. The sampling frequency was 10 Hz, and the average stop duration was 5.81 seconds. The features provided by the dyno lab sensors are shown in Table 3.1.

Table 3.1: Input features provided by dyno lab sensors

Feature	Unit	Description
Torque	Nm	Braking torque applied to the disc
Pressure	kPa	Pneumatic pressure in the braking system
Speed	km/h	Vehicle speed at the time of braking
Disc Inboard/Outboard Temp	°C	Surface temperature of the brake disc (both sides)
Pad Inboard/Outboard Temp	°C	Surface temperature of the brake pad (both sides)
Stop Number	—	Identifier for each braking event
Time	s	Timestamp of each sample (10 Hz resolution)

To avoid data leakage and ensure robust evaluation, a base pipeline was carefully constructed which acted as the base for the developed models. All models were developed with this method, in the following order:

- **Cleaning:** Data which was originally in different sections, was put in order inside one excel sheet. Any rows containing NaN values were dropped, and the index was reset to preserve consistency.
- **Stop Grouping:** The data was grouped by stop using a combination of the `Section` and `Stop_No` columns. Each stop was treated as an independent braking event throughout preprocessing, windowing, and modeling stages.
- **Train-Test Split:** The original CSV file was manually divided into train set and test set by randomly assigning entire stops to either set. This ensured no braking stop appeared in both training and testing.
- **Feature Engineering:**
  - Cumulative Torque was engineered as the cumulative sum of torque within each braking stop, resetting at each new stop. It provided the model with an estimate of accumulated braking energy, which correlates with total heat generation. Since each prediction window was limited to 4 seconds or less, the cumulative torque feature gave additional context that extended beyond the window length.
  - Braking Indicator was added as a binary feature to indicate whether active braking was occurring. The flag was set to 1 when torque exceeded 100 Nm, otherwise set to 0.
- **Windowing:** A sliding window was applied within each stop. All models were evaluated using a fixed window size. The window length varied between the different models for investigative purposes.
- **Scaling:** All input features were standardized using StandardScaler. The scaler was fitted on the training data only, and the same transformation (using the training set's mean and standard deviation) was applied to the test data to avoid data leakage.
- **Exporting:** The processed training and test sets, along with all fitted scalers, were saved using Python's pickle module as `.pkl` files. Additionally, the trained models were saved after training to allow testing on unseen test data.

Epochs, Learning Rate Scheduling, Patience and Early Stopping

An epoch refers to one complete pass through the entire training dataset during the training process. Throughout training, the model iteratively updates its weights at the end of each epoch based on the computed error. Typically, multiple epochs are required for the model to progressively learn and generalize from the data.

In this work, we employ learning rate scheduling and early stopping, both of which are closely tied to the concept of epochs. Specifically, we use a learning rate annealing strategy, where the learning rate is gradually reduced as training progresses. This approach allows the model to make larger weight updates in the early stages of training and smaller, more precise updates as it approaches convergence. Additionally, we apply early stopping with a defined patience parameter, which halts training if the validation loss does not improve for a set number of consecutive epochs. This helps prevent overfitting and avoids unnecessary computation once learning stagnates.

3.2 GRU model: 2-stage

The pipeline for this model was extended with certain engineering to improve the model's understanding of braking history and dynamics. It's a 2-stage GRU model where the results from the first stage help give context to make a more accurate prediction on the second stage. Stage 1 predicts the temperature 1 and 2 seconds ago (t-10 and t-20)[1], whereas stage 2 predicts the current timestep t=0 in the window. Stage 2 receives the same input window as Stage 1, with the addition of the two temperature estimates (t-10 and t-20) predicted by Stage 1. Each stage was implemented and trained as a separate PyTorch model to aid with modular evaluation and debugging. The window size was set to 40 timesteps (4 seconds). Instead of targeting both disc inboard and outboard temperatures like the other models presented in this thesis, this model targets disc inboard solely, without specific justification.

- **Features t-20 and t-10:** The estimated temperature 1 and 2 seconds ago, used as input for stage 2 of the model. The purpose of these was to introduce a trendline of the heat development in the brake disc to aid the model with its prediction

- **Delta Features:** First-order deltas (i.e., change between consecutive time steps) and 1-second deltas (i.e., change over a 1-second interval) were added for torque, pressure and speed. These features help capture both fast and medium-term trends that affect thermal behavior.

[1]Time in this thesis is based on sample indices with a sampling frequency of 10 Hz, meaning 10 samples per second.

- **GRU Architecture:** The model architecture included three stacked GRU layers with hidden sizes of 256, 128 and 64, respectively. A dropout rate of 0.3 was applied between layers to prevent overfitting. Training used the Adam optimizer with a learning rate of 0.001. The batch size was 32, and early stopping was triggered if validation loss did not improve for 20 epochs.

- **Permutation Feature Importance:** This evaluation technique was used for this model to showcase importance of each feature. it works by measuring how much the model's error increases when a feature is randomly shuffled.

3.3 GRU Model with shorter window size

In contrast to the two-stage GRU architecture used in the previous approach, this model utilizes a shorter window size and a single-stage GRU enhanced with an attention mechanism. A GRU with attention is a GRU model enhanced with an attention layer that helps the model focus on the most relevant time steps in the input sequence when making a prediction. Instead of treating all past inputs equally, it learns to weigh them based on their importance. The window size was set to 10 time-steps (1 second).

- **Use of Past Data and Autoregressive Lag Features:** To better capture how brake disc temperatures evolve over time, the model uses an autoregressive approach by including lagged values of the target variables (`Disc_inboard` and `Disc_outboard`) at time steps $t-1$, $t-2$, and $t-3$ as additional input features. These lag features are taken strictly from within the 10-step input window, ensuring that only past data is used at each prediction step. This design allows the model to learn temporal dependencies while fully preventing data leakage, since no future information (beyond the current prediction time) is ever used during training or inference. This is a widely used and effective technique in time series prediction tasks, particularly when modeling systems with strong temporal dynamics.

- **Feature Engineering:** Several domain-informed features were engineered to enrich the temporal context:
  - Rolling statistics: 5-step rolling mean and standard deviations for `torque`, `pressure` and `speed` to capture local temporal patterns.
  - Interaction terms: features such as `pressure × speed` were included to model nonlinear effects.
  - Derived physical quantities: metrics like `brake power` and `angular velocity` were computed from base variables to reflect physical braking dynamics.
  - Temporal context: features indicating the position of each step within the braking event (e.g., relative time, cumulative time) were added to provide temporal structure.

- **GRU with Attention Architecture:** The model consists of two stacked GRU layers (hidden size = 128, dropout = 0.3), followed by an attention layer

that computes a context vector from the hidden states. This vector is passed to a feedforward network to produce the final temperature prediction.

- **Training Configuration:** The model was trained using the AdamW optimizer (learning rate = 1×10^{-3}, weight decay = 1×10^{-4}), with a batch size of 128 and cosine annealing learning rate scheduling. Early stopping mechanism,as shown in Algorithm 1, was applied with a patience of 20 epochs.

3.4 Baseline Model: Random Forest Regressor

To establish a traditional machine learning baseline for disc brake temperature prediction, a Random Forest Regressor was employed. This model served as a point of comparison against the deep learning-based GRU model. The employed hyperparameters are given in Table 3.2.

Table 3.2: Hyperparameters of the Random Forest Regressor

Hyperparameter	Value
`n_estimators`	100
`max_depth`	10
`random_state`	42

The model was trained using the `RandomForestRegressor` implementation from `scikit-learn`. Input features included both raw signals (e.g., torque, pressure, and speed) and engineered features such as torque-speed interaction, brake power, and acceleration. These features were extracted from the last time step of each window in the dataset to form a fixed-size input vector, appropriate for use with tree-based models.

All features were scaled using a `RobustScaler`, and target temperatures (inboard and outboard disc temperatures) were also normalized using a pre-fitted scaler before training. After fitting, predictions were inverse-transformed to real temperature values in degrees Celsius for evaluation. The model was assessed using the same evaluation metrics as the other models and was later saved for use in test set evaluation and comparison with the GRU-based architecture.

3.5 LSTM Baseline Model as a Comparative Benchmark

To rigorously evaluate the effectiveness of the proposed GRU-based attention model, we implemented an LSTM baseline model with a comparable architecture and training configuration. Both models use two stacked recurrent layers with hidden size 128 and dropout of 0.3 applied between layers to mitigate overfitting. Similarly, they are trained using the AdamW optimizer (learning rate 1×10^{-3}, weight decay 1×10^{-4}),

batch size 128, cosine annealing learning rate scheduling, and early stopping with a patience of 20 epochs.

The LSTM baseline processes the input sequence through its recurrent layers and relies solely on the final hidden state of the last time step to summarize the temporal information before passing it to a fully connected feedforward network that generates the prediction. This approach benefits from a simpler architecture and reduced computational complexity, while leveraging LSTM's proven ability to capture long-term dependencies in sequential data.

In contrast, the GRU model incorporates an attention mechanism on top of its two stacked GRU layers, allowing it to dynamically assign weights to hidden states at *all* time steps. This enables the model to selectively emphasize the most relevant moments in the sequence, capturing important temporal patterns that might be diluted or lost when relying only on the last hidden state.
The attention mechanism adds several advantages over the LSTM baseline:

- It enhances the model's capacity to represent complex and distributed temporal dependencies by learning a weighted context vector from the entire sequence.
- It improves interpretability by revealing which time steps contribute most significantly to the prediction.
- It may yield better generalization, especially on sequences with variable temporal dynamics, since it avoids compressing all information into a single fixed-length vector.

Using the LSTM baseline as a comparative benchmark provides a reliable point of reference, given LSTM's established success in sequential modeling tasks. Comparing against this baseline isolates the impact of the attention mechanism and demonstrates whether dynamically weighting time steps yields meaningful performance gains beyond standard recurrent architectures. This comparison also highlights trade-offs between model complexity, computational cost, and predictive accuracy in the context of braking temperature forecasting.

3.6 CNN Baseline Model as a Comparative Benchmark

To assess the effectiveness of the GRU-based attention model, we implemented a one-dimensional convolutional neural network (CNN) as a baseline architecture for comparative purposes. The CNN model was designed to process sequential time-series data by applying temporal convolutional filters over the input features. The model consists of two convolutional layers. The first layer uses 64 filters with a kernel size of 3 and padding of 1, followed by batch normalization and a ReLU activation function. A dropout layer with a probability of 0.3 is added to improve generalization and reduce overfitting. The second convolutional layer increases the feature dimension to 128 channels, again using a kernel size of 3 and padding of 1,

followed by batch normalization and ReLU activation.

After feature extraction through convolution, the model employs global average pooling to aggregate information across the temporal dimension, producing a fixed-length feature representation regardless of input sequence length. This vector is then passed through a fully connected feedforward network consisting of a 64-unit ReLU-activated hidden layer and an output layer with two neurons corresponding to the predicted disc temperatures.

The model was trained using the same optimization settings as the GRU model, including the AdamW optimizer, cosine annealing learning rate scheduler, early stopping with a patience of 20 epochs, and mean squared error as the loss function. By using this consistent training setup, we ensure a fair comparison between the CNN and GRU-based attention models.

3.7 Model Evaluation

To evaluate the performance and generalization capability of the AI virtual sensor models for predicting disc brake temperature, three metric commonly used in neural network regression metrics were employed: Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Coefficient of Determination (R^2). These metrics provide a comprehensive assessment of how closely the model's predictions align with the ground truth.

MAE measures the average absolute difference between predicted and actual temperature values. It provides a direct interpretation of the average error magnitude, without considering its direction:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{T}_i - T_i\right| \tag{3.1}$$

- $\hat{T}_i$ is the predicted temperature,
- T_i is the true temperature,
- N is the total number of samples.

RMSE provides a quadratic scoring of the prediction error, penalizing larger deviations more than MAE:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{T}_i - T_i\right)^2} \tag{3.2}$$

This is particularly useful in safety-critical applications such as thermal management in braking systems.

The R^2 metric evaluates the proportion of variance in the actual temperature values that is captured by the model. It is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left(\hat{T}_i - T_i\right)^2}{\sum_{i=1}^{N} \left(T_i - \bar{T}\right)^2} \tag{3.3}$$

where $\bar{T}$ is the mean of the actual temperature values. An R^2 value of 1 indicates perfect prediction, while 0 means the model performs no better than predicting the mean of the target variable.

4
Results and Discussion

This section presents the performance evaluation of the developed models. Metrics such as MAE, RMSE and R^2 are used to quantify model performance. The results are presented in plots and are aimed to illustrate prediction trends and residual patterns. Each test corresponds to a specific modelling approach or configuration, enabling direct comparison across architectures.

Test 1: GRU Model: 2-Stage

In the first experiment, the GRU model was developed with a 2 stage architecture, with delta-deatures and with a longer temporal context (window size of 40). It was trained with hyperparameters shown in Table 4.1. For this test, the only output parameter that we were aiming to predict was the disc inboard temperature.

Table 4.1: Model Hyperparameters for Stage 1 and Stage 2 of the GRU model

Hyperparameter	**Value**
Learning Rate	1×10^{-3}
Max Epochs	150
Patience (Early Stopping)	40
Window Size	40 (4 seconds)
Hidden Sizes (GRU Layers)	[256, 128, 64]
Dropout Rate	0.3
Loss Function	MSE
Optimizer	Adam

The results show that the GRU model is capable of accurately predicting the disc surface temperature across a wide range of operational conditions, achieving a low MAE of 27.15 °C and a high R^2 of 0.9554, see table 4.2. This indicates a strong correlation between predicted and actual values. The relatively low RMSE further confirms that large prediction errors are rare.

The scatter plot in Figure 4.1 demonstrates that the model generalizes well, with the majority of predictions lying close to the identity line. However, some vertical deviations can be observed, where the model both underestimates and overestimates the true values. These vertically clustered artefacts are probably a direct result of how the dynamometer tests were configured. Because the input variables were also created in clusters, with initial disc inboard temperatures set to 100°C, 200°C and

300°C etc. This causes the true target temperatures to cluster around specific values, leading to distinct vertical bands in the plot. The error spread within each band reflects the model's sensitivity to varying input signals.

The permutation feature importance test showed that `pred_t-10` and `pred_t-20` were by far the most important features, consistently dominating across all runs, as shown in Figure 4.2. Here they cause an increased MAE of around 51°C and 34°C respectively when shuffled. Other inputs such as `Pressure`, `Torque`, Δ`Torque` and `Cumulative_Torque` varied in importance between runs and had noticeably less impact. This confirms that the model relies heavily on recent thermal context to make accurate predictions. Overall, the 2-stage GRU model effectively captures the temporal dependencies in the braking process, particularly when supplied with context information from previous temperature states ($t - 10$ and $t - 20$)

Table 4.2: Evaluation Metrics for GRU Model on Test Set

Metric	Value
MAE	27.15 °C
RMSE	33.12 °C
R^2	0.9554

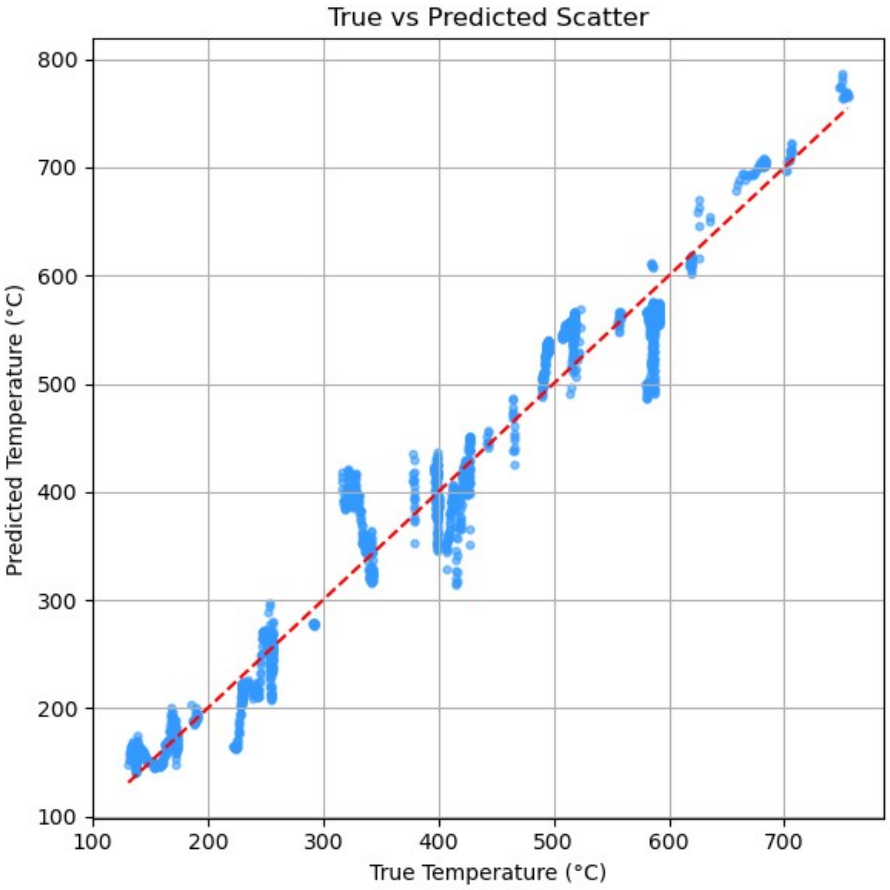


Figure 4.1: True vs predicted temperature for the 2-stage GRU model.

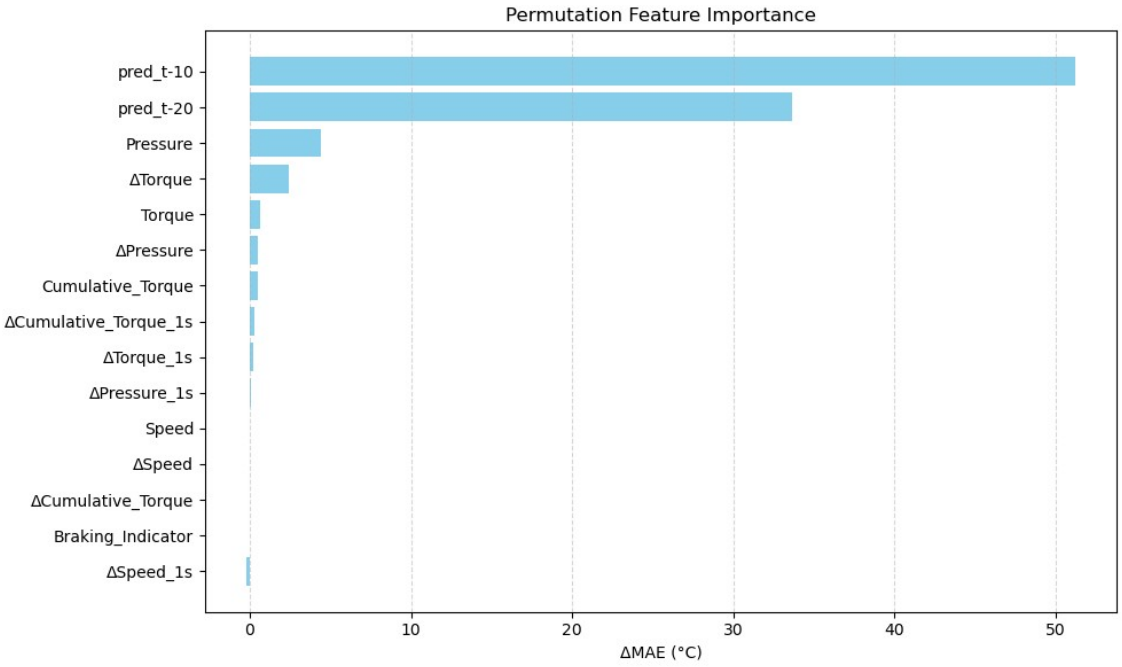


Figure 4.2: Permutation feature importance for the 2-stage GRU model.

Test 2: GRU - Attention Mechanism and Window Size = 10

Based on the hyperparameter in table 4.3, during training and validation, the loss curves show that the training loss converges closely with the validation loss. This is considered desirable, as it suggests that the model not only fits the training data effectively but also generalizes well to new, unseen data. The loss convergence behavior is illustrated in the loss curve plot in Figure 4.3.

Table 4.3: GRU Model Hyperparameters for Test 2

Hyperparameter	**Value**
Window Size	10
Hidden Dimension	256
Output Dimension	2
Batch Size	64
Epochs	200
Patience	25
Learning Rate	1×10^{-3}
Weight Decay	1×10^{-4}
Number of GRU Layers	2
Dropout	0.1

The GRU-based model, using the specified set of hyperparameters, demonstrated promising performance in predicting disc brake temperatures. As we can see from the evaluation metrics in table 4.4, the R^2 scores, exceeding 0.69 for both the inboard and outboard discs, indicate that the model successfully captures approximately 70%

of the variance in the validation dataset. This suggests that the model effectively learns the general thermal dynamics of the braking system.

Evaluation on Validation Set:

Table 4.4: Evaluation Metrics for GRU Model on Validation Set

Disc Type	**RMSE** (°C)	**MAE** (°C)	$\mathbf{R^2}$ Score
Disc Inboard	871.93	23.05	0.708
Disc Outboard	991.88	24.88	0.691

However, generalization to unseen validation data is somewhat challenged, likely due to mild overfitting. This is observed in the training process, where the validation loss begins to fluctuate and increase beyond epoch 20, see Figure 4.3, even though regularization techniques such as dropout and weight decay were applied. These fluctuations suggest that the model may have begun to memorize patterns in the training data rather than learning generalized features.

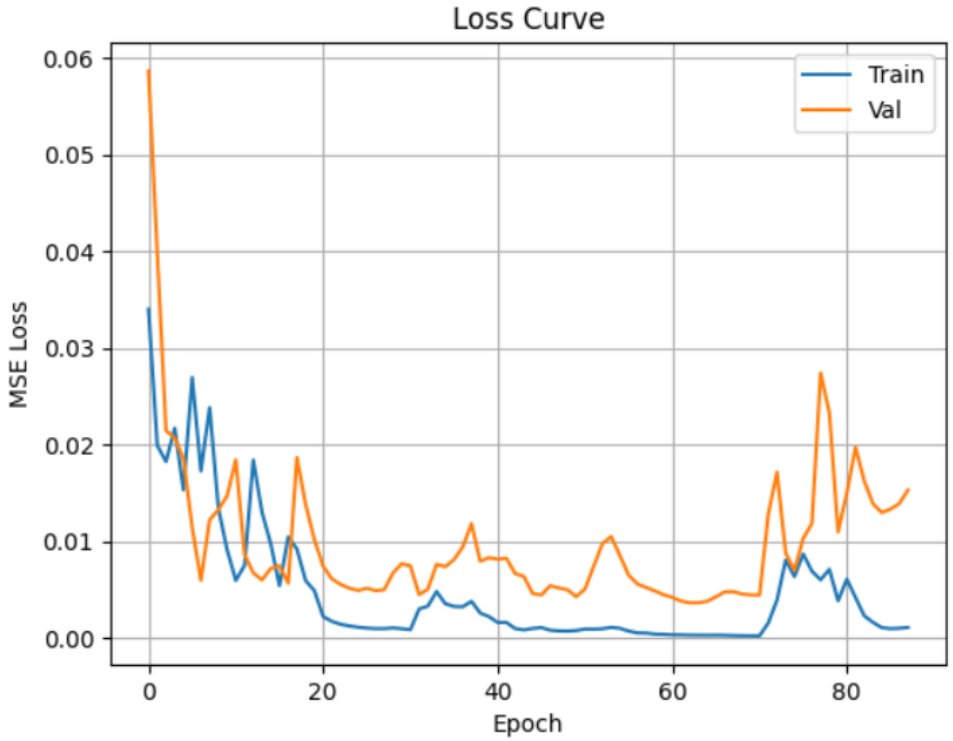


Figure 4.3: Training and validation loss curve for the GRU model with window size 10.

To mitigate this, future work may focus on optimizing the network architecture and training strategies. Adjustments such as tuning the dropout rate, employing more sophisticated early stopping criteria, and performing advanced feature engineering may help improve the model's robustness and reduce prediction variance.

Overall, the results support the use of recurrent neural networks, particularly GRU-based architectures, for modeling the thermal behavior of automotive braking systems.

Evaluation on Test Set:
The final evaluation on the test set, as we can see at table 4.5, confirms the GRU model's strong generalization capability. On the test set, the GRU model achieved significantly improved results, with a MAE of 15.97 °C, a RMSE of 25.09 °C and a R^2 of 0.9785. These scores reflect a substantial boost in predictive accuracy, indicating that the model is capable of explaining nearly 98% of the variance in the disc brake temperature data.

The use of an attention mechanism likely contributed to this performance improvement. By enabling the model to focus on the most relevant segments within the input time window, attention mitigates the inherent limitations of fixed window-based sequence modeling. This mechanism allows the model to assign greater importance to informative past time steps while down-weighting less relevant ones. As a result, the model benefits from enhanced accuracy and robustness, achieving strong generalization without overfitting as can clearly be seen in Figure 4.4.

Table 4.5: Evaluation Metrics for GRU Model on Test Set

Metric	Value
MAE	15.97 °C
RMSE	25.09 °C
R^2	0.9785

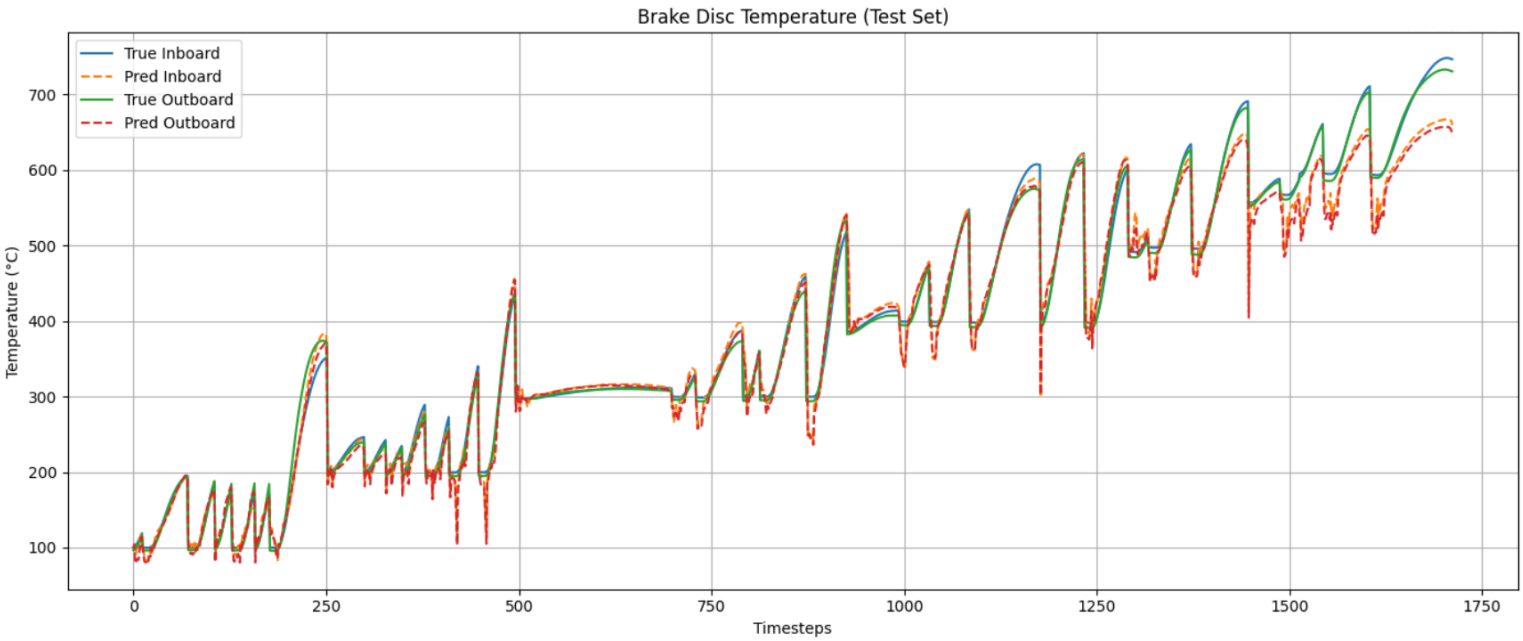


Figure 4.4: Temperature prediction for the GRU model with window size 10 on test set

Test 3: GRU - Attention Mechanism and Window Size = 10 (with Hyperparameter Modification)

Parameter	Value
Window Size	10
Hidden Dimension	128
Output Dimension	2
Batch Size	64
Epochs	200
Patience	35
Learning Rate	1×10^{-3}
Weight Decay	1×10^{-4}
Number of GRU Layers	2
Dropout	0.2

Table 4.6: Hyperparameters used for GRU model in Test 3

Evaluation on Validation Set:

Based on the Hyperparameters in 4.6, the results, shown in Table 4.7 indicate a significant improvement over earlier iterations. With both R^2 values exceeding 0.82, the model effectively captures more than 82% of the variance in the disc temperature data. Additionally, reductions in both RMSE and MAE indicate improved control over prediction outliers and lower average prediction errors. This is likely due to the balanced combination of dropout regularization and reduced model capacity, which helps in mitigating overfitting.

Table 4.7: Evaluation Metrics for GRU Model on Validation Set

Disc Type	**RMSE** (°C)	**MAE** (°C)	$\mathbf{R^2}$ Score
Disc Inboard	496.98	15.68	0.834
Disc Outboard	556.52	16.10	0.827

The updated GRU model, configured with a smaller hidden dimension (128) and increased dropout rate (0.2), demonstrates improved generalization behavior. As observed in the loss curves in Figure 4.5, both training and validation losses exhibit similar downward trends in the early epochs. The validation loss stabilizes after some fluctuations, suggesting that the model benefits from effective regularization and reduced overfitting, compared to previous configurations.

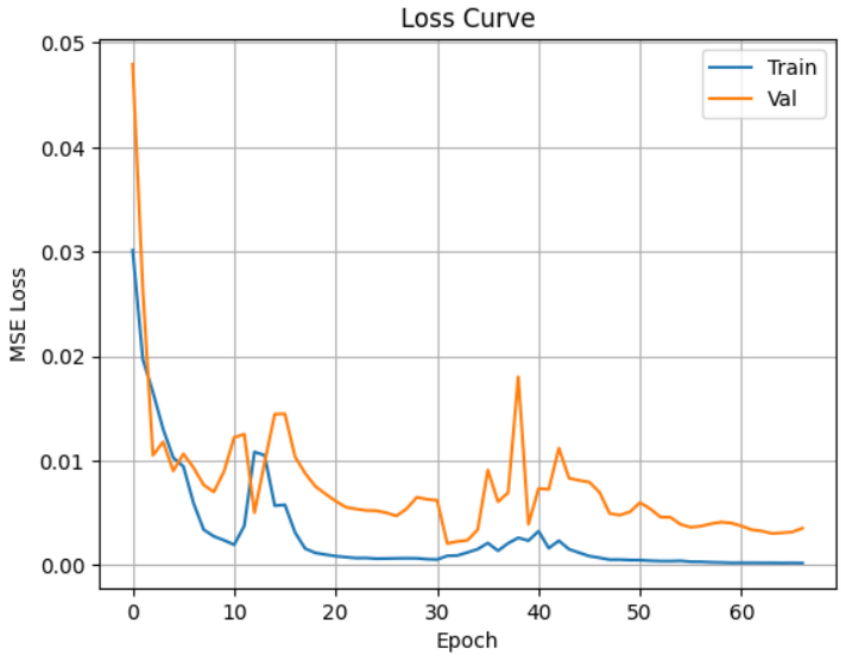


Figure 4.5: Training and validation loss curve for the GRU model with hidden dim 128 and dropout=0.2

Despite the relatively small window size of 10, the model captures relevant temporal patterns in the data without being overly sensitive to short-term noise. The slight increase in patience to 35 allows for more comprehensive convergence during training and avoids premature stopping. This could contribute to the model's stability and generalization.

Evaluation on Test Set:

These test metrics further validate the model's robustness. The use of an attention mechanism enhances the GRU's ability to learn context-aware temporal dependencies, by dynamically weighting the most relevant segments of the input sequence, this is clear from the resulting Figure 4.6, which is very accurate. As can be seen in the final evaluation metrics Table 4.8, with an R^2 nearing 0.97, the model captures nearly all of the variance in the test data. This indicates both high predictive accuracy and strong generalization capabilities.

Table 4.8: Final Evaluation Metrics on Test Set (GRU Model with Attention-Mechanism)

Metric	Value
MAE (°C)	22.31
RMSE (°C)	30.07
R^2 Score	0.9691

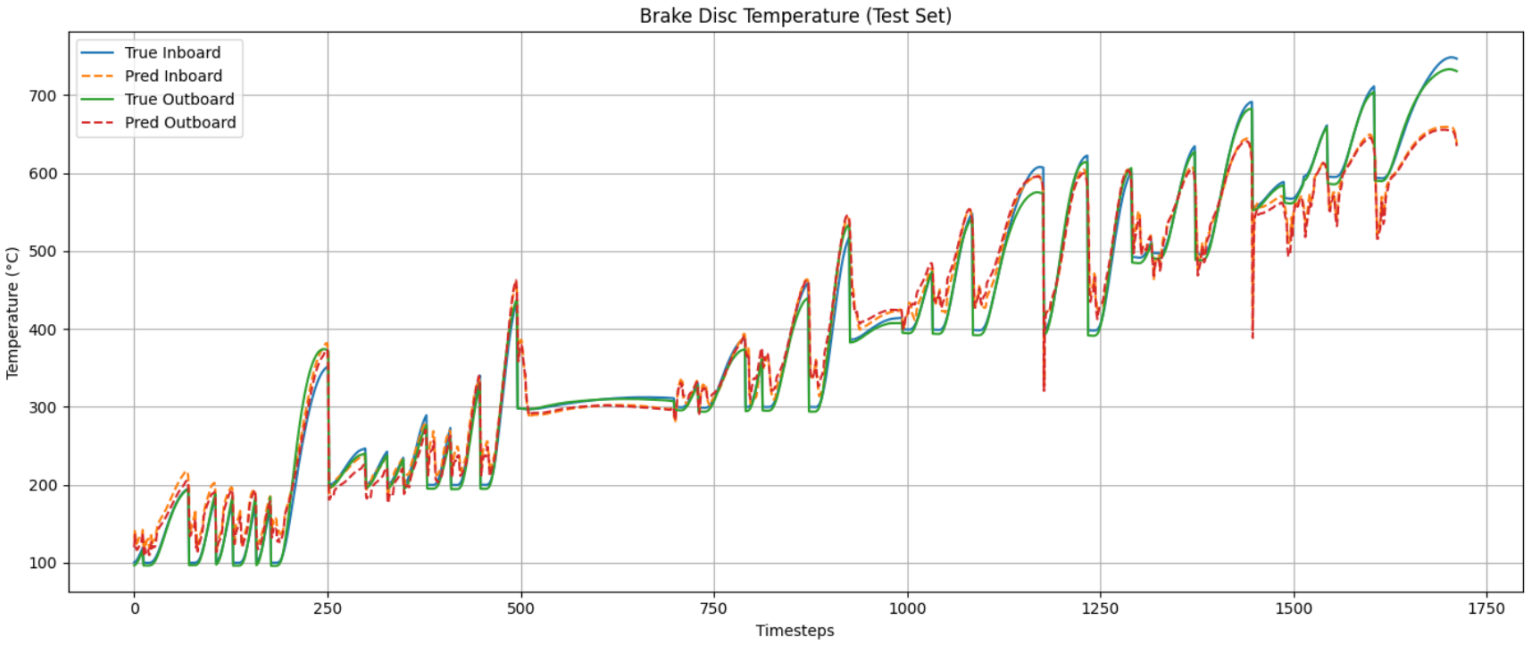


Figure 4.6: Temperature prediction for the GRU model with attention-mechanism on test set

Test 4: GRU vs Random Forest

For this experiment, the Random Forest Regressor was configured with its model-specific parameters given in Table 3.2, while the GRU model was evaluated using the hyperparameters listed in Table 4.9. A reduced patience value of 20 forces earlier stopping, potentially reducing overfitting, while a higher dropout rate of 0.3 enhances regularization by randomly disabling neuron connections during training. The increased batch size of 128 may also stabilize gradient updates, improving convergence for larger datasets. This comparative analysis against a Random Forest model was performed to highlight the benefits of temporal modeling via recurrent neural networks.

Hyperparameter	**Value**
Window Size	10
Hidden Dimension	128
Output Dimension	2
Batch Size	128
Epochs	200
Patience	20
Learning Rate	1×10^{-3}
Weight Decay	1×10^{-4}
Number of Layers	2
Dropout	0.3

Table 4.9: Hyperparameters used for Test 4 GRU model.

Evaluation on Validation Set

The GRU model demonstrated strong performance on the validation set. These metrics, shown in Table 4.10 indicate that the model can explain 93% and 88% of the variance for inboard and outboard disc temperatures, respectively. In contrast, the Random Forest baseline yielded poor results on the same validation data, with negative R^2 scores for both outputs. These negative values indicate performance worse than a naive mean predictor, highlighting the model's inability to capture temporal dependencies. This performance disparity reinforces the importance of using recurrent architectures such as GRUs for modeling time-dependent phenomena like thermal dynamics in braking systems.

Table 4.10: Comparison of GRU and Random Forest Models on Validation Set

Disc Type	**Model**	**RMSE** (°C)	**MAE** (°C)	$\mathbf{R^2}$ Score
Disc Inboard	GRU	208.22	9.15	0.930
Disc Outboard	GRU	378.48	12.95	0.882
Disc Inboard	RF	4800.88	47.89	-0.608
Disc Outboard	RF	4414.61	44.42	-0.373

Evaluation on Test Set:

The test evaluation results given in Table 4.11 clearly indicate that the GRU model outperforms the Random Forest model in predicting disc brake temperatures. the GRU model achieves significantly lower RMSE and MAE values, along with higher R^2 scores (0.980 and 0.981, respectively). These results reflect a more accurate and reliable performance in modeling the underlying temporal patterns of the temperature data. On the other hand, the Random Forest model shows considerably higher prediction errors and lower R^2 values, highlighting the limitations of traditional machine learning approaches for this time series forecasting task. The large offset observed in Figure 4.7 in the Random Forest predictions is likely because the model converges toward predicting values near the mean of the training data. This behavior minimizes error penalties in the absence of strong temporal modeling, resulting in predictions that follow the general trend but are vertically shifted from the true temperatures.

Table 4.11: Comparison of GRU and Random Forest Models on Test Set

Disc Type	**Model**	**RMSE** (°C)	**MAE** (°C)	$\mathbf{R^2}$ Score
Disc inboard	GRU	603.08	15.60	0.980
Disc outboard	GRU	561.01	15.30	0.981
Disc inboard	RF	2419.27	25.13	0.918
Disc outboard	RF	2097.94	22.95	0.928

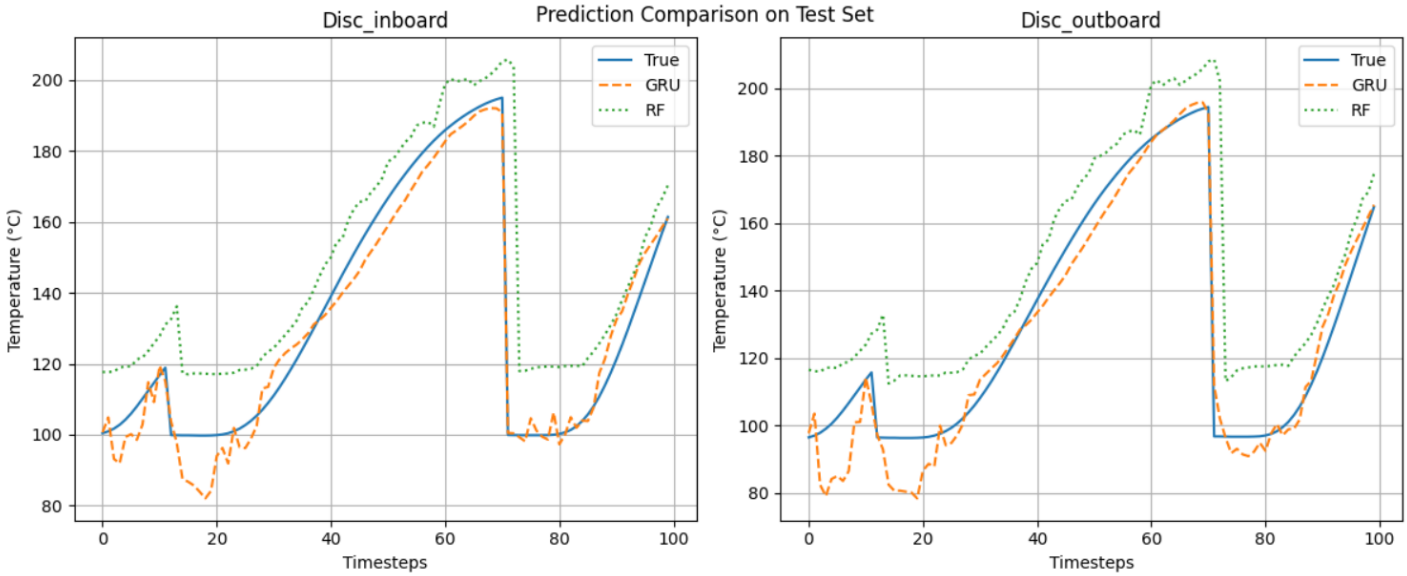


Figure 4.7: Temperature prediction for the GRU model with window size 10 on test set vs Random Forest

Test 5: GRU vs LSTM

Evaluation on Validation Set

The comparison between GRU of test 5 and LSTM baseline reveals a clear performance advantage in favor of the GRU architecture. For the disc inboard temperature prediction, see Table 4.12, GRU achieved a significantly lower RMSE of 208.219 and MAE of 9.15, compared to the LSTM baseline's RMSE of 759.87 and MAE of 19.40. The R^2 score for GRU was also higher at 0.930, indicating better model fit and predictive power.

Similarly, for the disc outboard temperature, GRU delivered an RMSE of 378.48 and an MAE of 12.95, outperforming LSTM baseline's RMSE of 1230.98 and MAE of 23.28. The improvement in R^2 from 0.617 (LSTM) to 0.882 (GRU) further underscores the GRU model's superior capacity to capture underlying temporal patterns and relationships.

Table 4.12: Comparison of GRU and LSTM Models

Disc Type	**Model**	**RMSE** (°C)	**MAE** (°C)	$\mathbf{R^2}$ Score
Disc inboard	GRU	208.22	9.15	0.930
Disc outboard	GRU	378.48	12.95	0.882
Disc inboard	LSTM	759.87	19.40	0.745
Disc outboard	LSTM	1230.98	23.28	0.617

Evaluation on Test Set

The performance of the GRU and LSTM models on the test set offers further insights into their respective strengths. The GRU model significantly outperformed the LSTM model in both disc inboard and disc outboard temperature predictions.

Specifically, if we look at the disc inboard on Table 4.13, GRU achieved an RMSE of 248.64 and an MAE of 10.37, whereas LSTM resulted in a much higher RMSE of 1666.85 and MAE of 31.27. Despite the LSTM showing a high R^2 value of 0.943, the error metrics indicate considerable deviation from actual values, likely due to overfitting or sensitivity to data fluctuations. For the outboard disc, the GRU model again showed stronger performance with an RMSE of 323.41 and MAE of 12.09, compared to the LSTM's RMSE of 1612.66 and MAE of 30.79. The GRU model achieved an impressive R^2 score of 0.989, suggesting a very close match between predicted and true values.

Table 4.13: Comparison of LSTM and GRU Models on Test Set

Disc Type	**Model**	**RMSE** (°C)	**MAE** (°C)	$\mathbf{R^2}$ Score
Disc inboard	GRU	248.64	10.37	0.992
Disc outboard	GRU	323.41	12.09	0.989
Disc inboard	LSTM	1666.85	31.27	0.943
Disc outboard	LSTM	1612.66	30.79	0.945

These results, shown in Figure 4.8 and 4.9 , confirm the GRU based models can provide highly accurate predictions for time series data in braking systems. GRUs offer a simpler yet effective alternative to LSTMs, likely due to their reduced parameter complexity and better handling of vanishing gradients. The large vertical offset in the LSTM prediction shown in Figure 4.9 is, similar to the Random Forest model, caused by a regression towards the mean of the training distribution, reducing error penalties but resulting in poor performance. The findings reinforce the GRU model's suitability for real time or embedded applications where computational efficiency and accuracy are both critical.

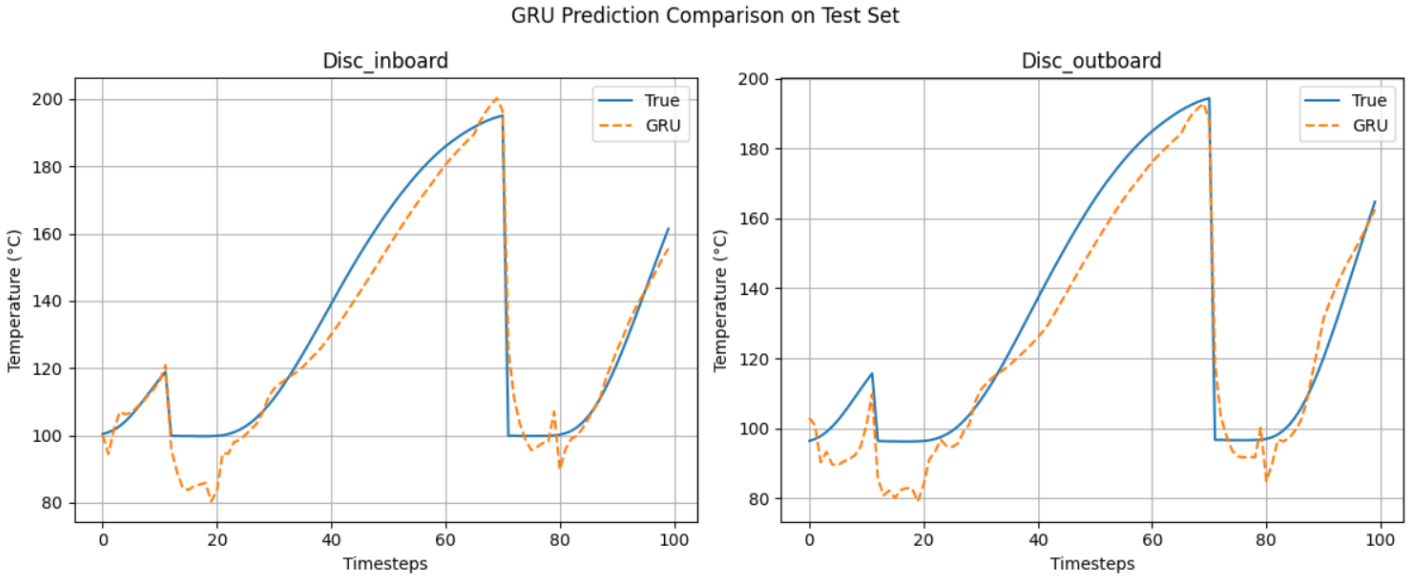


Figure 4.8: Temperature prediction for the GRU model with window size 10 on test set

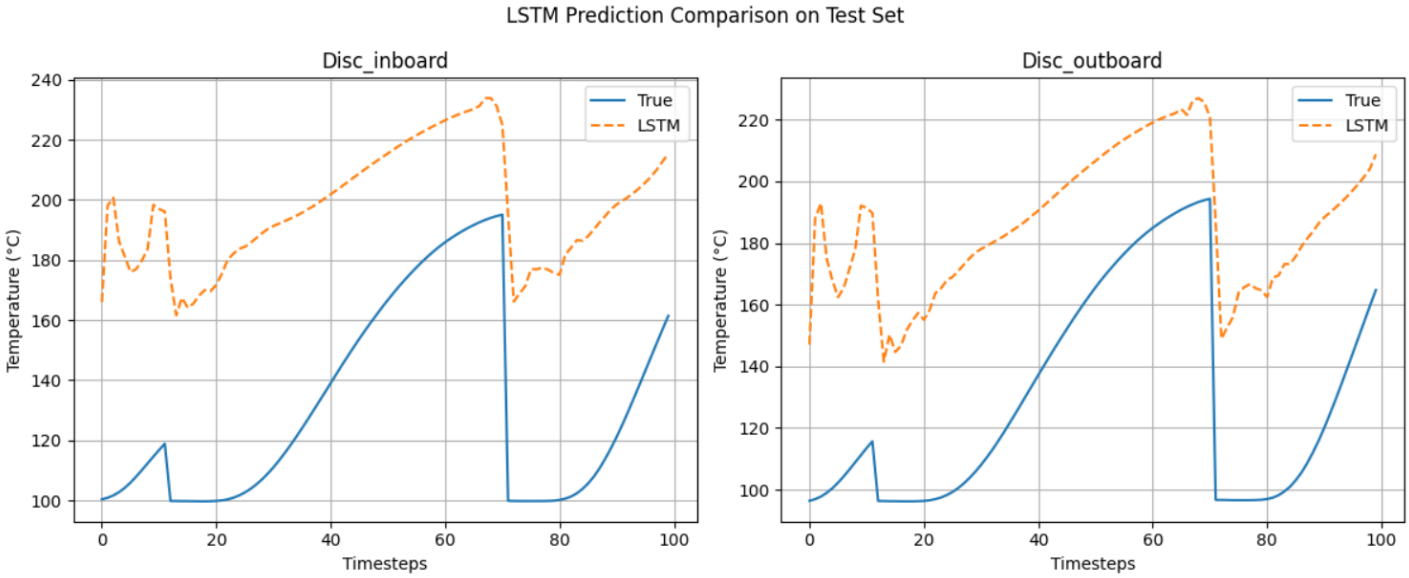


Figure 4.9: Temperature prediction for the LSTM model with window size 10 on test set

Test 6: GRU vs CNN

Evaluation on Validation Set

The GRU model significantly outperforms the CNN baseline on the validation set. The CNN model's poor performance, with negative R^2 scores, see Table 4.14, indicates a failure to generalize. Conversely, the GRU model captures over 88% of the variance in the outboard temperature and over 93% in the inboard temperature, demonstrating strong temporal pattern recognition.

Table 4.14: Validation Performance of GRU-Attention and CNN Models

Disc Type	**Model**	**RMSE** (°C)	**MAE** (°C)	$\mathbf{R^2}$ Score
Disc Inboard	GRU	208.22	9.15	0.930
Disc Outboard	GRU	378.48	12.95	0.882
Disc Inboard	CNN	16428.88	91.99	-4.503
Disc Outboard	CNN	16157.02	94.31	-4.026

Evaluation on Test Set

The final test results, see Figures 4.10 and 4.11, highlight the robustness and high accuracy of the GRU model. With R^2 values above 0.98, see Table 4.15, and low RMSE/MAE scores, the model generalizes well to unseen test data. In contrast, the CNN model's R^2 scores barely exceed 0.3, indicating that it captures only a small portion of the data's variance. This further emphasizes the point: GRU predictions remain tightly aligned with the ground truth, while CNN predictions become unstable and highly divergent. Overall, this experiment confirms the suitability of GRU-based architectures for modeling sequential temperature data in dynamic systems, especially when combined with attention mechanisms.

Table 4.15: Test Set Evaluation: CNN vs. GRU-Attention Models

Disc Type	Model	RMSE (°C)	MAE (°C)	R^2 Score
Disc inboard	GRU	248.64	10.37	0.992
Disc outboard	GRU	323.41	12.09	0.989
Disc inboard	CNN	20577.02	122.34	0.302
Disc outboard	CNN	19985.27	120.47	0.312

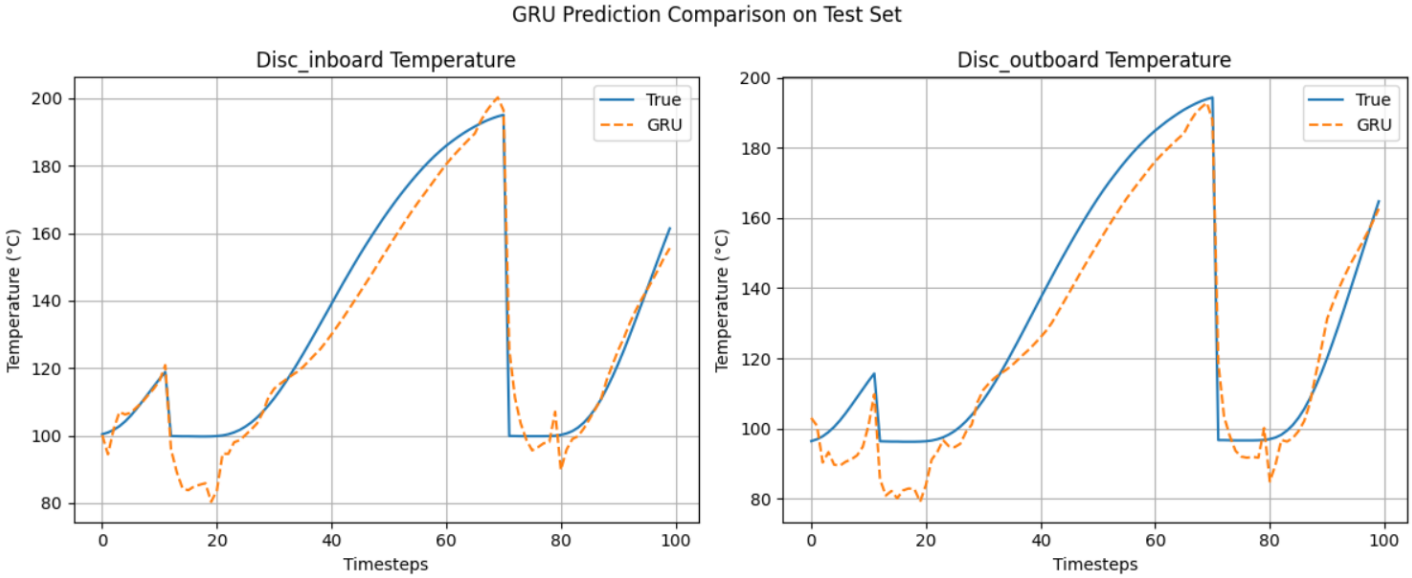


Figure 4.10: Temperature prediction for the GRU model with window size 10 on test set

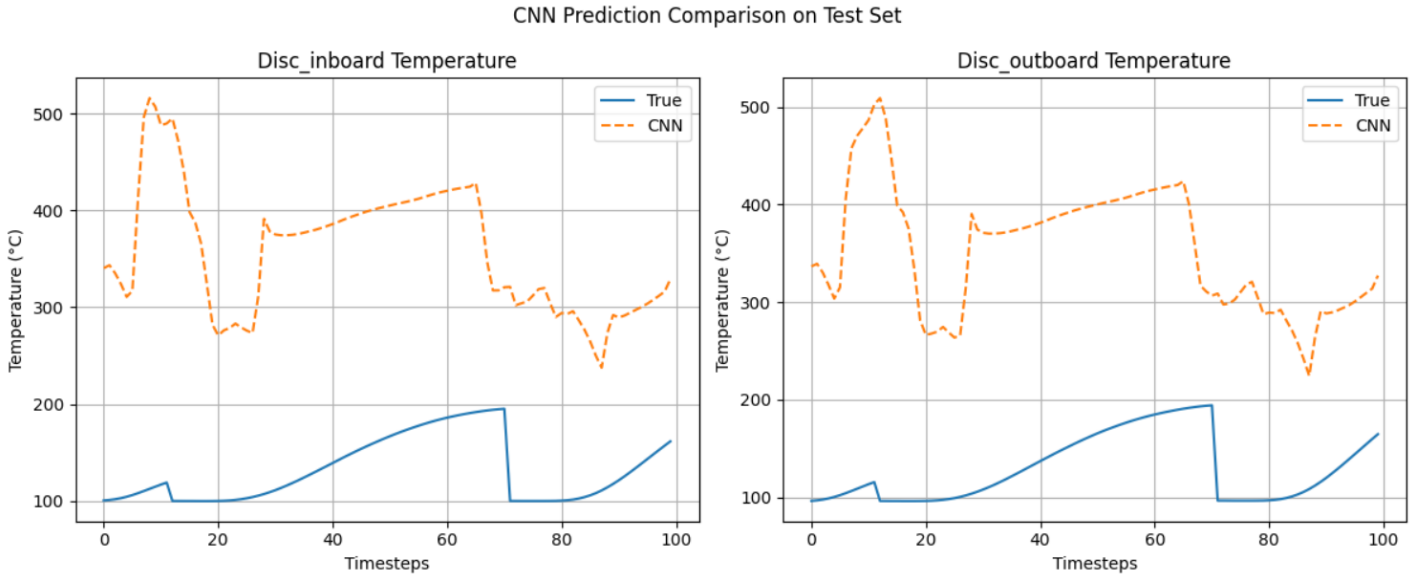


Figure 4.11: Temperature prediction for the CNN model with window size 10 on test set

Test 7: Field test results

Although this was beyond the formal scope of our thesis, we aimed to evaluate the performance of the GRU model from test 3 on real-world data by predicting the temperatures of the front left disc and a rear left disc. The field data was based on the England Cycle, a real-world test characterized by repeated braking and thermal load typical of UK hilly terrain. Unlike the dyno data, it also included cooling phases between active braking events, allowing the model to account for post-braking temperature drops. The field data provides a more realistic basis, improving the model's generalizability beyond the isolated braking events seen in the dyno data.

To conduct this analysis, we applied mutual information regression to select the top 60 features most influencing the target output, such as vehicle speed, ambient temperature, brake pressure and disc temperature. To prevent data leakage, a causal Gaussian filter was used, given the high noise levels in the data. Unlike standard filters, which use both past and future data points to compute a smoothed value at a given time step, a causal filter only uses current and past data, which is critical in time-series tasks for preventing data leakage. This preserved the temporal integrity of the data and helped accurately assess the model's real-world performance.

As shown in Table 4.16, the model achieved a MAE of 7.108, RMSE of 9.065 and an R^2 score of 0.988. This shows an excellent result and proves that the model performed better on the field data, as seen in Figures 4.12 and 4.13, than the dyno data. This is most likely because the dyno tests included extreme braking scenarios, with temperatures ranging from 99°C to nearly 750°C. In contrast, the field data reflected more typical driving conditions, with fewer extreme temperature variations.

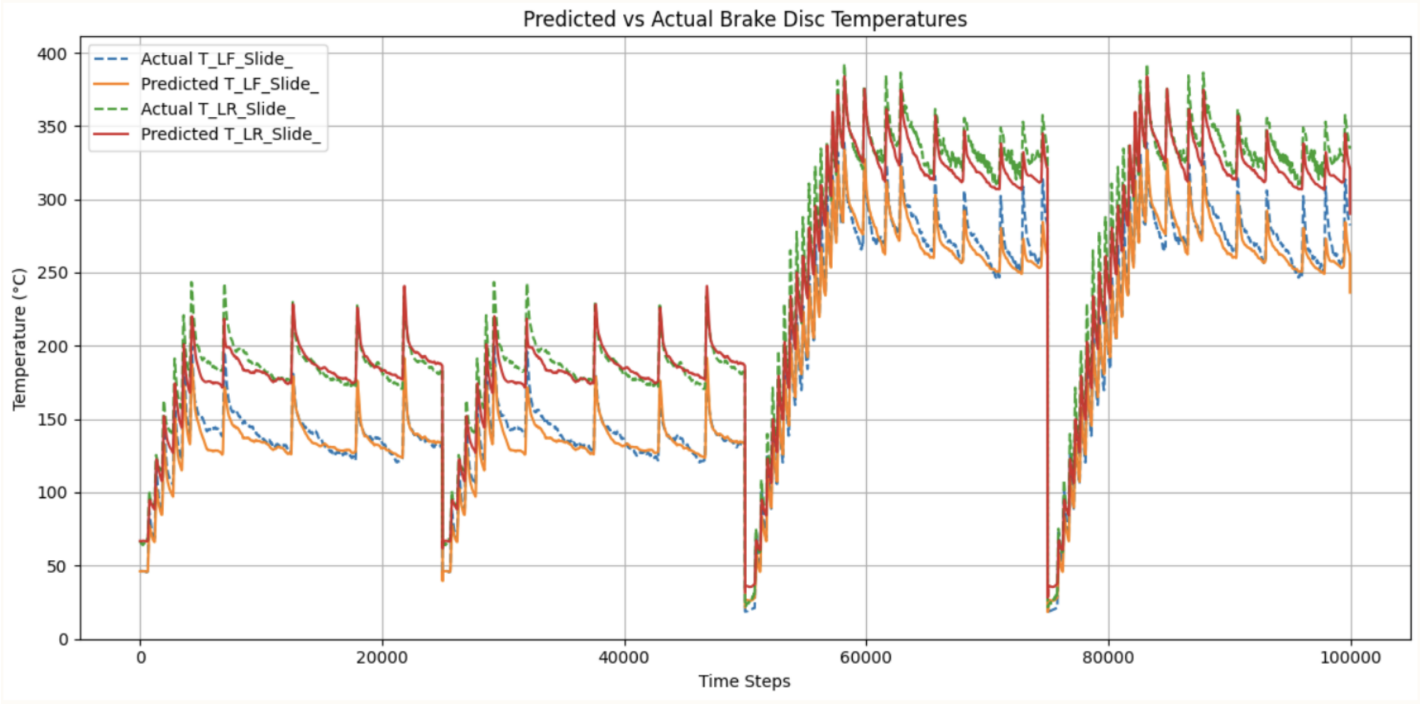


Figure 4.12: Temperature prediction for the GRU model on field data

Table 4.16: GRU Model Performance on Real-World Data

MAE (°C)	RMSE (°C)	R^2
7.108	9.065	0.988

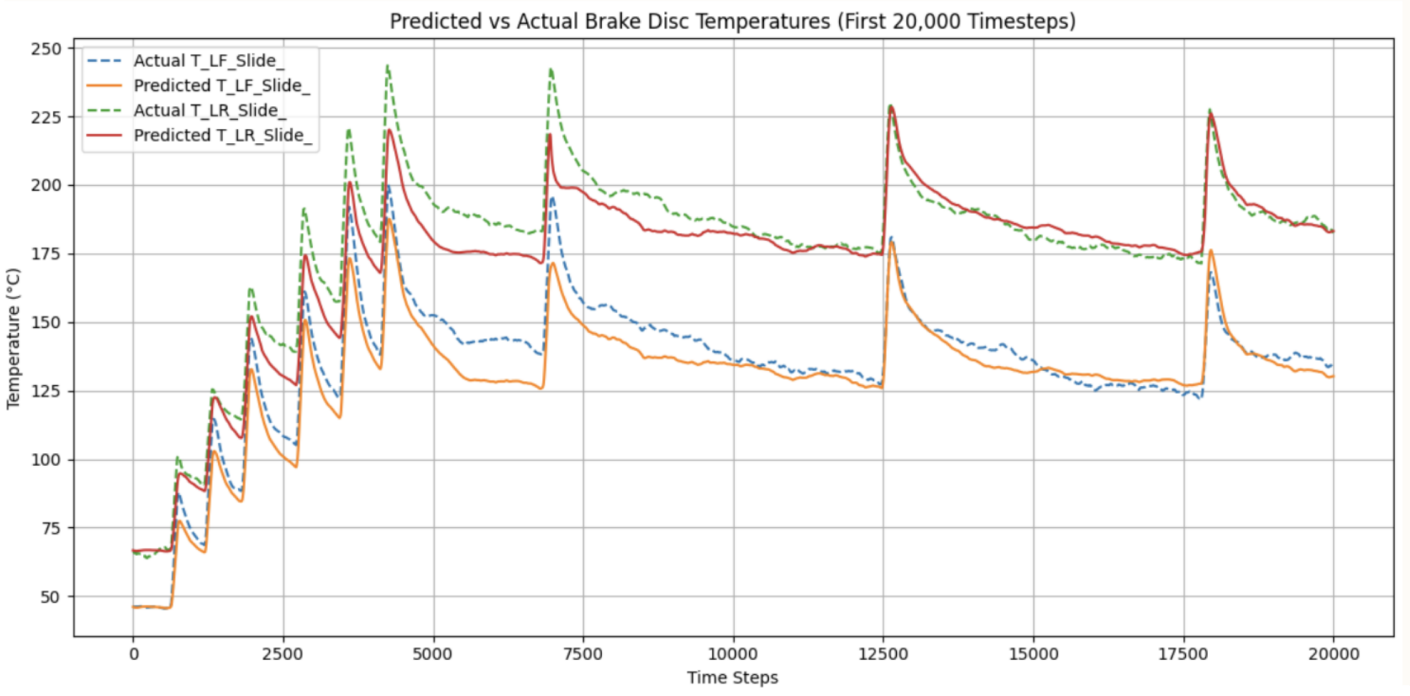


Figure 4.13: Temperature prediction for the GRU model on field data (zoomed)

5

Conclusion and Future Scope

This final chapter summarizes the key outcomes of the thesis and outlines future directions for improving brake temperature modeling in real-world applications. The work demonstrates that GRU-based models, when properly configured and evaluated under leak-free conditions, can accurately capture thermal dynamics across a wide range of braking events. These findings lay the groundwork for deploying real-time temperature prediction systems in heavy-duty vehicles and motivate further research into model generalization and physics integration.

5.1 Conclusion

This thesis investigated the application of GRU-based architectures for modeling the thermal behavior of disc brakes under dynamic conditions. Through a series of experiments involving different architectures, hyperparameter tuning and comparative evaluations, the GRU models proved to be highly effective, especially those enhanced with attention mechanisms.

The 2-stage GRU model with a larger temporal context delivered strong baseline performance, achieving an R^2 score of 0.9554 on the test set. By optimizing the network architecture and incorporating dropout regularization, subsequent models achieved further gains in predictive accuracy and generalization. Notably, the attention-augmented GRU models consistently outperformed their counterparts, reaching R^2 scores as high as 0.980 on unseen data. These models showed not only low MAE and RMSE values but also robustness against overfitting. The analysis revealed the clear superiority of GRU models over conventional machine learning approaches, such as Random Forests and CNNs. GRU also performed better than the even more complex and computationally heavy LSTM model. The GRU model with attention mechanism achieved significantly lower prediction errors and higher R^2 values in both validation and test scenarios, highlighting their ability to effectively capture long-range dependencies in time series data.

In summary, the experimental results confirm the suitability of GRU-based recurrent architectures, particularly those incorporating attention mechanisms, for accurately predicting disc brake temperatures. These models demonstrate high potential for integration into real-time monitoring and control systems within automotive braking applications, offering a practical blend of predictive power and computational efficiency.

5.2 Future Scope

To better replicate real-world conditions in the dyno data, future work must account for the entire thermal cycle, not just the braking phase. The dataset used only included active braking, with no cooling phases between braking events. In contrast, the field data captured these cooling phases, providing a more realistic view of disc temperature evolution. While conductive, convective and radiative cooling mechanisms are implicitly captured during braking itself, the absence of post-braking or coasting periods in the dyno data remains a key difference compared to the field data.

Additionally, proper modelling of cooling is essential. Ambient temperature, vehicle speed and airflow become major contributors to convective cooling, particularly through forced convection over ventilated discs. Radiative losses, though less dominant, also grow with temperature and can be significant in real braking scenarios. Without incorporating these effects that are captured in the field data, the model's ability to generalize to typical driving cycles remains limited. Integrating data from full braking and cooling phases, with sensors measuring ambient conditions and wheel speed, is therefore a prerequisite for mimicking field data in the dyno lab.

An additional direction explored in this project was the development of PINNs. A PINN was implemented by embedding a simplified axisymmetric heat equation into the network loss function. However, the impact of the physics term was minimal due to the lack of system-specific boundary conditions, such as accurate definitions of surface heat flux between disc and pad, and convection coefficients. As a result, the physics component had negligible influence on training, and the PINN behaved similarly to a conventional data-driven model.

This outcome highlights a key challenge in using PINNs for thermal modeling: physical accuracy requires not just the governing equations, but also the correct boundary conditions and heat flux formulations, similar to those used in FE-simulations. With further refinement, including more detailed physical constraints and system-specific parameters, PINNs could offer a powerful hybrid solution. Their ability to generalize with limited data, while enforcing known thermodynamic laws, makes them especially promising for industrial settings where experimental campaigns are expensive or limited. A well-calibrated PINN could eventually replace FE-simulations, allowing real-time, physically consistent brake temperature estimation. If properly configured with accurate physical constraints, PINNs become a viable option for deployment in live truck environments, offering a balance between interpretability and predictive performance.

One critical limitation of this study is that the dataset was collected using a single brake disc and pad configuration. While this setup ensured consistency across tests, it introduces epistemic uncertainty. Since the model has not encountered other friction materials or disc geometries, its ability to generalize to different braking systems remains untested. Aleatoric uncertainty, arising from measurement noise, is relatively low in the controlled dynamometer environment but would likely in-

crease in field applications due to sensor variability and environmental factors. In addition, the dataset featured a narrow distribution of initial speeds and disc temperatures, which limits the model's ability to learn behaviors under more extreme conditions. High-temperature data, in particular, was underrepresented, restricting the exposure to the thermal regimes most relevant for safety-critical braking scenarios. Future datasets should therefore include a broader range of initial conditions, braking system types, ambient temperatures and load cases.

The choice of temporal window size is another area with practical implications. Shorter windows may be desirable in low-latency environments and offer finer resolution of fast transients in disc temperature. However, they provide less context and can lead to instability in the prediction, especially in time series with long thermal memory. Larger windows, on the other hand, offer improved stability and context awareness but increase the computational burden and may smooth over rapid thermal events. Future work could explore adaptive window strategies or transformer-based models that process long contexts more efficiently. Investigating this tradeoff more systematically could yield insights into optimal configurations for different use cases.

To conclude, the results in this thesis offer a strong foundation for future real-time brake temperature prediction using GRU's. Future work should focus on integrating full thermal cycles, expanding dataset diversity and embedding physical knowledge more effectively. These improvements will be crucial to enabling safe and efficient deployment of predictive braking systems in real-world transport environments.

Bibliography

[1] G. Le Gigan, "On improvement of cast iron brake discs for heavy vehicles: Laboratory experiments, material modelling and fatigue life assessment," Ph.D. dissertation, Chalmers University of Technology, Göteborg, 2015.

[2] P. Jia and Q. Xin, "Compression-release engine brake modeling and braking performance simulation," *SAE Technical Paper*, vol. 2012-01-1968, 2012. DOI: 10.4271/2012-01-1968. [Online]. Available: https://www.sae.org/publications/technical-papers/content/2012-01-1968/.

[3] W. Zhang, Z. Zhang, Y. Zhang, and D. Wang, "Hydraulic retarders for heavy vehicles: Analysis of fluid mechanics and computational fluid dynamics on braking torque and temperature rise," *International Journal of Automotive Technology*, vol. 18, no. 3, pp. 437–446, 2017. DOI: 10.1007/s12239-017-0039-z. [Online]. Available: https://link.springer.com/article/10.1007/s12239-017-0039-z.

[4] W. Zhang, Y. Liu, and D. Wang, "Testing and performance analysis of an integrated electromagnetic retarder for heavy vehicles," *Robotics and Computer-Integrated Manufacturing*, vol. 84, p. 102 532, 2023. DOI: 10.1016/j.rcim.2023.102532. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0952197623010904.

[5] S. International, "Case studies of parking brake fires in commercial vehicles," *SAE Technical Paper*, vol. 2013-01-0207, 2013. DOI: 10.4271/2013-01-0207. [Online]. Available: https://www.sae.org/publications/technical-papers/content/2013-01-0207/.

[6] D. Wang, Y. Yang, W. Yu, J. Yong, and X. Dong, "Research on the hill-start assist of commercial vehicles based on electronic parking brake system," *Strojniški vestnik - Journal of Mechanical Engineering*, vol. 64, no. 1, pp. 3–11, 2018. DOI: 10.5545/sv-jme.2017.5422. [Online]. Available: https://www.researchgate.net/publication/333037765_Research_on_the_Hill-Start_Assist_of_Commercial_Vehicles_Based_on_Electronic_Parking_Brake_System.

[7] F. Talati and S. Jalalifar, "Analysis of heat conduction in a disk brake system," en, *Heat and Mass Transfer*, vol. 45, no. 8, pp. 1047–1059, Jun. 2009, ISSN: 0947-7411, 1432-1181. DOI: 10.1007/s00231-009-0476-y. [Online]. Available: http://link.springer.com/10.1007/s00231-009-0476-y (visited on 03/07/2025).

[8] H. Mazidi, S. Jalalifar, S. Jalalifar, and J. Chakhoo, "Mathematical Modeling of Heat Conduction in a Disk Brake System During Braking," *Asian Journal of Applied Sciences*, vol. 4, no. 2, pp. 119–136, Feb. 2011, ISSN: 19963343. DOI:

10.3923/ajaps.2011.119.136. [Online]. Available: https://www.scialert.net/abstract/?doi=ajaps.2011.119.136 (visited on 03/07/2025).

[9] J. F. Archard, "Contact and Rubbing of Flat Surfaces," en, *Journal of Applied Physics*, vol. 24, no. 8, pp. 981–988, Aug. 1953, ISSN: 0021-8979, 1089-7550. DOI: 10.1063/1.1721448. [Online]. Available: https://pubs.aip.org/jap/article/24/8/981/160178/Contact-and-Rubbing-of-Flat-Surfaces (visited on 06/01/2025).

[10] D. Chen and G. Zou, "Research on Contact Pressure Distribution of Pad," *MATEC Web of Conferences*, vol. 198, H. Ding and H. Yuan, Eds., p. 01 002, 2018, ISSN: 2261-236X. DOI: 10.1051/matecconf/201819801002. [Online]. Available: https://www.matec-conferences.org/10.1051/matecconf/201819801002 (visited on 06/01/2025).

[11] A. Day, "Chapter 7 - Thermal Effects in Friction Brakes," in *Braking of Road Vehicles*, A. Day, Ed., Oxford: Butterworth-Heinemann, Jan. 2014, pp. 215–258, ISBN: 9780123973146. DOI: 10.1016/B978-0-12-397314-6.00007-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123973146000073 (visited on 05/14/2025).

[12] F. P. Incropera, D. P. DeWitt, T. L. Bergman, and A. S. Lavine, Eds., *Fundamentals of heat and mass transfer*, eng, 6. ed. Hoboken, NJ: Wiley, 2007, ISBN: 9780471457282.

[13] R. He and Z. Jing, "Study on braking stability of commercial vehicles: An optimized air brake system," en, *Advances in Mechanical Engineering*, vol. 11, no. 5, p. 1 687 814 019 848 593, May 2019, ISSN: 1687-8132, 1687-8140. DOI: 10.1177/1687814019848593. [Online]. Available: https://journals.sagepub.com/doi/10.1177/1687814019848593 (visited on 06/01/2025).

[14] E. Umaras, A. Barari, and M. S. G. Tsuzuki, "Heavy Vehicles Brake Drums — An Accurate Evaluation on Thermal Loads in Severe Service Conditions," en, *International Journal of Automotive Technology*, vol. 22, no. 2, pp. 371–382, Apr. 2021, ISSN: 1976-3832. DOI: 10.1007/s12239-021-0035-1. [Online]. Available: https://doi.org/10.1007/s12239-021-0035-1 (visited on 06/01/2025).

[15] C. H. Gao, J. M. Huang, X. Z. Lin, and X. S. Tang, "Stress Analysis of Thermal Fatigue Fracture of Brake Disks Based on Thermomechanical Coupling," en, *Journal of Tribology*, vol. 129, no. 3, pp. 536–543, Jul. 2007, ISSN: 0742-4787, 1528-8897. [Online]. Available: https://www.researchgate.net/publication/245371241_Stress_Analysis_of_Thermal_Fatigue_Fracture_of_Brake_Disks_Based_on_Thermomechanical_Coupling (visited on 06/01/2025).

[16] M. Maleque, S. Dyuti, and M. Rahman, "Material selection method in design of automotive brake disc," *IAENG Transactions on Engineering Technologies*, vol. 1, pp. 1–6, 2006.

[17] V. Pavelčík and E. Kuba, "Application of basic machine learning algorithms in railway brake disc temperature prediction," *Transportation Research Procedia*, vol. 55, pp. 715–722, 2021. DOI: 10.1016/j.trpro.2021.07.040. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352146521004397.

[18] A. Yevtushenko, M. Kuciej, P. Grzes, and P. Wasilewski, "Temperature in the railway disc brake at a repetitive short-term mode of braking," *International Communications in Heat and Mass Transfer*, vol. 84, pp. 102–109, 2017. DOI: 10.1016/j.icheatmasstransfer.2017.04.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0735193317300684.

[19] Z. Luo and J. Zuo, "Conjugate heat transfer study on a ventilated disc of high-speed trains during braking," *Journal of Mechanical Science and Technology*, vol. 28, no. 5, pp. 1887–1897, 2014. DOI: 10.1007/s12206-014-0336-7.

[20] S. Panier, P. Dufrénoy, and D. Weichert, "An experimental investigation of hot spots in railway disc brakes," *Wear*, vol. 256, no. 7, pp. 764–773, 2004. DOI: 10.1016/S0043-1648(03)00459-9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0043164803004599.

[21] N. Kühl, M. Schemmer, M. Goutier, *et al.*, "Artificial intelligence and machine learning: Definitions, applications, and opportunities," *Business & Information Systems Engineering*, vol. 65, no. 1, pp. 3–17, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s12525-022-00598-0#citeas.

[22] V. R. Boppana, *Machine learning and ai learning: Understanding the revolution*, 2022. [Online]. Available: https://acadexpinnara.com/index.php/JIT/article/view/368/389.

[23] A. V. Joshi, *Machine Learning and Artificial Intelligence*. Cham: Springer, 2020. [Online]. Available: https://link.springer.com/content/pdf/10.1007/978-3-031-12282-8.pdf.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016. [Online]. Available: https://www.deeplearningbook.org/.

[25] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*, 2nd. Cham, Switzerland: Springer, 2018.

[26] M. H. Sazli, "A brief review of feed-forward neural networks," *Communications Faculty of Sciences University of Ankara Series A1 Mathematics and Statistics*, vol. 50, no. 2, pp. 11–17, 2006.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[28] A. Y. Ng, *Cs229 lecture notes: Supervised learning*, https://cs229.stanford.edu/lectures-spring2022/main_notes.pdf, Stanford University, Spring 2022, Stanford, CA, 2022.

[29] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, Atlanta, Georgia, USA, 2013. [Online]. Available: https://web.stanford.edu/~amaas/papers/relu_hybrid_icml2013_final.pdf.

[30] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *International Journal of Engineering Applied Sciences and Technology (IJEAST)*, vol. 4, no. 12, pp. 310–316, 2020, Published Online April 2020, ISSN: 2455-2143. [Online]. Available: http://www.ijeast.com.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. [Online]. Available: https://arxiv.org/abs/1412.6980.

[32] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2018.

[33] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[34] S. Cuomo, V. Schiano Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific machine learning through physics-informed neural networks: Where we are and what's next," *Journal of Scientific Computing*, vol. 92, no. 3, p. 88, 2022.

[35] R. Dey and F. M. Salem, "Gate-variants of Gated Recurrent Unit (GRU) neural networks," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, ISSN: 1558-3899, Aug. 2017, pp. 1597–1600. DOI: 10.1109/MWSCAS.2017.8053243. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8053243.

[36] K. Cho, B. v. Merrienboer, C. Gulcehre, *et al.*, *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*, arXiv:1406.1078, Sep. 2014. DOI: 10.48550/arXiv.1406.1078. [Online]. Available: http://arxiv.org/abs/1406.1078 (visited on 05/05/2025).

[37] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd. Springer, 2009.

[38] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[39] S. Kiranyaz, T. Ince, and M. Gabbouj, "1d convolutional neural networks and applications: A survey," *Mechanical Systems and Signal Processing*, vol. 151, p. 107 398, 2021.

[40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[41] B. Ghadimi, D. Cueva, T. Raïssi, X. Hilaire, B. Furet, and G. Clément, "Heat-flux on-line estimation in a locomotive brake disc using artificial neural networks," *International Journal of Thermal Sciences*, vol. 90, pp. 203–213, 2015.

[42] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks for heat transfer problems," *Journal of Heat Transfer*, vol. 143, no. 6, p. 060 801, 2021. DOI: 10.1115/1.4050542.

[43] N. Grochevaia, "Prediction of brake squeal: A deep learning approach analysis by means of recurrent neural networks," Master's thesis, Chalmers University of Technology, Gothenburg, Sweden, 2020.

[44] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, SciPy, 2010, pp. 51–56. [Online]. Available: https://pandas.pydata.org.

[45] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: A structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011. [Online]. Available: https://numpy.org.

[46] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: `https://scikit-learn.org`.

[47] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, 2019, pp. 8024–8035. [Online]. Available: `https://pytorch.org`.

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY