INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL

isep

Manutenção Preditiva Contínua para Diagnóstico de Falhas Ferroviárias

DIOGO FERREIRA RISCA

Outubro de 2024

P.PORTO

Lifelong Predictive Maintenance for Railway Fault Diagnosis

Diogo Ferreira Risca
Student No.: 1220186

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialization Area of Artificial Inteligence Engineering

Supervisor: Ana Maria Neves Almeida Baptista Figueiredo
Co-Supervisor: Maria Goreti Carvalho Marreiros

Evaluation Committee:
President:
Diogo Emanuel Pereira Martinho, Professor Adjunto do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Members:
Doutor Jorge Alexandre Novo Meira, Professor Adjunto Convidado do Instituto Superior de Engenharia do Instituto Politécnico do Porto
Ana Maria Neves Almeida Baptista Figueiredo, Professora Coordenadora do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Porto, October 6, 2024

Abstract

The integration of advanced sensor technologies with machine learning algorithms has revolutionized fault diagnosis in railway systems, particularly at the wheel-track interface. Although numerous models have been proposed to detect irregularities such as wheel out-of-roundness, they often fall short in real-world applications due to the dynamic and nonstationary nature of railway operations. This thesis explores the challenges and opportunities of applying continual learning for predictive maintenance in railway systems, where the model's ability to share knowledge between domains is critical to improving performance over time. By allowing the model to continuously learn and adapt as new data become available, continual learning overcomes the issue of catastrophic forgetting, which often plagues traditional models. The model retains past knowledge while improving predictive accuracy with each new learning episode, leveraging knowledge sharing mechanisms to adapt to evolving operational conditions, such as changes in speed, load, and track irregularities. Techniques such as experience replay and regularization-based strategies enhance model performance across multiple domains, making it particularly suitable for complex real-world environments. The methodology is validated through comprehensive simulations of train-track dynamic interactions, which capture realistic railway operating conditions. The proposed model demonstrates significant improvements in identifying wheel defects and other irregularities, establishing a reliable sequence for maintenance interventions. Future work will focus on field trials to assess the robustness of the approach in real-world railway environments, including challenges posed by track environments such as bridges and tunnels.

Keywords: Continual Learning; Predictive Maintenance; Railway; Fault Diagnosis; Deep Learning

Resumo

A integração de tecnologias avançadas de sensores com algoritmos de aprendizagem automática revolucionou o diagnóstico de falhas em sistemas ferroviários, particularmente na interface roda-linha. Embora tenham sido propostos vários modelos para detetar irregularidades, como a falta de circularidade das rodas, estes modelos são frequentemente insuficientes em aplicações reais devido à natureza dinâmica e não estacionária das operações ferroviárias. Esta tese explora os desafios e oportunidades da aplicação da aprendizagem contínua à manutenção preditiva em sistemas ferroviários, onde a capacidade do modelo para partilhar conhecimentos entre domínios é fundamental para melhorar o desempenho ao longo do tempo. Ao permitir que o modelo aprenda e se adapte continuamente à medida que novos dados se tornam disponíveis, a aprendizagem contínua supera a questão do *catastrophic forgetting*, que frequentemente afeta os modelos tradicionais. O modelo retém o conhecimento passado, e melhora a precisão da previsão com cada novo episódio de aprendizagem, e aproveita sempre os mecanismos de partilha de conhecimento para se adaptar à evolução das condições operacionais, tais como alterações na velocidade, carga e irregularidades na linha. Técnicas como *experience replay* e estratégias baseadas na regularização melhoram o desempenho do modelo em vários domínios, tornando-o particularmente adequado para ambientes complexos do mundo real. A metodologia é validada através de simulações exaustivas de interações dinâmicas comboio-linha, que captam condições de funcionamento ferroviário realistas. O modelo proposto demonstra melhorias significativas na identificação de defeitos nas rodas e outras irregularidades, e estabelece uma sequência fiável para intervenções de manutenção. O trabalho futuro centrar-se-á em ensaios no terreno para avaliar a robustez da abordagem em ambientes ferroviários reais, incluindo os desafios colocados por ambientes de linha como pontes e túneis.

Palavras-chave: Continual Learning; Predictive Maintenance; Railway; Fault Diagnosis; Deep Learning

Acknowledgement

It gives me great pleasure to present the thesis report in its completed form.

This work has been supported by the European Union under the Next Generation EU, through a grant of the Portuguese Republic's Recovery and Resilience Plan (PRR) Partnership Agreement, within the scope of the project PRODUTECH R3. It has received Portuguese National Funds through the Portuguese Foundation for Science and Technology under the project UIDP/00760/2020 (https://doi.org/10.54499/UIDP/00760/2020).

First of all, I would like to express my sincere gratitude to my supervisors, Professor Ana Almeida, whose invaluable guidance and teachings greatly contributed to the success of this work, and Professor Goreti Marreiros for her guidance and, more importantly, for the moral support she provided throughout this research work. Without her timely advice and deep knowledge in machine learning, this research would not have been possible with such success. I am extremely grateful for their unwavering support.

I also extend my sincere gratitude to my colleague, Ph.D. candidate Afonso Lourenço, for his invaluable support throughout this project. His expertise and thoughtful guidance have significantly improved my understanding of the field. This work would not have been the same without his generous assistance and insightful contributions.

I am deeply grateful to my master's team members, João Azevedo, Miguel Panda, and Gonçalo Lopes, as well as to my colleagues at GECAD, João Rodrigo, Francisca Osório, and Lara Neves, for their incredible support throughout this journey.

I appreciate the provision and support on data from CONSTRUCT-LESE, which was invaluable for this research.

Finally, I am profoundly grateful to my family and friends for standing by me over the years, for their patience, support, and constant encouragement.

Contents

List of Figures

List of Tables

List of Acronyms

AI	artificial intelligence.
CL	continual learning.
CNNs	convolutional neural networks.
DL	deep learning.
DNNs	deep neural networks.
ER	experience replay.
EWC	elastic weight consolidation.
IoT	internet of things.
LwF	learning without forgetting.
ML	machine learning.
MTF	Markov transition field.
PdM	predictive maintenance.
SI	synaptic intelligence.

Chapter 1

Introduction

The railway sector relies on reliability and maintainability to ensure smooth and secure passenger and freight transport (Hidirov and Guler 2019). With trains designed to operate for decades and maintenance costs accounting for a significant portion of total expenditure, there is an urgent need to keep safety and quality of service at an optimal level while minimizing operational costs. In fact, with high traffic levels, massive axle loads, and constantly changing conditions, even minor defects on the railway track can become significant problems. In particular, the wheel-rail interface has received a lot of attention in the literature, since it incurs most of the maintenance cost for both the railway vehicles and the infrastructure (Fröhling and Hettasch 2010). Examples of such components include wheels, rails, sleepers, fastenings, and ballast, among others. These components are exposed to various environmental and operational conditions, which can cause corrosion, cracks, and other forms of damage (Gonçalves et al. 2023; Guedes et al. 2023; Mosleh, Pedro Aires Montenegro, et al. 2021). In addition, the surfaces of the wheel-rail are subjected to high contact, sliding, and stress stresses in rolling contact, with increasing risk function being a common pattern observed in the failure rates of these mechanical components (Chong, Lee, and H. J. Shin 2010; Mohammadi et al. 2023). To address this increasing probability of failure increase over time, it is imperative to perform maintenance actions proactively.

By implementing a preventive maintenance strategy, railway operators can reduce the likelihood of downtime, increase the lifespan of the equipment, and avoid costly emergency repairs. The railway industry often uses periodic preventive maintenance based on the tonnage or kilometers traveled by vehicles and the accumulated load or passing traffic on the tracks (Lagnebäck 2007). However, these maintenance interventions are intuitively affected by expert knowledge on the effect of structural parameters on the degradation rate. For example, a change in rail profiles, closeness to switches, and direct-fixing systems increases the rate of degradation. In addition, the railway industry tends to have a comprehensive maintenance plan by proactively lubricating, refurbishing, calibrating, tamping, and driving-by visual inspecting equipment on a regularly scheduled basis. However, these plans tend to be too conservative to compensate for the fact that they fail to address the specific root cause, which not only results in very high maintenance costs, but can also reduce the component life. For example, it has been proven that despite altering the geometry of the tracks and improving the condition of the track geometry, excessive ballast settlement can degrade the structural parameters of railway systems (Barke and Chiu 2005). Ideally, the maintenance action should be performed just before the machine failure. To achieve this goal, onboard and side condition monitoring is increasingly being embraced with analysis tools to assess rail load (Alemi, Corman, and Lodewijks 2017).

The condition of a system is quantified by periodically or even continuously obtaining information from the sensors mounted on the components, taking actions only when there is evidence of abnormal behaviors. However, condition-based maintenance strategies are more expensive and thus only beneficial for crucial components whose failure does not occur instantaneously and leads to severe function loss and safety risk. In recent years, the railway industry has witnessed the emergence of low-cost and easily maintainable condition monitoring systems, enabled by digital automation systems, big data, and industrial internet of things (IoT) technologies. This has led to a growing trend of using artificial intelligence (AI) for condition-based maintenance, and numerous publications have explored this approach (Xie et al. 2020).

Over the years, industries have relied mainly on preventive and corrective maintenance strategies, but both have drawbacks. Preventive maintenance involves regularly scheduled checks and servicing to prevent potential failures before they occur. Although this sounds good in theory, it can sometimes lead to unnecessary work and higher costs if not managed carefully (Swanson 2001; Waeyenbergh and Pintelon 2002). However, corrective maintenance waits until something breaks before fixing it, which can cause unexpected downtime and expensive repairs. As industrial systems become more complex and the need for efficiency increases, these limitations have become more apparent (Jardine, D. Lin, and Banjevic 2006). This has sparked a shift towards more proactive strategies such as predictive maintenance (PdM), which uses data analysis and real-time monitoring to predict and prevent failures before they occur (Selcuk 2017). This approach is increasingly based on on-board and side condition monitoring and is increasingly being embraced by analysis tools to assess rail load (Alemi, Corman, and Lodewijks 2017).

PdM is regarded as a proactive strategy developed to predict when a component is likely to fail using data analytics and machine learning (ML). By forecasting potential failures, maintenance can be scheduled before the anticipated time of failure. This approach reduces planned and unplanned downtime, reduces maintenance costs, and extends the remaining useful life of the components (Molęda et al. 2023; Pech, Vrchota, and Bednář 2021).

Despite these efforts, the implementation of data mining techniques for railway maintenance has proven challenging due to the lack of a common framework and standards. To address this issue, this thesis investigates the challenges and opportunities of employing AI for condition-based maintenance at the wheel-track interface of railway systems, with a particular emphasis on continual learning (CL), and proposes potential solutions to address these challenges.

The strategies developed were validated with 3D numerical train-track dynamic interaction simulations, carried out with the in-house software Vehicle-Structure Interaction Analysis (P. A. Montenegro et al. 2015). By defining unknown variables one by one and observing how the methodology deals with the effect of these unknown parameters on responses, the decision support system can be deemed reliable. Ultimately, the proposed methodologies aim to be generic and with successful validation have the potential to be applied to real experimental data considering different types of trains and wheel defects. With regard to the scope and aim of this thesis, it is worth noting that the maintenance paradigm encompasses a broad range of activities, from data collection to maintenance scheduling and strategic decision-making supported by cost analysis.

1.1 Problem Description

Despite significant advances in fault diagnosis systems for railway applications, existing models still face several key challenges. There is a lack of models that efficiently leverage knowledge sharing between domains in a CL context, which is critical to improve performance over time, especially in complex real-world scenarios. In this work, we addressed these gaps by developing and testing a CL framework capable of adapting to multiple domains, handling different train configurations and improving performance despite variability in operational conditions. Recent advances in PdM, such as the use of automated green ML, further enhance the fault detection process by employing automated hyperparameter tuning to detect anomalies in real time, while also improving resource efficiency and sustainability. This approach has been applied to railway systems, showing improved accuracy in the detection of wheel defects under varying operating conditions (Lourenço, Ferraz, Meira, et al. 2023).

By visual inspection of the signals we get from the sensors, it becomes clear which shows an anomalous wheel, as shown in Figure 1.1. By comparing multiple examples of signals, normal and defective, the variations become apparent, allowing us to identify patterns indicative of wheel damage.

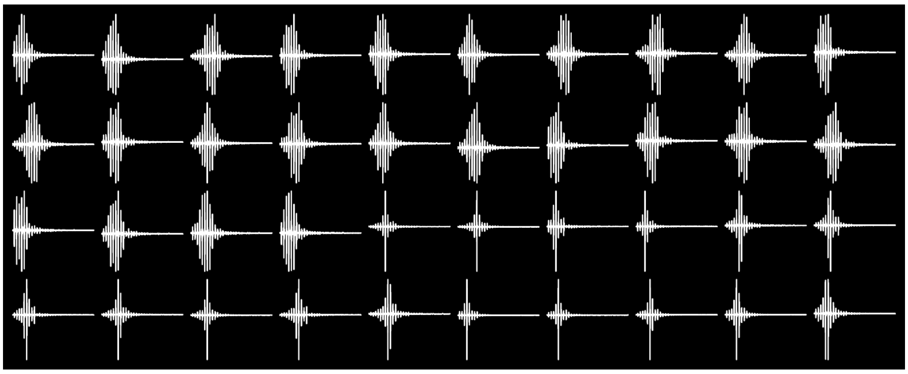


Figure 1.1: Display of signal patterns, with both normal and anomalous signals. Defective signals typically show noticeable spikes or irregularities, while normal signals maintain consistent shapes throughout.

Inspired by this, we propose to convert time series to images and use a convolutional neural network (CNN). This allows for invariant size of the image and other advantages. CNNs are adept at recognizing patterns, and this has been attested through their performance in classifying images. Using the power of such a network, it becomes quite plausible to learn time dynamics and spatial relationships within a time series dataset. CNN applies 1D convolution on the time axis, which allows the network to be trained on the temporal features and dependencies present in the data. It can capture various patterns and correlations working at multiple time scales through the use of a bank of convolutional filters. The model then leverages the strengths of CNNs in automatically and adaptively learning the spatial hierarchies of features from raw time series data, which have proved to be very effective in several domains for the analysis of transformed time series data. More specifically, dimensionality reduction using the pooling layers of CNN reduces the dimensions in the data, thus speeding up the computation and mitigating overfitting to have robust and accurate fault detection in railway systems (Patil and Rane 2021).

CNNs are one of the classes of strategies in the deep learning (DL) model designed to process data with grid-like topology and, particularly, image processing (LeCun, Bengio, and G. Hinton 2015; Schmidhuber 2015). These consist of several layers, convolutional layers for applying filters to detect features, pooling layers to reduce dimensionality, and fully connected layers to perform classification. CNNs are very good at learning automatically spatial hierarchies of features from the input images, especially being effective in procedures involving image recognition and object detection. This is what makes it possible for a really good job on image classification benchmarks like MNIST and ImageNet with the performance of the CNNs, outperforming traditional ML techniques by using the spatial correlations of pixels in images.

One of the key strengths of CNNs is the ability to extract features from complex patterns from visual data, a capability demonstrated by their state-of-the-art performance on data sets like MNIST (Alvear-Sandoval, Sancho-Gómez, and Figueiras-Vidal 2019; Chauhan, Ghanshala, and Joshi 2018; Jiang et al. 2019). The MNIST dataset consists of images of handwritten digits (LeCun, Cortes, and Burges 2010), as demonstrated in Figure 1.2 and demonstrates the way CNN learns and recognizes patterns and shapes within the data.

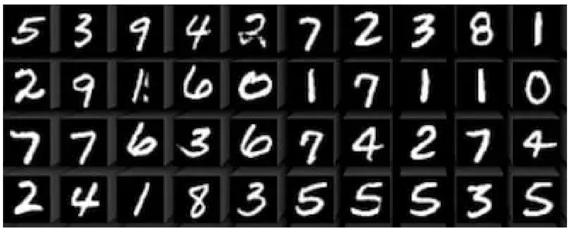


Figure 1.2: Images from the MNIST training set

In this thesis, rather than directly using the raw time series data, we transformed the signals into Markov transition field (MTF) images. MTF is a method to visualize time series data that encodes temporal dependencies and patterns into a 2D matrix, making it suitable for CNN-based analysis. This transformation process is similar to how CNNs analyze handwritten digits in MNIST, as it enables the network to recognize intricate patterns within the time series once they are converted into visual form. Applying CNNs directly to raw time series can be challenging due to its inherent complexities, such as non-stationarity and autocorrelation, but visual representations like MTF images simplify this process. This ability enhances the scope under which CNNs can be applied and increases their utility in areas such as PdM on the railway, which always includes understanding and interpreting time series data (Fawaz et al. 2019; Jardine, D. Lin, and Banjevic 2006; LeCun, Bottou, et al. 1998).

Figure 1.3 presents a grid of MTF images created from the original time series data, showcasing the different patterns that CNNs can analyze. This approach enhances the applicability of CNNs to time series domains, such as PdM in the railway, where understanding and interpreting time-dependent data is critical.

Beyond dealing with the challenge of capturing a time series structure, the dynamic nature of railway operations presents significant challenges for PdM. Rail conditions are never static, as their state is in constant flux due to weather, wear and tear, and traffic variation. This requires an adaptive maintenance approach that goes beyond traditional predictive models. Here, systems of continual learning (CL) step in as a subfield of ML. It works toward developing algorithms that learn from continuous data streams to adapt to any newly available information (plasticity) while keeping the knowledge that has already been acquired (stability) (Beaulieu et al. 2020; Hadsell et al. 2020; G. Lin, Chu, and Lai 2022; Mirzadeh et

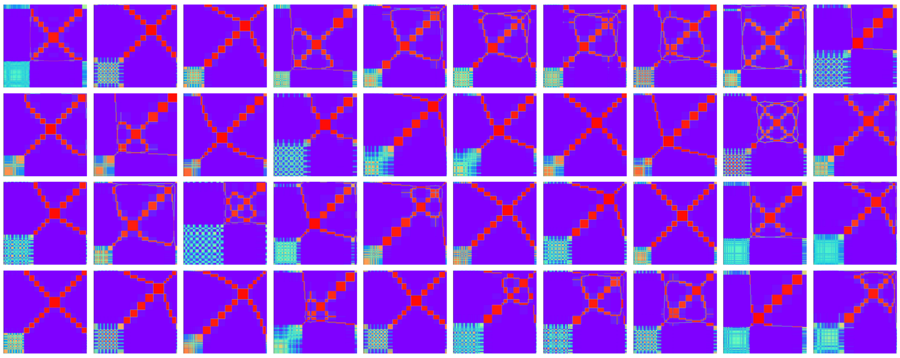


Figure 1.3: Grid of MTF images created from the original time series data, visually representing the temporal transitions and dependencies. These MTF images allow CNNs to analyze complex patterns and anomalies in the time series.

al. 2020). CL-enabled PdM systems can evolve in parallel with these changing conditions, continuously improving the accuracy of failure predictions.

A CL system could learn to target new wheels wear patterns caused by changing track conditions or environmental factors such as weather or load variations. As these domains change over time, the system needs to adapt to be able to detect anomalies that were not present in past data. Using CL allows PdM systems to continuously adjust to these varying operational and environmental domains, making sure that they remain effective in different scenarios. It is also worth noting that trains operating at higher speeds with heavier loads tend to experience larger impacts from wheel-rail irregularities, while lighter trains with slower speeds exhibit different tolerance thresholds. These factors are critical in understanding the performance of fault diagnosis models, as the ability to detect and predict irregularities depends on capturing the distinct dynamics of each train's operational profile. The varying tolerances to irregularities require models that adapt to different scenarios, ensuring robust detection across diverse conditions. With this, we can improve the effectiveness and adaptability of railway maintenance strategies, since detecting anomalies under constantly shifting conditions is mandatory, as these shifts happen every day with or without prior notice. Figures 1.4 and 1.5 illustrate how these domains change over time with different operational and environmental variations, showcasing the need for continual adaptation.

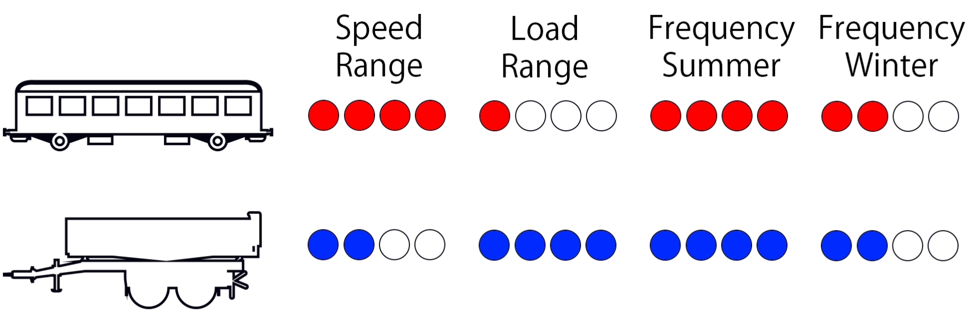


Figure 1.4: Variations of Operational and Environmental Railway Domains (Top - Alfa | Bottom - Laggrss)

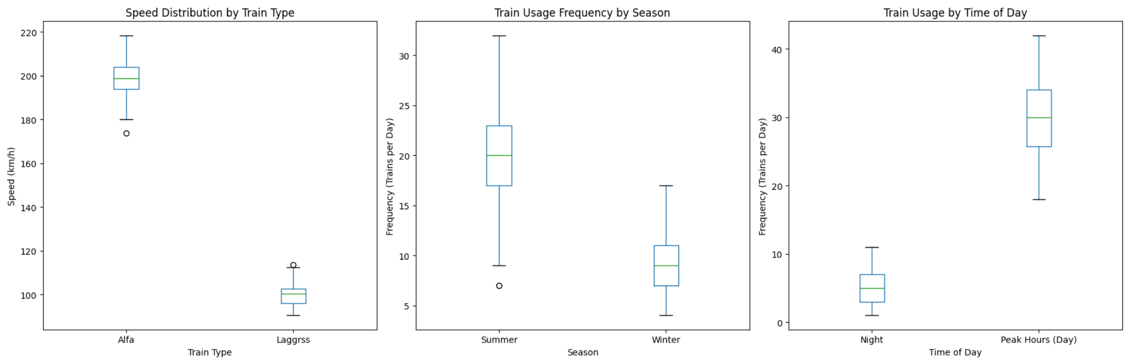


Figure 1.5: Boxplots Showing Variation in Train Operations Across Different Domains

1.2 Overview

Model sharing across domains lies on a spectrum, where full sharing and no sharing, the most common approaches to the problem of CL, form the two extrema, and compositionality is somewhere in between. Developing algorithms that can autonomously recover where the optimal solution to a set (or sequence) of learning domains lies on this spectrum remains an open challenge.

Instead, we adopted the option of a separate model for some groups of similar domains. The inspiration comes from a hypothesis that has been formulated and tested by psychological researchers (context, learning, and extinction). The hypothesis suggests that the key to human beings' capability of solving a domain with little training data is the way how human beings organize the learned knowledge from domains. As bits of domains impinge on us, we human beings cluster the domains into several states based on domain similarity, so that the learning occurs within each cluster instead of across cluster boundaries. So, when a new domain enters, it can quickly take advantage of the knowledge learned within the group to which it belongs or initiate a new group if it is completely different from any existing groups, as illustrated in Figure 1.6.

In addition to grouping domains, knowledge sharing between these groups plays a crucial role in enabling the model to generalize well across a broad range of operational scenarios. As represented in Figure 1.7, each domain shares information with others, allowing new domains to benefit not only from internal knowledge but also from the insights gained by overlapping or adjacent domains. This process facilitates smoother transitions and faster learning when encountering new operational conditions, ensuring that the system adapts effectively to changing environments while maintaining a high level of accuracy.

Experiencing the similarity of the domain leads to positive and backward transfer. When the different domains that must be learned are related, it should be possible to exploit their similarity to achieve "positive transfer" between domains. Positive transfer means that due to learning one domain, the network also becomes better at another domain, either directly in terms of performance improvement, or indirectly by making (re)learning of that domain easier. For example, once a human has learned to play a first musical instrument (e.g., the piano), it is typically easier for them to master a second one (e.g., the violin). In CL, there are two types of transfer: forward transfer, whereby learning a new domain facilitates future domains; and backward transfer, whereby learning a new domain benefits previously learned

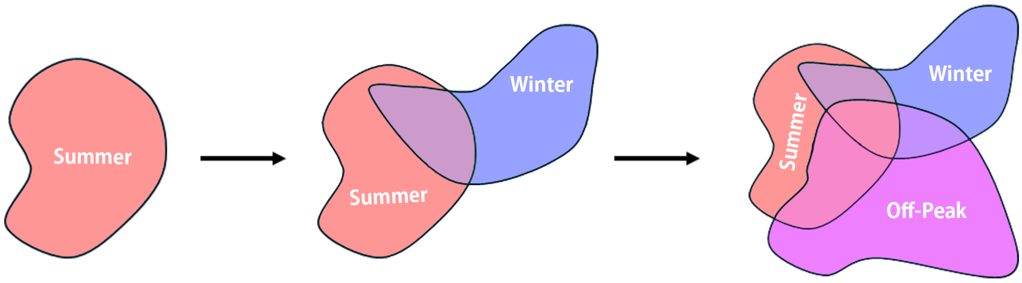


Figure 1.6: **Overlapping Operational Domains in Railway Domains**
Different operational domains evolve and overlap over time. The leftmost shape represents the summer time domain, characterized by high train speeds and traffic flow. The middle shape introduces the winter time domain, overlapping with the summer domain as conditions change but with slower train speeds and reduced traffic. The rightmost shape adds the off-peak domain, where trains operate at reduced speeds and capacity, showcasing how various operational and seasonal domains interact over time. continual learning adapts to these changing domains, maintaining predictive accuracy in railway maintenance systems.

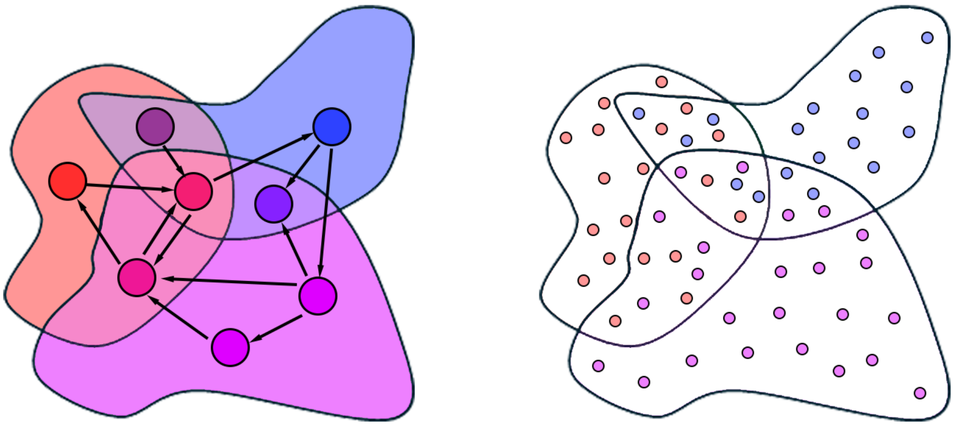


Figure 1.7: **Knowledge Sharing Between Domains and Train Distribution Across Domains in Continual Learning**
The left panel shows the knowledge sharing between different domains, with overlapping areas indicating shared characteristics between operational contexts. The right panel visualizes how individual trains (represented as points) operate across these domains. Some trains exist in all domains, while others are specific to one or two domains. This visual representation emphasizes how continual learning systems benefit from overlapping knowledge between different operational scenarios.

domains. Especially positive backward transfer has proven to be a challenging feature to achieve with artificial deep neural networks (DNNs) (Lopez-Paz and Ranzato 2017).

1.3 Contributions

The primary contribution of this thesis is going to be submitted to a peer-reviewed journal with the title: **Continual Learning for Railway Fault Prediction.**

1.4 Thesis Scope within the Project

This thesis is conducted within the scope of the PRODUTECH R3 project, an initiative aimed at empowering the production technologies sector (FTP) to take advantage of the opportunities brought by green and digital transitions. The core mission of the project is to reduce external technological dependence, enhance national added value, and drive structural changes in the Portuguese economy. To achieve this, PRODUTECH R3 brings together 112 companies and entities from key industrial sectors and the scientific and technological system, focusing on the development of 90 innovative products and services, showcased through 53 pilot projects.

In line with these objectives, this thesis explores PdM for railway fault diagnosis, a field that contributes to the digitalization of industrial processes and enhances the sustainability and efficiency of critical infrastructure. The work addresses the growing need for advanced diagnostic tools capable of improving the operational reliability of railway systems, which aligns with PRODUTECH R3's focus on innovation and structural transformation in production technologies.

By developing predictive models and fault detection mechanisms, this thesis not only supports the creation of smarter, more efficient railway systems, but also contributes to the wider goals of the PRODUTECH R3 project. These goals include promoting innovation ecosystems and boosting the capacity of the Portuguese industry to adapt to the demands of the future, particularly in sectors that rely on efficient and reliable technological solutions.

1.5 Data Protection, Security and Ethical Aspects

In this thesis, real-world data was used to develop and validate the proposed methodologies. The use of such data requires strict adherence to data protection regulations, security protocols, and ethical considerations to ensure the confidentiality, integrity, and appropriate use of the information.

All data handling procedures were conducted in accordance with applicable data protection laws and regulations, such as the General Data Protection Regulation (GDPR) of the European Union. Personal identifiable information (PII) was anonymized or pseudonymized to prevent identification of individuals.

The collaboration with CONSTRUCT-LESE was conducted transparently, with clear communication about data usage, research objectives, and expected results.

By implementing rigorous data protection and security measures and adhering to ethical research practices, this thesis ensures that the use of real-world data contributes positively to the field without compromising the rights or interests of any stakeholders involved.

1.6 Thesis Outline

The present thesis is divided into 6 chapters.

This Section 1 introduces the foundational topics of railways, the wheel-track interface, and maintenance practices that drive the research questions. It provides a detailed problem description, outlines contributions made through published works, and discusses ethical considerations pertinent to the study.

Chapter 2 presents a comprehensive literature review structured as a systematic literature review. It defines the research methodology and offers an overview of related work identified through the search results. The chapter delves into research questions, data sources, search terms, inclusion and exclusion criteria, and data collection queries. It also discusses the use of the PRISMA model and provides detailed answers to research questions.

Chapter 3 explains the process of simulating train-track dynamic interaction and focuses on the numerical modeling used to represent the monitoring systems on the wayside. This chapter provides an in-depth explanation of the simulation techniques and modeling approaches employed in the research.

Chapter 4 presents the methodologies and models used in this research. Theoretical foundations, formulas, and architectural designs that form the basis of proposed solutions are discussed. This chapter sets the groundwork for the experimental procedures and sets the stage for the results presented in the following chapter.

Chapter 5 presents and discusses the results of the proposed methodology for condition-based maintenance of wheel out-of-roundness. It analyzes the performance of the proposed CL model, compares it with an isolated model, and interprets the findings in the context of improving maintenance practices.

Finally, Chapter 6 summarizes the main conclusions drawn from the research. Reflects on the achievements and contributions of the thesis, addresses any limitations encountered, and proposes ideas and recommendations for future work in the field of railway maintenance and CL applications.

Chapter 2

Literature Review

This section reviews the current state-of-the-art in continual learning (CL) for fault diagnosis in railway systems, focusing on the use of deep neural networks (DNNs) in multi-domain environments. Following the guidelines of the Preferred Reporting Items for Systematic Reviews and Meta-Analysis (PRISMA) statement, this literature review provides a comprehensive overview of recent advancements in CL models, particularly those designed to handle varying operational conditions, such as different configurations of trains, speeds, and loads. The objective is to contextualize the development of models that adapt knowledge sharing across domains to improve performance in complex real-world scenarios.

A detailed examination of relevant methods is essential to address research challenges in CL and domain adaptation for fault diagnosis. This discussion highlights various approaches that enhance the performance of DNNs in multi-domain environments. Notable methods include techniques for managing catastrophic forgetting, optimizing knowledge transfer between domains, and handling diverse operational conditions such as different train speeds, loads, and track irregularities. These approaches form the foundation for the development of the CL framework aimed at improving fault diagnosis accuracy across varying real-world scenarios.

Throughout this chapter, we will have the research questions, reveal the queries driving this research, and outline the parameters for the systematic review of the literature, including inclusion and exclusion criteria. Following this, we will examine the research questions in depth, survey the relevant literature, and engage in a discussion that dissects the data and leads to the drawing of meaningful conclusions.

2.1 Systematic Literature Review

In this section, an exploration of research questions will be presented, providing a foundation for the current state-of-the-art. Then, an outline of the sources used in the gathering of the papers will be described. After that, the search terms, accompanied by the correspondent queries for research, will be detailed. Clear inclusion and exclusion criteria will be labeled. The section ends with an overview of the study, visualized through the PRISMA flow diagram.

2.1.1 Research Questions

For better understanding of the focus and the research, the research questions in Table 2.1 were posed:

Table 2.1: Specific Questions

ID	Question
RQ1	How can continual learning (CL) improve predictive maintenance (PdM) in railway systems by addressing the challenges of non-stationary environments?
RQ2	What are the most effective continual learning (CL) methods for minimizing catastrophic forgetting while maintaining model adaptability in fault diagnosis for industrial systems?
RQ3	How do differences in domain similarity and complexity affect the performance of continual learning (CL) models in railway predictive maintenance (PdM)?

The first research question explores how the CL models can be applied to PdM in railway systems, focusing on their ability to identify and diagnose faults in real-time. This investigates whether CL models can handle the dynamic nature of railway systems and consistently make accurate predictions as data evolve.

The second research question examines how CL models can adapt to non-stationary environments, specifically in the context of varying operational conditions such as changes in train load, speeds, and track irregularities to prevent catastrophic forgetting.

The third research question investigates the ability of the CL models to balance knowledge sharing between different domains, ensuring that the model not only adapts to new domains, but also maintains performance in previously learned domains.

These research questions were addressed through a comprehensive review of the literature and experimental work focused on CL for PdM in railway systems. Leveraging the identified research gaps, the following contributions were made:

- Models were developed and evaluated against state-of-the-art algorithms in railway PdM, focusing on the ability to diagnose faults under varying conditions, as identified in RQ1.
- Models were validated in non-stationary environments, where train loads, speeds, and track irregularities evolved between episodes, addressing the adaptability challenges described in RQ2.
- CL strategies were employed to maintain model performance across different operational domains while avoiding catastrophic forgetting, as discussed in RQ3.

2.1.2 Data Sources

The data sources from which the papers will be retrieved must be defined prior to the actual research. Table 2.2 shows the ones that were chosen, after a quick review of each one and their relevance in the research field.

2.1.3 Search Terms

To be able to search for the right papers and get the best results to be able to answer the research questions, the search terms must be specified in advance.

Table 2.2: Data Sources

Data source	Link
Science Direct	https://www.sciencedirect.com/
IEEE Xplore	https://ieeexplore.ieee.org/
Springer Link	https://link.springer.com/
Arxiv	https://arxiv.org/

After examining the areas relevant to this study, the most effective keywords were identified as shown in Table 2.3.

Table 2.3: Keywords

Field	Keywords
Railway Predictive Maintenance	Fault Diagnosis, Predictive Maintenance, Train-Track Dynamics, Wheel-Track Interaction, Railway Condition Monitoring, Railway Irregularities
Continual Learning	Continual Learning, Catastrophic Forgetting, Knowledge Retention, Knowledge Sharing in CL, Elastic Weight Consolidation, Synaptic Intelligence, Learning without Forgetting
Machine Learning for PdM	Deep Learning for Predictive Maintenance, Anomaly Detection, Time Series Data, Industrial Machine Learning, Convolutional Neural Networks for PdM

In order to search for potential related papers, some queries were created on the basis of the keywords previously defined. Table 2.4 shows the queries that were formulated for each of the data sources used.

2.1.4 Criterias

In the process of conducting a systematic literature review, the initial search for papers is based on keywords. However, the papers obtained may not all be relevant to the specific focus of the research. As a result, it is essential to establish inclusion and exclusion criteria to sift through the papers that are relevant to the research and discard those that do not meet or are not within the specified exclusion criteria. This guarantees that the review process remains concentrated and effective, directly supporting the research objectives.

In this study, the selection of literature adheres to carefully designed criteria to ensure relevance to the central theme of CL and PdM in railway systems. Every paper must be written in English and published after 2016 to reflect the latest advancements in the field. The scope of selected studies should address the adaptability of machine learning models, particularly in non-stationary environments such as railway systems, where operational conditions and anomalies vary over time. Papers that explore strategies to mitigate catastrophic forgetting and concept drift are prioritized, as these are critical challenges in CL and PdM applications.

Table 2.4: Queries

Data Source	Field	Queries
ScienceDirect	Railway Predictive Maintenance	("Predictive Maintenance" OR "Fault Diagnosis") AND ("Railway Systems" OR "Wheel-Track Interaction" OR "Railway Condition Monitoring")
	Continual Learning	("Continual Learning" OR "Catastrophic Forgetting" OR "Knowledge Sharing") AND ("Elastic Weight Consolidation" OR "Synaptic Intelligence" OR "Learning without Forgetting")
	Machine Learning for PdM	("Anomaly Detection" OR "Deep Learning for Predictive Maintenance") AND ("Time Series Data" OR "Convolutional Neural Networks")
IEEE Xplore	Railway Predictive Maintenance	("Predictive Maintenance" OR "Fault Diagnosis") AND ("Railway Systems" OR "Wheel-Track Interaction" OR "Railway Condition Monitoring")
	Continual Learning	("Continual Learning" OR "Catastrophic Forgetting" OR "Knowledge Sharing") AND ("Elastic Weight Consolidation" OR "Synaptic Intelligence" OR "Learning without Forgetting")
	Machine Learning for PdM	("Anomaly Detection" OR "Deep Learning for Predictive Maintenance") AND ("Time Series Data" OR "Convolutional Neural Networks")
SpringerLink	Railway Predictive Maintenance	("Predictive Maintenance" OR "Fault Diagnosis") AND ("Railway Systems" OR "Wheel-Track Interaction" OR "Railway Condition Monitoring")
	Continual Learning	("Continual Learning" OR "Catastrophic Forgetting" OR "Knowledge Sharing") AND ("Elastic Weight Consolidation" OR "Synaptic Intelligence" OR "Learning without Forgetting")
	Machine Learning for PdM	("Anomaly Detection" OR "Deep Learning for Predictive Maintenance") AND ("Time Series Data" OR "Convolutional Neural Networks")
Arxiv	Railway Predictive Maintenance	("Predictive Maintenance" OR "Fault Diagnosis") AND ("Railway Systems" OR "Wheel-Track Interaction" OR "Railway Condition Monitoring")
	Continual Learning	("Continual Learning" OR "Catastrophic Forgetting" OR "Knowledge Sharing") AND ("Elastic Weight Consolidation" OR "Synaptic Intelligence" OR "Learning without Forgetting")
	Machine Learning for PdM	("Anomaly Detection" OR "Deep Learning for Predictive Maintenance") AND ("Time Series Data" OR "Convolutional Neural Networks")

In addition, studies should focus on the impact of anomalies and irregularities in railway systems, including the effects of wheel and track imperfections on predictive models.

To improve clarity and structure, Tables 2.5 and 2.6 presented below outline the inclusion and exclusion criteria, respectively. These tables methodically establish the guidelines used to filter relevant studies, guaranteeing that research is focused on relevant literature that aligns with the goals established.

Table 2.5: Inclusion Criteria

ID	Criteria
IC1	The paper explores CL models with a focus on PdM or anomaly detection in dynamic environments.
IC2	The paper addresses the handling of real-time data streams and adaptive learning, particularly in railway systems or similar non-stationary environments.
IC3	The paper focuses on strategies to mitigate catastrophic forgetting in CL.
IC4	The paper discusses the effects of concept drift and variability on ML models, especially in the context of dynamic, evolving environments.
IC5	The paper includes research on the impact of anomalies or irregularities in predictive models for railway systems.
IC6	The paper is published in 2016 or later to ensure coverage of the most recent advancements in CL and PdM.
IC7	The paper offers practical case studies demonstrating the application of CL in real-world industrial settings, particularly in PdM for railways.

2.1.5 Data Collection

In order to thoroughly survey the scope of the literature, the use of keywords and queries was crucial. The definition of exclusion criteria was carefully carried out to limit the inclusion of papers that may not be relevant to current research.

After an exhaustive investigation, Table 2.7 was assembled, documenting the papers that were obtained and remained after the implementation of the pre-established exclusion criteria.

2.1.6 PRISMA Model

The current study will use the most recent version of the PRISMA model to highlight the importance of the papers obtained. The PRISMA flow chart, shown in Figure 2.1, offers a graphical illustration of the total of papers initially collected and the subset that satisfies inclusion standards.

Table 2.6: Exclusion Criteria

ID	**Criteria**
EC1	Paper not written in English.
EC2	Paper published before 2016.
EC3	Paper does not focus on CL, PdM, or related anomaly detection topics in dynamic environments.
EC4	Paper focuses on static datasets or environments, without consideration for evolving or non-stationary conditions.
EC5	Paper discusses CL or PdM without any practical application or real-world validation.
EC6	The paper was not published in a journal or conference.
EC7	Duplicated or redundant papers.

Table 2.7: Search results

Database	**Papers Retrieved**
ScienceDirect	28 papers
IEEE Xplore	24 papers
SpringerLink	14 papers
Arxiv	14 papers
Other	12 papers
Total	92 papers

An exhaustive search of the databases produced a total of 92 papers. Before the screening phase, a careful analysis verified the non-existence of duplicates. The screening phase required a prudent examination of titles and abstracts, resulting in the removal of 7 papers that were not related to the current topic. Following this, the eligibility phase entailed a detailed evaluation based on the pre-set inclusion and exclusion criteria, ending in the removal of an additional 14 papers. This procedure yielded a collection of 71 suitable papers. Each paper was subjected to a rigorous review to ensure that only those of direct relevance were included in this research.

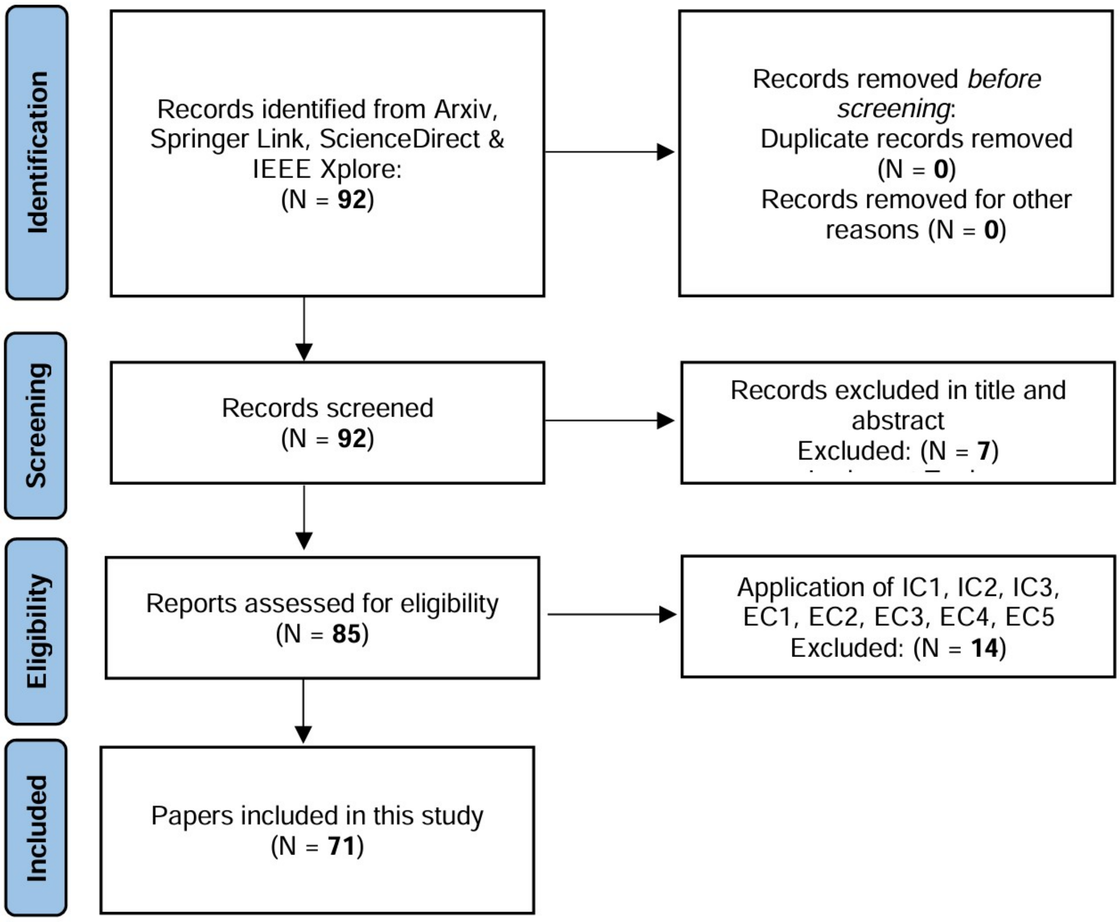


Figure 2.1: PRISMA Flow Diagram

2.2 Results

This section will focus on the results of the screening process that was represented in Section 2.1.6 by going through the defined research questions and answering them, referencing the specific papers that answer the questions.

RQ1: How can continual learning (CL) improve predictive maintenance (PdM) in railway systems by addressing the challenges of non-stationary environments? Continual learning (CL) has proven essential for predictive maintenance (PdM) systems to adapt to the ever changing non-stationary environments typical of real-world applications. Traditional deep learning (DL) models often struggle with generalization over time due to the static nature of their training data, but CL addresses this limitation by enabling models to continuously learn and adapt to new data while preserving previously acquired knowledge. This capability is particularly beneficial for PdM, where machinery and operational conditions evolve constantly (Hurtado et al. 2023). For example, CL methods such as elastic weight consolidation (EWC) have been used to predict faults in turbofan engines (Maschler, Vietz, et al. 2020), providing a distributed and cooperative learning framework that addresses challenges related to data privacy and decentralization. The effectiveness of EWC in preventing catastrophic forgetting while adapting to new data has also been shown to work well for fault prediction in lithium-ion batteries (Maschler, Tatiyosyan, and Weyrich 2022), helping to balance model stability and plasticity over time. Regularization-based CL strategies like

EWC, synaptic intelligence (SI), and learning without forgetting (LwF) are crucial to allowing models to adapt to new data without forgetting previously learned information, which is vital to maintaining high detection accuracy in dynamic industrial environments (Maschler, Pham, and Weyrich 2021). In railway systems, where time series data, such as acceleration signals or track irregularities, can vary significantly, models must be able to handle these changes effectively. It is important that dynamic models can learn and adapt over time due to the complexities involved in processing such non-stationary time series data in railway systems (Lourenço, Ribeiro, et al. 2024).

RQ2: What are the most effective continual learning (CL) methods for minimizing catastrophic forgetting while maintaining model adaptability in fault diagnosis for industrial systems? Catastrophic forgetting is a critical challenge in fault diagnosis for industrial systems, particularly in predictive maintenance (PdM) and continual learning (CL). An effective approach to address this is elastic weight consolidation (EWC), which helps the model to retain important parameters from previous tasks, preventing them from being overwritten by new ones. EWC has been widely applied in various industrial settings, such as turbofan engines and lithium-ion batteries, to maintain model performance over time by balancing stability and plasticity (Kirkpatrick et al. 2017). Another method, synaptic intelligence (SI), functions similarly by tracking the importance of model weights and penalizing significant changes, offering another robust way to handle catastrophic forgetting in dynamic environments (Zenke, Poole, and Ganguli 2017). In addition to regularization-based approaches, replay methods such as experience replay (ER) are commonly used to prevent forgetting by storing and replaying past examples during training. This is especially useful in industrial systems where CL is necessary, but storage restrictions can limit access to historical data (H. Shin et al. 2017). Replay mechanisms are often combined with regularization techniques like EWC to further enhance the adaptability of the model, offering a hybrid solution to maintain knowledge in incremental learning tasks (Rolnick et al. 2019). Another promising method is learning without forgetting (LwF), which avoids forgetting by using knowledge distillation to transfer knowledge from previous models to new ones. This is particularly beneficial in industrial settings, where models must continually integrate new fault types while retaining previously learned knowledge (Li and Hoiem 2018). This method shares similarities with ER in that it preserves historical knowledge while still allowing the model to adapt to new conditions. The literature also highlights the effectiveness of hybrid approaches that combine replay and regularization, such as applying both ER and EWC. These methods help balance the trade-off between maintaining past knowledge and learning new information, which is critical in complex industry systems such as railway systems, where operational conditions are constantly changing (Parisi et al. 2019). Whether through replay mechanisms or regularization, these approaches represent the most effective methods to minimize catastrophic forgetting while maintaining the adaptability needed for accurate fault diagnosis.

RQ3: How do differences in domain similarity and complexity affect the performance of continual learning (CL) models in railway predictive maintenance (PdM)? Differences in domain similarity and complexity significantly influence the performance of the CL models in PdM for railway systems. One of the main challenges lies in finding a shared solution that can generalize across all incremental domains, particularly when using techniques such as experience replay (ER) or regularization. The difficulty arises from severe inter-domain interference, which becomes more pronounced as the number of domains increases and their similarity decreases. In cases where domains differ substantially in terms of operational factors, such as train speed, load, or environmental conditions, the CL model must continuously adjust without compromising previously acquired knowledge. However, learning-invariant representations across diverse domains may inadvertently reduce adaptability, as the ability to tailor the model to specific tasks or conditions is constrained. This problem becomes increasingly difficult as the parameter space becomes more irregular, making it challenging to maintain both stability and plasticity. In fact, this issue has been shown to be NP-hard in general, which means that the model's ability to adapt to new domains while preserving performance on older domains is inherently complex and difficult to optimize (Bouvier 2021; Knoblauch, Husain, and Diethe 2020).

In addressing the research questions, we identified key findings and strategies that are crucial for advancing PdM in railway systems through CL. These are summarized in Table 2.8 for ease of reference.

Table 2.8: Search results

RQ	Findings	Strategies	Works
1	Adaptation to non-stationary environments; Importance of CL in PdM for railways; CL and knowledge preservation	Regularization-based; EWC; Dynamic models	Hurtado et al. 2023; Lourenço, Ribeiro, et al. 2024; Maschler, Pham, and Weyrich 2021; Maschler, Tatiyosyan, and Weyrich 2022; Maschler, Vietz, et al. 2020
2	Catastrophic forgetting; Minimize forgetting; Hybrid approaches	Regularization; Replay; Knowledge distillation; Hybrid approaches	Kirkpatrick et al. 2017; Li and Hoiem 2018; Parisi et al. 2019; Rolnick et al. 2019; H. Shin et al. 2017; Zenke, Poole, and Ganguli 2017
3	Domain similarity; Inter-domain interference; Stability and plasticity	Interference understanding; Optimization techniques; Customized models	Bouvier 2021; Knoblauch, Husain, and Diethe 2020

Chapter 3

Data

This section outlines the data collection process and details how the data was obtained and the simulations conducted to generate it. In collaboration with CONSTRUCT-LESE, we gained access to their data generator software, which was developed through their collaborative efforts. This software allowed for the simulation of various operational scenarios, allowing the generation of the necessary data for this study.

3.1 Numerical Modeling

The wheel out-of-roundness detection system incorporates an accelerometer and a strain gauge positioned along the track. To validate the proposed methodology, extensive 3D simulations based on the train-track interaction model were performed.

A virtual simulation comparing the baseline and damaged wheel scenarios was performed to validate the automatic out-of-roundness diagnosis proposed in this study. Once validated, this method can be applied to real data for various trains with different wheel defect conditions. In practical scenarios, wheels have minor imperfections, such as flats and polygonization, leading to significant variations in wheel–rail contact forces and vibrations in train and track components.

For wheel flats, two flat length intervals (L_w) were considered, designated as L1 and L2. The uniform distributions U (25, 50) mm and U (50,100) mm define the lower and upper limits of the flat length of the wheel for each interval L1 and L2, respectively. The wheel flat depth (D_w) is calculated based on the equation (Zhai et al. 2001): $D_w = \frac{L_w^2}{16R_w}$, in which R_w is the radius of the wheel. The vertical profile deviation of the wheel flat is defined as:

$$Z = -\frac{D_w}{2}\left(1 - \cos\frac{2\pi x_w}{L_w}\right) \cdot H\left(x_w - (2\pi R_w - L_w)\right), \quad 0 \leq x \leq 2\pi R, \tag{3.1}$$

where H represents the Heaviside function and x_w is the coordinate aligned with the longitudinal direction of the track. Figure 3.1 illustrates the effect of different severities of the wheels flat.

For wheel polygonization, the periodic irregularity of the radial tread around the circumference of the wheel was considered by varying the wavelengths (λ) as a function of harmonic order (θ) and wheel radius.

$$\lambda = \frac{2\pi R_w}{\theta}, \theta = 1, 2, 3 \cdots, n \tag{3.2}$$

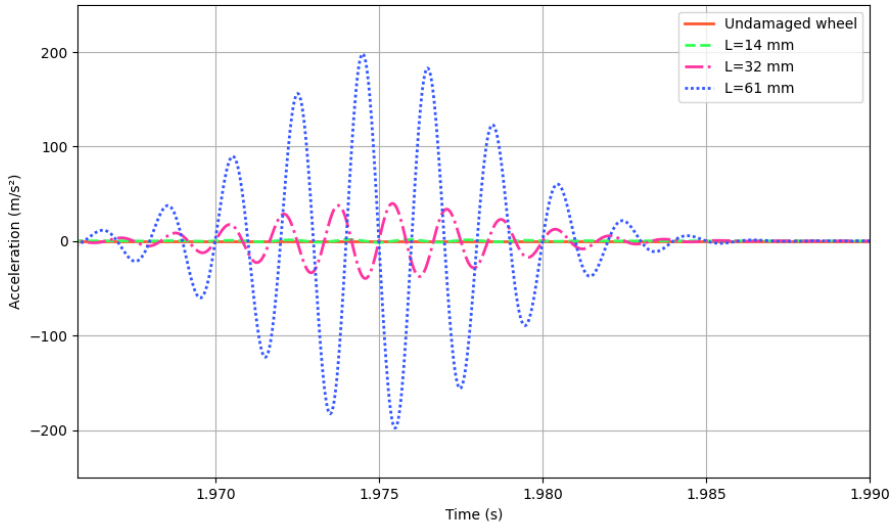


Figure 3.1: Acceleration time series for different wheel flat lengths

The selected wheel profiles were characterized by the wavelengths in the first 20 harmonics, with the sixth to eighth harmonic orders being dominant, and different irregularity wheel profiles generated based on the sum of sine functions ($H = 20$) as follows:

$$w(x_w) = \sum_{[03B8]=1}^{H} A_\theta \sin\left(\frac{2\pi}{\lambda}x_w + \varphi_\theta\right), \tag{3.3}$$

where A_θ is the amplitude of the sine function for each wavelength, which is calculated by the function:

$$A_\theta = \sqrt{2} \cdot 10^{\frac{L_W}{20}} \cdot w_{\mathrm{ref}}, \tag{3.4}$$

with $w_{ref} = 1\mu m$. The levels of wheel irregularity (L_w) were selected based on the irregularity spectrum in Figure 3.2, produced with the measurement values of four wheels with polygonal damage (Cai et al. 2019). Taking into account the phase angles to the sine functions that are uniformly and randomly distributed between 0 and 2π, several irregularity profiles of the wheel were generated to obtain different damage severities between 0.8 and 1.2 mm.

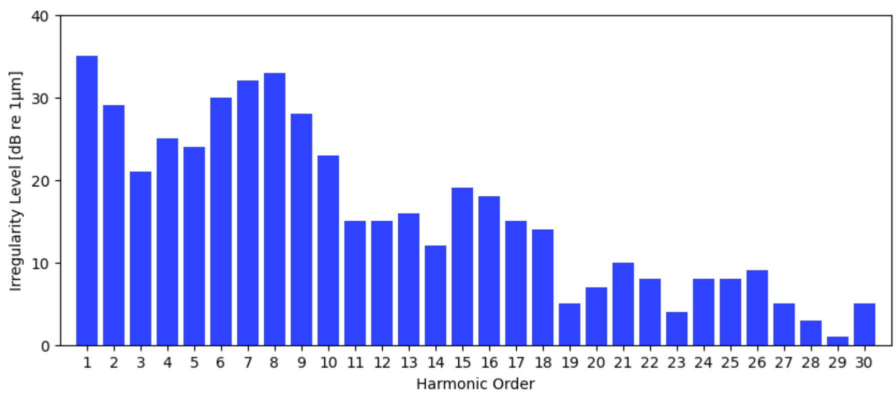


Figure 3.2: Wheel irregularity amplitude spectra (L_w) and harmonic order (θ)

3.2 Train-Track Dynamic Interaction

Train-track dynamic interactions were numerically investigated using VSI software, whose validation and detailed description are documented in previous work by the authors P. A. Montenegro et al. 2015. The VSI model integrates the train and the track through a three-dimensional wheel-rail contact model that uses Hertzian theory to calculate normal contact forces and the USETAB routine to determine the tangential forces arising from rolling friction creep (Hertz 1882; Kalker 1996). Designed in MATLAB®, the VSI software incorporates structural matrices from both the vehicle and the track, which have been independently modeled using finite element (FE) analysis.

In this study, the track was characterized using beam elements to represent the rails and sleepers. Spring-dashpot elements were employed to simulate the flexible layers' behavior, including the ballast and fasteners/pads, whereas mass point elements were utilized to account for the ballast's mass. The train model was developed in ANSYS® via a multibody framework. This involved using spring-dashpot elements to mimic the flexibility of both primary and secondary suspensions, rigid beams to represent the vehicle's rigid body motions, and mass point elements positioned at the center of gravity of each component, such as the carbody, bogies, and wheel sets, to capture their mass and inertial properties. Figure 3.3 provides a graphical representation of the numerical model.

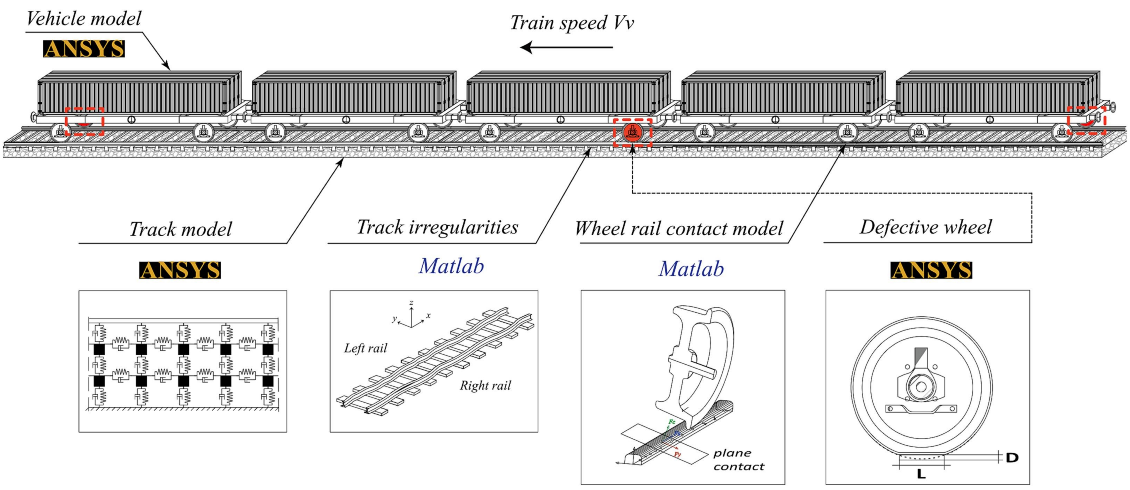


Figure 3.3: Train-track dynamic interaction

In real conditions, the rails also present small imperfections that can significantly affect the values of the wheel-rail contact force. In this context, 10 irregularity profiles, shown in Figure 3.4, were generated for wavelengths between 1m and 30m, covering the D1 wavelength interval defined by the EN 13848-2 standard (Standard 2003), with a sampling discretization of 1mm.

The amplitude of the irregularity profile was varied between -2mm and 2mm, for a total simulation length of 100m. These were based on real track irregularities of the Northern Line of the Portuguese Railway Network, measured with a track inspection vehicle EM120 every six months. It is important to note that the wavelengths used to generate these track irregularities are significantly longer than the wheel flat and polygonization. Thus, the exit frequencies due to the track are much lower than those of a defective wheel. More details on the generation of unevenness profiles can be found in the work of Mosleh, Costa, and Calçada 2020.

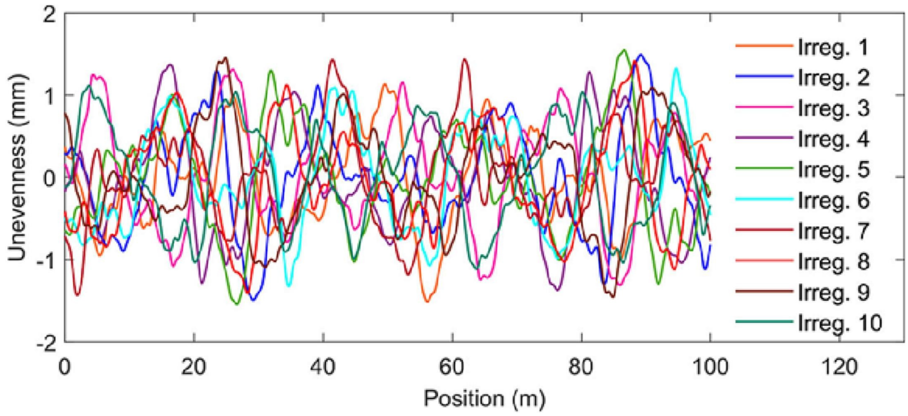


Figure 3.4: Irregularities rail profiles

3.3 Train Segmentation

After turning the signals into images, the data set is broken down into various domains. It is important to ensure that the signals from the domains have a very similar structure. This in turn allows the correct application of the model framework as described earlier. This, in other words, means that with proper organization of data and coherent domains, we can rightfully utilize the model framework for maintenance, have different models specializing in each of the varying types of damage trained for it, and hence improve the accuracy and ease out the process involved in fault diagnosis.

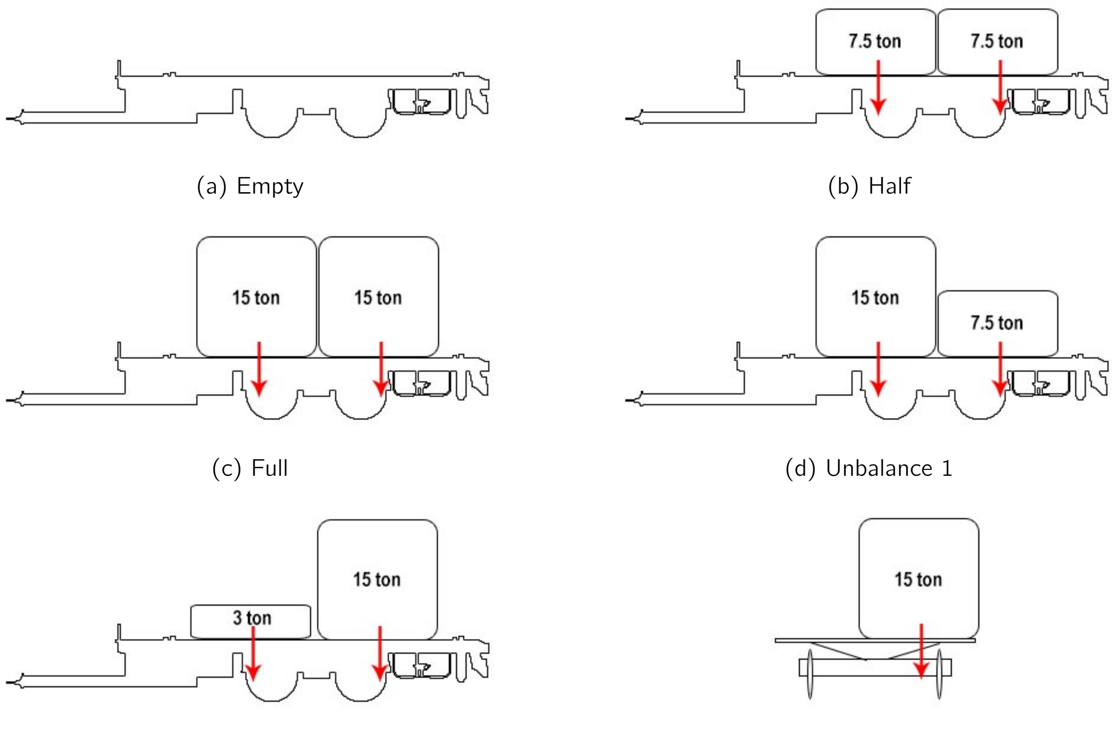
The analyses that exist on the train refer to the one with five carriages. In terms of baseline, we have 113 cases in which speeds, loads, and track irregularities vary. Figure 3.5 represents the train load schemes in the baseline domains. In terms of damage domains, all were performed with a fully loaded train with equal distribution, represented in Figure 3.5c. The data used in this work refer to the accelerations of the right and left sensors between sleepers. Tables 3.1 and 3.2 summarize the wheel domain in the Laagrss and Alfa Pendular trains, respectively.

Many implementations operate under the assumption that data are classified as either in-distribution (ID) or out-of-distribution (OOD). However, this assumption is often invalid in practical applications. In reality, data generally adhere to a natural distribution, and preserving this sequence is crucial, even if it does not consistently improve model performance.

To truly integrate aspects of CL into practical applications, the data must be carefully partitioned and identified. In this thesis, it is critical to ensure that the model can handle various data domains. Therefore, several domains were developed to reflect potential temporal evolutions that honor the natural-time sequence.

These divisions are seasonal and recurrent, designed to rigorously assess the model's ability to retain previous knowledge while managing new information. The domains are as follows and are also illustrated in Figure 3.6:

- **Domains 1 & 6**: Characterized by higher speed trains to minimize traffic disruptions during peak seasons.
- **Domains 2 & 7**: Features slower speed trains, operating in a less congested railway setting, typical of off-peak seasons.



(a) Empty (b) Half (c) Full (d) Unbalance 1 (e) Unbalance 2 (f) Unbalance 3

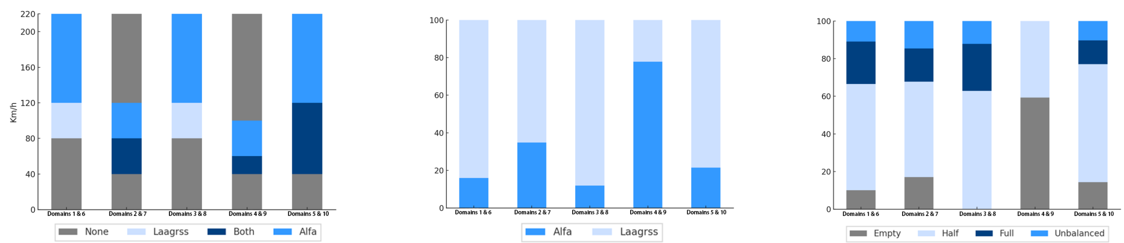
Figure 3.5: Representation of various train load schemes used in the study to evaluate the impact of different load distributions on wheel damage detection. The schemes include (a) an empty train, (b) a half-loaded train with equal distribution, (c) a fully loaded train with equal distribution, and (d-f) unbalanced load configurations.

- **Domains 3 & 8**: High flow of commerce and transport, characterized by high-speed trains with fully loaded wagons, representing summer time.
- **Domains 4 & 9**: The inverse of domains 3 and 8, featuring the slowest speeds and never completely filled wagons, indicative of a slow flow of transport and goods, akin to winter time.
- **Domains 5 & 10**: Trains operating at medium speeds, optimizing for balanced fuel efficiency and travel time. An example could be a period of regular maintenance operations where speed is moderated to ensure safety while maintaining a steady flow of traffic.

These carefully designed domains enable a thorough evaluation of the model's ability to address ongoing learning challenges in real-world environments, maintaining the integrity of the natural data sequence while effectively handling various operational conditions.

Table 3.1: Comparison of Baseline and Damage Domains on Laagrss Train

	Baseline Domains	Damage Domains	
		Flat Wheel	**Polygonized Wheel**
Vehicle	Laagrss	Laagrss	Laagrss
Train Load	6 (full, half, empty, 3 types of unbalanced)	1 (full)	1 (full)
Irregularity Profiles	4	1	1
Train Speeds	40-120 km/h	[60, 80, 100] km/h	[60, 80, 100] km/h
Defect Locations	-	1st wheel left (3rd wagon)	1st wheel right
Amplitude of the Defect (W)	-	L1: F10-20 mm, L2: F25-50 mm, L3: F55-100 mm	A1: 0.25-0.35 mm, A2: 0.65-0.75 mm
Order of Harmonics (H)	-	-	H1: H6-8, H2: H12-14, H3: H17-18
Total Number of Analysis	113*24	140	140



(a) (b) (c)

Figure 3.6: (a) Displays the distribution of train speeds (in km/h) across different train types (None, Laagrss, Both, Alfa) for each domain, illustrating variations in speed. (b) Shows the proportional usage of two train types, Laagrss and Alfa, in each domain, highlighting the dominance of each type per domain. (c) Represents the cargo load distribution across four states (Empty, Half, Full, Unbalanced) for each domain, showing the variations in load handling across different situations.

Table 3.2: Comparison of Baseline and Damage Domains on Alfa Pendular Train

	Baseline Domains	Damage Domains	
		Flat Wheel	**Polygonized Wheel**
Vehicle	Alfa Pendular	Alfa Pendular	Alfa Pendular
Train Load	3 (full, half, empty)	1 (full)	1 (full)
Irregularity Profiles	1-4	1	1
Train Speeds	40-220 km/h	[120, 200] km/h	[120, 200] km/h
Defect Locations	-	3rd wheel left (3rd wagon)	1st wheel right (1st wagon)
Amplitude of the Defect (W)	-	L1: 10-20 mm, L2: 25-35 mm, L3: 40-50 mm	A1: 0.25-0.35 mm, A2: 0.55-0.65 mm
Depth of Defect / Order of Harmonics (H)	-	D1: 0.02-0.06 mm, D2: 0.09-0.16 mm, D3: 0.23-0.36 mm	H1: H6-8, H2: H12-14, H3: H19-20, H4: H29-30
Total Number of Analysis	115*24	140	140

Chapter 4

Proposed Approach and Methodology

In this section, we outline the frameworks and methodologies used for fault diagnosis on railway wheels. It provides an overview of the methodology used for damage detection without delving into technical details.

We apply a model framework to design specific configurations and adapt to predictive maintenance (PdM) of the railway wheels. As illustrated in Figure 4.1, the process involves four key steps, data collection, signal processing, convolutional neural networks (CNNs) training with continual learning (CL) and anomaly detection. Inspired by the model zoo (Ramesh and Chaudhari 2022), the model is a specially designed ensemble learning paradigm for CL domains in which its capacity grows by adding smaller models for new domains. Hence, this framework is particularly well suited for our application, as it allows us to adapt to new data and changing patterns in wheel damage at any time.

4.1 Signal Processing and Image Conversion

The methodology involves placing sensors on the rail track to measure signals each time a train passes. These signals are traditionally analyzed in time series, and the raw data that we use are in that format. However, to take full advantage of the current deep learning (DL) techniques, these time series signals were converted into images. With this, we can use CNNs, which are the most widely used DL method in image and pattern recognition (Debayle, Hatami, and Gavet 2018).

Instead of using the image of the plot of the signal directly, we adopt the spectrogram Markov transition field (MTF) to encode the dynamics of the signal into a structured image. MTF captures the probability of state transitions over time, preserving both the temporal relationships and the underlying structure of the data. This transformation plays to the strengths of CNNs in detecting complex patterns within images (Zhao et al. 2022).

The MTF is calculated based on the transition matrix of quantized time series data. Consider the time series $X = \{x_1, x_2, \ldots, x_T\}$ which is quantized into Q discrete states. Each value x_i is then mapped to a corresponding state q_j. From this, we construct a matrix W of size $Q * Q$. The element w_{ij} in the matrix represents the probability that a value in state j is followed by a value in state i where $\sum_{j=1}^{Q} w_{ij} = 1$. This matrix W is called the Markov transition matrix. However, the matrix W is insufficiently dependent on both the distribution and the time series X, leading to a significant loss of information from the original time series, necessitating an improvement. The elements M_{ij} of the improved Markov transition matrix

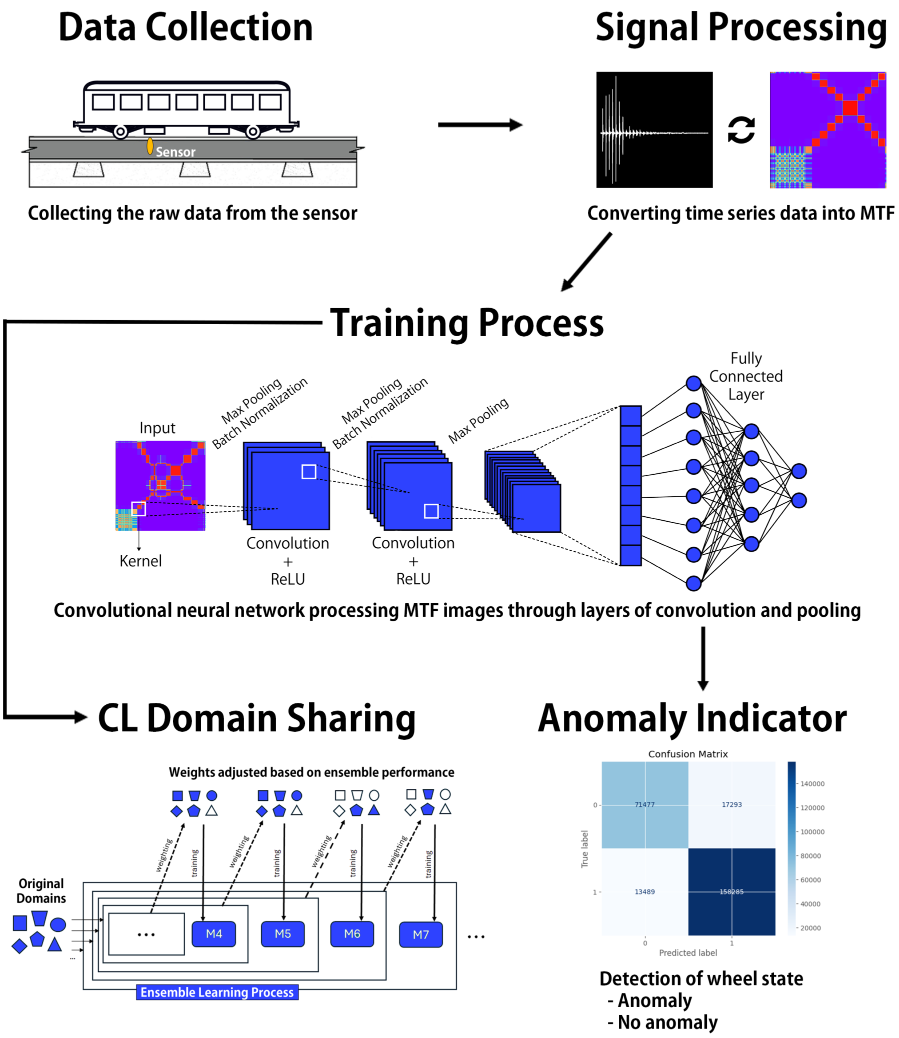


Figure 4.1: Flowchart of Railway Wheel Fault Diagnosis Methodology

M are defined as follows:

$$M_{ij} = \begin{bmatrix} w_{ij}|x_1 \in q_i, x_1 \in q_j & \cdots & w_{ij}|x_1 \in q_i, x_n \in q_j \\ w_{ij}|x_2 \in q_i, x_1 \in q_j & \cdots & w_{ij}|x_2 \in q_i, x_n \in q_j \\ \vdots & \ddots & \vdots \\ w_{ij}|x_n \in q_i, x_1 \in q_j & \cdots & w_{ij}|x_n \in q_i, x_n \in q_j \end{bmatrix}$$

To integrate temporal information into the matrix M, M_{ij} is used to represent the transition probability from state q_i to q_j. The matrix M is then constructed as:

$$M = \begin{bmatrix} M_{11} & \cdots & M_{1n} \\ M_{21} & \cdots & M_{2n} \\ \vdots & \ddots & \vdots \\ M_{n1} & \cdots & M_{nn} \end{bmatrix}$$

Here, $M_{ij}\|i-j\| = k$ denotes the transition matrix for points with a time interval k. Furthermore, in matrix M, the diagonal element M_{ii} is a special case where $k = 0$, representing the probability of staying in the same state at time i (Han et al. 2021; Yan, Kan, and Luo

2022).

Using MTF, we can enhance the CNNs performance by giving a more detailed representation of the signal to the model. As shown in Figure 4.2, the process involves first converting the time series signal into a Markov transition matrix, followed by applying CNNs for image recognition.

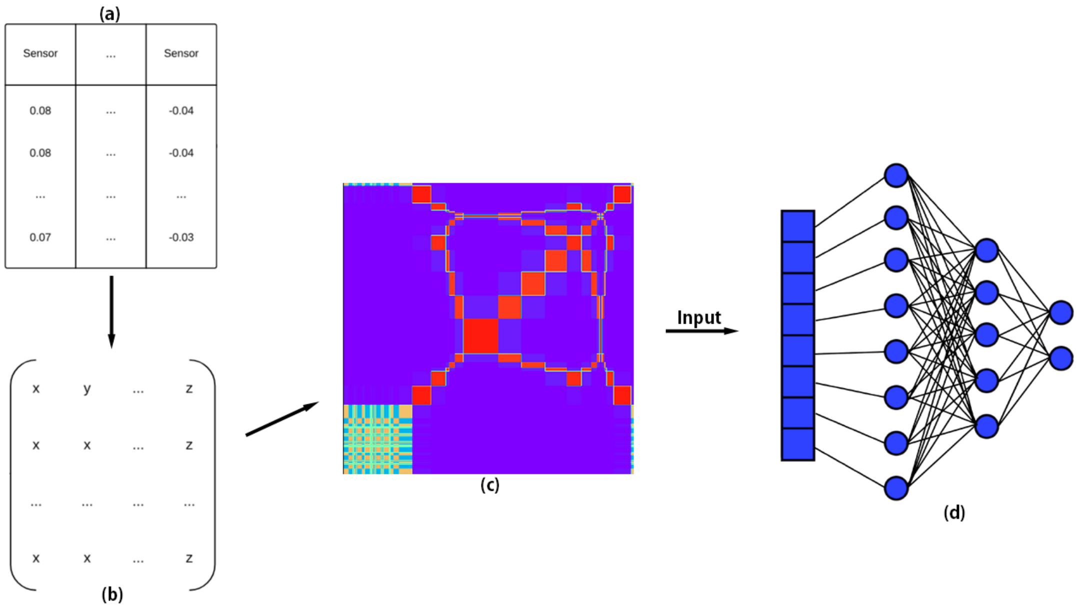


Figure 4.2: **Signal Processing with Markov transition field (MTF) for Anomaly Detection in Railway Wheels**
The process of converting time series signals into MTF and applying CNNs for detecting anomalies. (a) The initial time series signal is captured from sensors placed on the rail track. (b) This signal is then transformed into an MTF matrix, which encodes the temporal relationships between signal states. (c) The Markov transition matrix is converted into an image format. (d) The CNN processes this image, analyzing the temporal dynamics and identifying complex patterns indicative of wheel anomalies.

4.2 Model Architecture

This architecture is inspired by boosting techniques, where multiple "weak learners" are combined to create a stronger learner. Each CNN operates independently, however, they all contribute to the overall performance of the system, which will ensure robust handling of the various domains. In traditional boosting, the training weights for each instance in the next episode are adjusted based on the performance (weaknesses) of each individual model. However, in our approach, the weights for the next training episode are based on the performance of the entire ensemble up to that point, not just the individual CNNs. This difference allows the model to adapt its learning more effectively across multiple domains considering the collective knowledge of the ensemble.

Assume that $\overline{w}_{k,i} \in \mathbb{R}^n$ is a normalized vector of domain-specific weights. After episode k:

$$\overline{w}_{k,i} \propto \exp\left(-1/m \sum\nolimits_{(x,y)\in S_i} \log p_{k,i}(y|x)\right) \tag{4.1}$$

for each domain D_i with $i \leq k$; for $i > k, \overline{w}_{k,i} = 0$. In the next episode, domains $\overline{D}_{k+1}$ are drawn from a multinomial distribution with weights $\overline{w}_k$. With this, it makes it possible to put the domains with a lower empirical risk with a lower weight for the next boosting episode.

As illustrated in Figure 4.3, the entire ensemble is used to adjust the weighting of the domain in each subsequent training round. The image highlights how domains are weighted based on ensemble performance, rather than individual models' weaknesses, allowing for a more coordinated adaptation to challenging domains. This ensures that the system progressively concentrates on harder-to-classify domains by leveraging the collective knowledge of the ensemble. Similarly to how AdaBoost reduces the training error by progressively focusing on difficult samples (Schapire 2013), this model achieves a low empirical risk in all domains as additional CNNs are added, improving the overall robustness of the system.

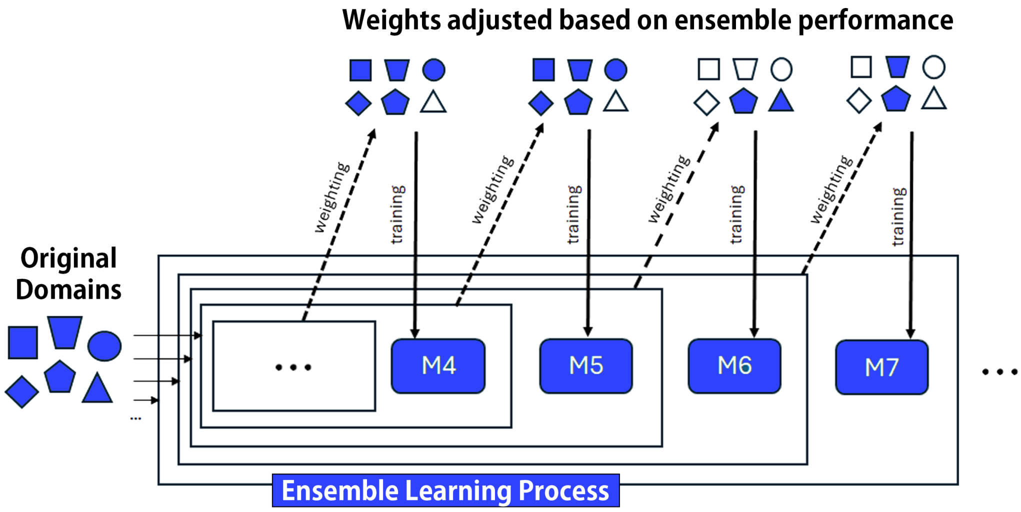


Figure 4.3: Boosting approach in the ensemble CNN architecture for domain learning

The reason why the model uses an ensemble of smaller CNNs is to be able to manage multiple domains without losing performance. Unlike traditional models that try to handle all domains with a single larger model, this model divides domains into several smaller CNNs. Each of those CNNs in the ensemble is trained in a specific domain or in a group of specific domains. The reason for this division is to help manage the complexity of learning with multiple domains by dividing the workload and to prevent the CL problem of catastrophic forgetting, which is when the model forgets what it learned with previous domains when learning in a new domain (Kirkpatrick et al. 2017).

The CNN ensemble in the model consists of a shared feature generator and domain-specific classifiers in each model. A significant contribution is the feature generator, which pulls out meaningful features from the input data, such as images in this case. The feature generator acts as a brilliant filter that finds and even enhances essential details of the input. For example, in analyzing the images of railway tracks, it may focus on the accelerometer that is more sensitive to damage, which is less affected by non-linear operational and environmental factors (Lourenço, Ferraz, Ribeiro, et al. 2023). The raw data are pre-processed by identifying features that serve as input to the domain-specific classifiers built over them. Such classifiers are layers that take the generated features and make predictions that are tailored to the domain. It is one such specialized component that makes its decisions or predictions about domains using the features that a feature generator works on. Doing so thereby centralizes the feature extraction process, making it a common one for all classifiers, thus being cost-effective in the use of resources, while at the same time allowing each classifier to specialize in its designated domain (Du et al. 2021; Tripuraneni, Jordan, and Jin 2020).

The model uses parameter isolation to make sure that the models do not interfere with each other's learning. The reason why the model makes use of parameter isolation is because with it, once the CNN is trained on a specific domain with its parameters, they are not updated when a new domain is given to the model. This approach makes sure that each CNN retains its learned knowledge without being affected by new given domains, tacking the issue of catastrophic forgetting. We can think of it as having a dedicated expert for each of the different domains who does not lose their expertise even when new experts are brought in. With the parameter isolation on each CNN, this model ensures that each of the domains past knowledge, which is essential for maintaining a good performance across all the domains.

The implementation of the model framework for the detection of railway wheel damage is illustrated in Figure 4.4. The first step will be to convert the time series sensor signals to images because that is the kind of input the entire framework takes. These images will then be categorized into different individual domains, whereby each of the domains will describe the situation for a specific scenario. The trained models will then encapsulate the inherent characteristics and patterns of the different damage classes. For every new damage data captured, there must be more training in the ensemble and the integration of added-in models, which will make the system responsive to damage of new and changed patterns. This approach, in turn, enables us to incrementally improve the performance of our PdM system back toward increased accuracy and robustness through timelier and more effective interventions in railway wheel maintenance.

4.3 Training Process

The model framework operates by training in a sequence of domains while sharing knowledge from previous domains. Let us assume that a sequence of domains $D_1, \dots, D_n$ is presented to the CL system, each sharing the same input X but different outputs $Y_1, \dots, Y_n$. In each episode k, the model is tasked with training in the current domain D_k and a selected subset of previous domains (to facilitate knowledge sharing). For example, during episode $k = 2$, the training involves a feature generator h and domain-specific classifiers, leading to the formation of the models $g_1 \circ h : X \to Y_1$ and $g_2 \circ h : X \to Y_2$. The model then classifies the inputs from both domains, producing a probability vector $p_{g_i \circ h}(y|x), \forall y \in Y_i$ based on the respective domain. Define the set of domains in episode k as $\overline{D}_k = \{D_{w_k^1}, \dots, D_{w_k^6}\}$, where $b \leq k$ serves as a hyper parameter, and $\omega_k^i \in \{1, \dots, k\}$. Training on $\overline{D}_k$ involves using a feature generator h_k and domain-specific classifiers $g_{(k,\omega_k^i)}$ for each chosen domain. These models collectively form our current model. After k episodes, the ability to predict data from D_i for $i \leq k$ is derived from averaging class probabilities output by all models that were applied to that domain, as indicated by:

$$p_{k,i}(y|x) \propto \sum_{l=1}^{k} 1_{\{P_i \in \overline{P}_l\}} g_{l,i} \circ h_l(x) \tag{4.2}$$

This formulation is also applicable during testing.

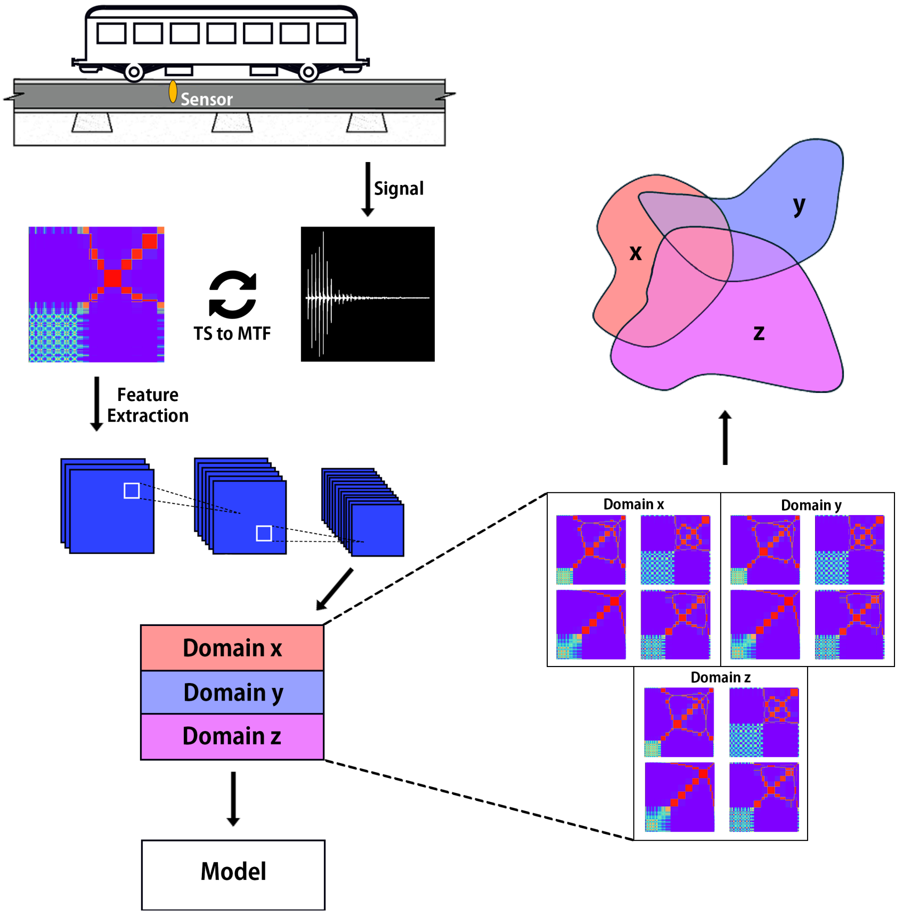


Figure 4.4: **Overview of the model framework for CL into railway PdM.** It transforms the signals into images and divides into domains where each domain represents a group of a specific damage, trains the domains in episodes with knowledge sharing between domains with CL and predicts the possible damage

The base model in the model is trained on the data for the first domain according to standard ML procedures. This is a very foundational training that sets the stage upon which the other models in the ensemble build. The carefully prepared training parameters, such as the learning rate that is responsible for the speed of updating knowledge, the batch size corresponding to a number of data points that the model observes before each update, and the settings for the optimizer methods to change the learning process and guarantee optimal training.

New models are trained sequentially, each new model incorporating knowledge from the highest-error domains in the previous episode. After each domain is learned, the system introduces a new model that focuses on the domains with the greatest need for improvement, as identified by their error rates. In each episode, the model incrementally learns a maximum of five domains, prioritizing those with the highest error percentages. This ensures continuous improvement while maintaining a comprehensive understanding of both newly introduced and previously learned domains. The sequential training process is illustrated in Figure 4.5.

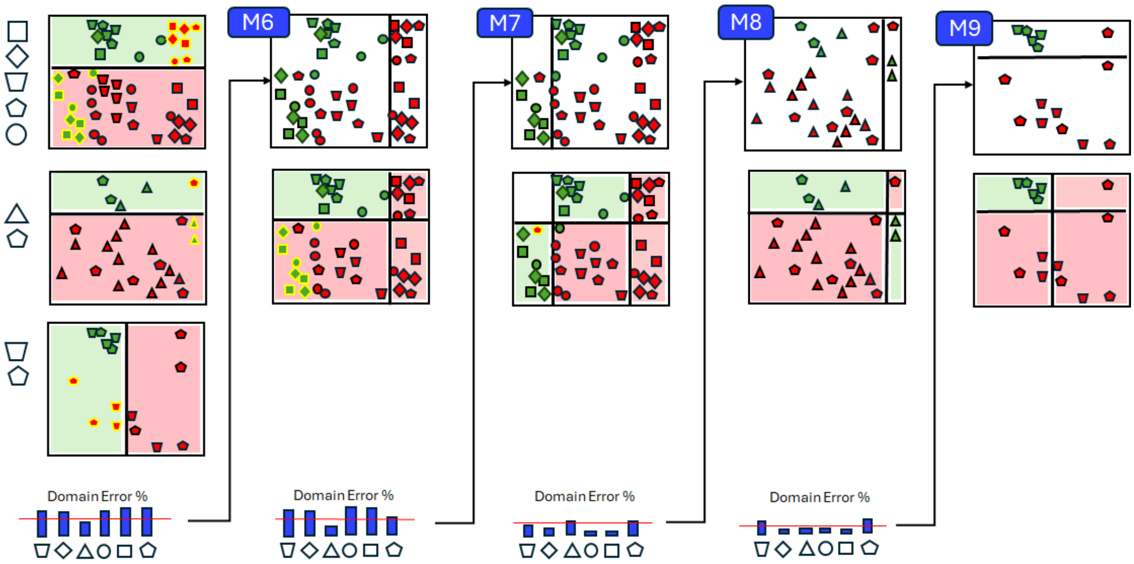


Figure 4.5: Domain selection and error-driven retraining process across episodes. At each step, domains are evaluated based on their error percentages (shown in the bar graphs below each plot), with domains exceeding the error threshold (red line) prioritized for retraining in the subsequent episode

4.4 Experimental Setup

The model consists of two convolutional layers. These layers are responsible for scanning the input data (which in this case are images that were converted from sensor time series data) and detecting important features related to a potential anomaly. Each of these convolutional layers applies a set of filters, which can be thought of as small sliding windows that move over the image to detect different patterns (Krizhevsky, Sutskever, and Geoffrey E. Hinton 2017).

The first convolutional layer uses 80 filters with a kernel size of 3x3 pixels, meaning it applies 80 different filters to the image, each looking for a unique type of feature. The second also applies 80 filters with the same kernel size, but that layer focuses on detecting more abstract patterns, building on the features identified in the earlier layers. By having multiple layers, the network can detect not just simple edges or textures but also more complex patterns that correspond to specific types of wheel damage (Zeiler and Fergus 2014). Figure 4.6 illustrates the architecture of the CNN model, showing the flow of data from the input layer through the convolutional layer, the max pooling, and the fully connected layers for classification.

To ensure that the learning process is stable and effective, the model employs batch normalization after the second convolutional layer. Batch normalization is a technique that helps speed up training by normalizing the output of these layers, preventing the model from becoming unstable during learning (Ioffe and Szegedy 2015).

Following each convolutional layer, the model applies a max pooling operation. Max pooling is a down-sampling strategy that minimizes the data's dimensionality by preserving only important features. This procedure is essential because it alleviates the computational burden on the system, thereby enhancing the model's efficiency without compromising vital information regarding potential damage. For example, after recognizing a crack in the wheel within

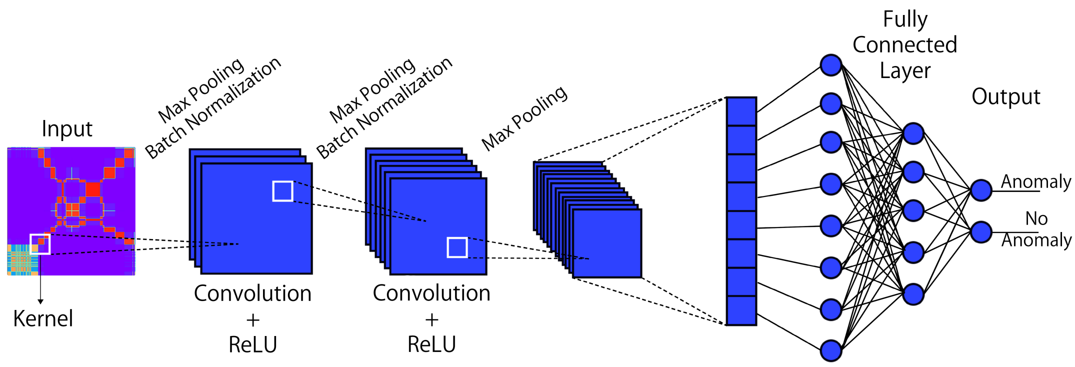


Figure 4.6: **Convolutional neural network (CNN) Architecture for Railway Fault Diagnosis**.
This diagram illustrates the flow of data through the CNN, starting from the input layer through the three convolutional layers with Max Pooling and Batch Normalization operations, and ending with the fully connected layer for classification of railway faults.

a specific image region, max pooling guarantees the retention of this critical information while discarding superfluous data (Rumelhart, G. E. Hinton, and Williams 2013). The layer information is summarized in Table 4.1.

Table 4.1: Summarization of the Layer Information

Layer	**Filters**	**Kernel Size**	**Activation Function**	**Additional Operations**
Convolution Layer 1	80	3x3	ReLU	MaxPooling, Batch Normalization
Convolution Layer 2	80	3x3	ReLU	MaxPooling, Batch Normalization
Fully Connected Layer	-	-	Softmax	-

The output of the convolutional and pooling layers is subsequently fed into a fully connected layer, which executes the terminal classification. This layer utilizes the abstract features extracted by the preceding layers to predict the presence of an anomaly. In this experiment, the fully connected layer is structured to manage multiple domains, being trained to classify various railway faults into predefined scenarios.

In addition, the architecture integrates a restricted data replay mechanism, revisiting a small fraction of data previously encountered during training sessions. This technique mitigates the risk of the model "forgetting" earlier domains while adapting to new ones. The comprehensive design and training strategy ensure that the CNN remains effective and scalable, capable of incorporating new domains without necessitating extensive retraining. This approach enhances the system's capacity as new domains are added, making it especially advantageous for railway maintenance applications.

Chapter 5

Fault Diagnosis

In this section, we present the analysis of the performance of our fault diagnosis model in various domains. The results demonstrate the impact of continual learning (CL) and hyperparameter tuning on the accuracy and robustness of the model. We also evaluate the model's ability to handle different train-track interactions, loads, and speeds, while highlighting three common learning metrics such as forward transfer, backward transfer, and domain-specific performance.

5.1 Evaluation Metrics

In this section, we will present the formulas and descriptions of each metric, the average domain accuracy, the learning accuracy (forward transfer), and the forgetting measure (backward transfer) used throughout the experimentation. These metrics provide insights into how well the model performs in different domains, how efficiently it adapts to new tasks, and how much knowledge is retained when learning new information.

The average domain accuracy (ACC) evaluates the overall performance of the CL model in all domains. Higher values indicate better final performance across all domains. It is calculated as:

$$\mathrm{ACC} = \frac{1}{D}\sum_{i=1}^{D} a_{D,i}, \tag{5.1}$$

where $a_{D,i}$ represents the accuracy on the i^{th} domain after training all the D domains, and D is the total number of domains.

The learning accuracy (LA), also known as forward transfer, evaluates the model's ability to learn new domains by using prior knowledge. Higher values indicate better learning transfer across domains. It is calculated as:

$$\mathrm{LA} = \frac{1}{D}\sum_{i=1}^{D} a_{i,i}, \tag{5.2}$$

where $a_{i,i}$ represents the accuracy on the i^{th} domain immediately after training in the i^{th} domain.

The forgetting measure (FM), also known as backward transfer, measures how much the model has "forgotten" previous domains after learning new domains. Lower values are better because they indicate that the model retains more knowledge from previous domains. It is calculated as:

$$\text{FM} = \frac{1}{D-1} \sum_{i=1}^{D-1} \max_{l \in \{1,\ldots,D-1\}} (a_{l,i} - a_{D,i}) \tag{5.3}$$

where $a_{l,i}$ represents the accuracy on the i^{th} domain after learning the l^{th} domain, and $a_{D,i}$ represents the accuracy on the i^{th} domain after learning all domains.

5.2 Continual Learning and Isolated Models

The CL model and the isolated model differentiate in how they handle knowledge in different domains. The CL model is designed to continuously learn from new domains while retaining knowledge from previous ones, allowing knowledge sharing between tasks. This enables the CL model to adapt and improve its performance over time, even in later domains. In contrast, the isolated model treats each domain independently, without leveraging any prior knowledge. As a result, the isolated model does not benefit from accumulated learning across tasks, which often limits its ability to adapt and perform well in dynamic, non-stationary environments.

The evolution of the accuracy of the domain in episodes, comparing the CL model with the isolated model, is shown in Figure 5.1. The results highlight the ability of the CL model to improve its accuracy over time, even in domains where the initial performance may have been lower. Although the isolated model, represented by crosses, occasionally outperforms the CL model in earlier episodes, the CL model demonstrates an adaptive capacity to improve, eventually surpassing the isolated model as the episodes progress.

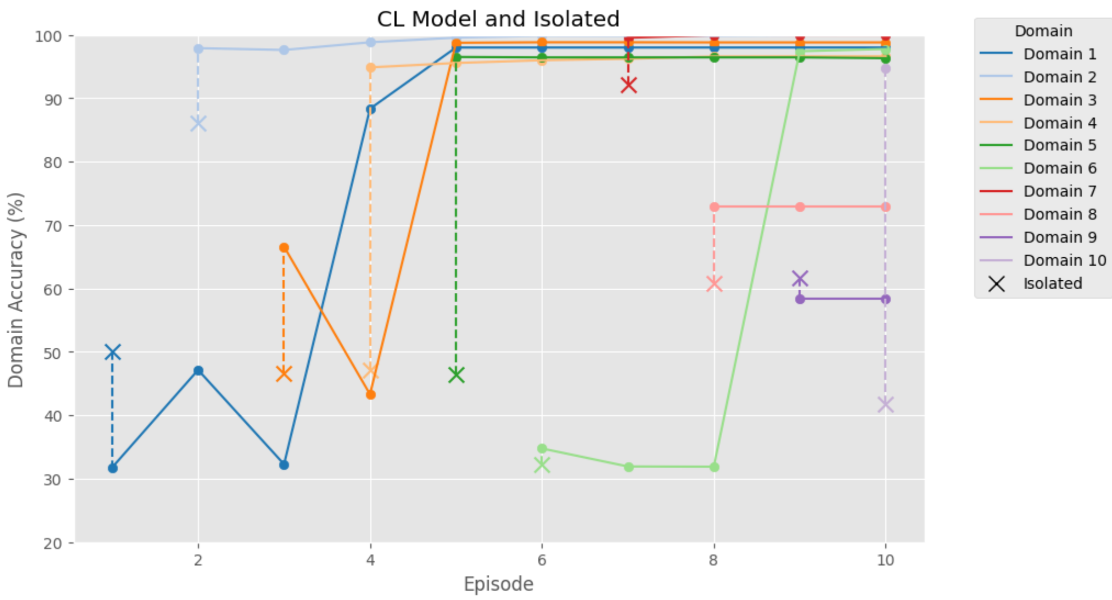


Figure 5.1: Evolution of Domain Accuracy

This gradual improvement highlights the strength of the CL model in learning from multiple domains and using knowledge sharing, even when some domains initially pose greater challenges. Over time, the CL model consistently outperforms the isolated model, showcasing its advantage in handling multiple domains and continuous learning.

5.3 Model Evaluation and Discussion

To further evaluate the model's robustness, we ran the experiment 10 times, varying hyper-parameters and layer configurations across runs, with each run trained for only one epoch. For each domain, we calculated the mean accuracy and standard deviation and used these to compute 95% confidence intervals, providing a measure of the variability in the model performance. The mean accuracy μ_A is simply the average of the accuracy of all runs, which are represented by $A_1, A_2, ..., A_{10}$, and is given by $\mu_A = \frac{1}{n}\sum_{i=1}^{n} A_i$, where A_i is the accuracy for the i^{th} run and n is the total number of runs, which is 10 in our case. The standard deviation σ_A of the accuracy scores measures the amount of variation dispersion of the accuracy between runs, which is given by $\sigma_A = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(A_i - \mu_A)^2}$, where A_i is the accuracy of the i^{th} run, μ_A is the mean accuracy (as calculated above) and n is the number of runs. Figure 5.2 shows the mean accuracies with error bars representing the confidence intervals, highlighting the stability and reliability of the model predictions across different domains.

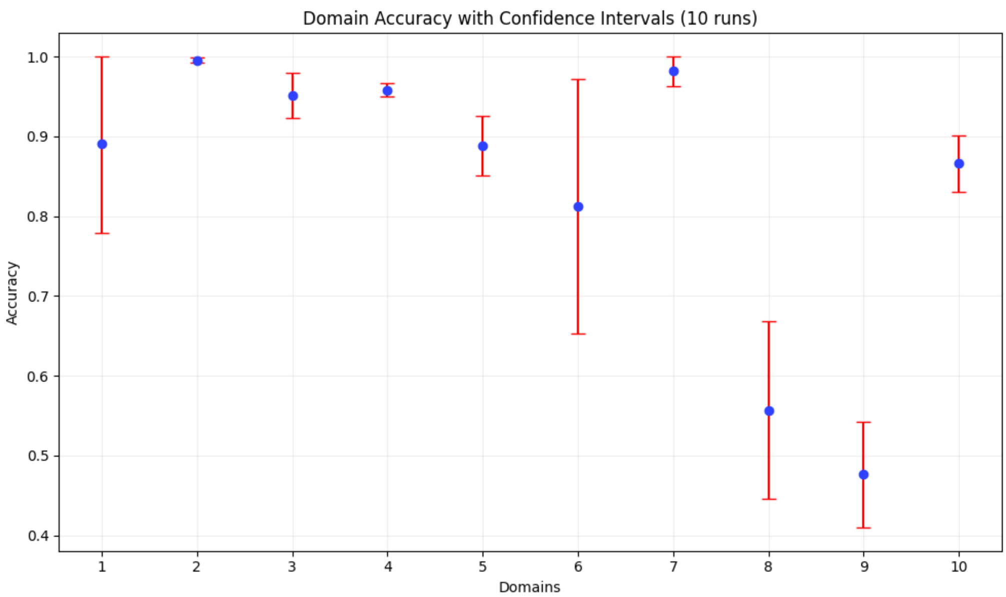


Figure 5.2: Domain accuracy with 95% confidence intervals based on 10 runs. Each point represents the mean accuracy for a specific domain, with red error bars indicating the 95% confidence interval.

Although the model performed robustly overall, domains 8 and 9 consistently showed the lowest accuracy and showed no improvement over episodes in all runs. In particular, domains 3 and 8, and domains 4 and 9, share identical scenarios. The underperformance in domains 8 and 9 can be ascribed to several factors, one of them being that these domains are evaluated later in the training, potentially suffering from task interference and catastrophic forgetting. Moreover, despite operational similarities with domains 3 and 4, domains 8 and 9 have greater noise and variability, making generalization difficult. The short training duration (one epoch per run) also hinders the model adaptation to these challenging scenarios. These results underscore the limitations of the model in later episodes, where increased complexity and data variability may hinder performance.

In addition to the accuracy comparisons between episodes, some key metrics were employed to quantify the performance of the CL model versus the isolated model. These metrics include the average domain accuracy, reflecting the overall accuracy across all domains, forward transfer (learning accuracy), assessing the model's ability to apply past knowledge to new tasks, and backward transfer (forgetting), evaluating the retention of previously learned knowledge after new domain learning. Table 5.1 summarizes these metrics, underscoring the CL model's benefits over the isolated model.

Table 5.1: Metrics for CL Model vs. Isolated Model

Metric	CL Model	Isolated
Average Domain Accuracy	0.913659	0.565098
Learning Accuracy (Forward Transfer)	0.748197	0.565098
Forgetting (Backward Transfer)	0.000241064	0

The CL model considerably surpasses the isolated model in average domain accuracy, achieving 0.91 versus 0.57. This indicates the CL model's capability of using prior domain knowledge to improve overall performance. The forward transfer score further underscores this benefit, with the CL model scoring 0.75, suggesting superior knowledge transfer. Moreover, the CL model exhibits minimal forgetting (0.00024), highlighting its ability to retain learned information, unlike the isolated model, which shows no backward transfer.

To complement overall accuracy, we computed domain-specific performance metrics such as precision, recall, and F1-score. Precision assesses the model's accuracy in identifying true positives among all predicted positives, recall measures its success in detecting all true positives, and the F1-score is the harmonic mean of precision and recall, providing a balanced performance metric. Table 5.2 presents these metrics for each domain.

Table 5.2: Performance Metrics for Each Domain

Domain	Precision	Recall	F1-Score
1	**1.00**	**1.00**	**1.00**
2	**1.00**	**1.00**	**1.00**
3	0.98	0.99	0.98
4	0.99	0.94	0.96
5	0.91	0.99	0.95
6	0.87	1.00	0.93
7	**1.00**	**1.00**	**1.00**
8	0.79	0.84	0.81
9	0.62	0.98	0.76
10	0.96	0.81	0.88

The results indicate a near-perfect performance of the model in domains 1, 2, and 7 with F1-scores of 1.00, which means optimal fault detection. In contrast, domains 8 and 9 show the lowest performance, with F1-scores of 0.81 and 0.76, aligning with previously noted lower accuracy and higher variability. These metrics highlight performance variability across different operational scenarios, especially under challenging conditions.

The performance of the anomaly detection model is evaluated using a confusion matrix, summarizing the model's predictive results across all domains. This matrix highlights the model's proficiency in distinguishing normal operations (no anomaly) from fault conditions (anomaly). Table 5.3 presents the confusion matrix, with true labels in rows and predicted labels in columns.

Table 5.3: Confusion Matrix for Anomaly Detection

	No Anomaly	**Anomaly**
No Anomaly	77075	11695
Anomaly	18205	153569

The results show that the model accurately detected 77075 instances of normal operations and 153569 anomalies. However, it incorrectly classified 11695 normal instances as anomalies (false positives) and 18205 anomalies as normal (false negatives). These results indicate that, while the model performs well overall in detecting anomalies, there is still room for improvement in reducing the number of false positives and false negatives.

5.4 Hyperparameter Tuning

For hyperparameter tuning in the 10 runs, we varied key parameters to examine their effects on the performance of the model. We specifically altered the learning rate (LR), the momentum, the weight decay, the number of convolutional layers, and the number of batch normalization layers. The learning rate controls the size of the optimization step, while momentum integrates prior gradients for smoother updates. Weight decay was used for regularization to reduce overfitting. Changes in the number of convolutional and batch normalization layers affected feature extraction capacity and training stability. The hyperparameters for each run are detailed in Table 5.4.

Table 5.4: Hyperparameters Used Across 10 Experimental Runs

Run	Learning Rate	Momentum	Weight Decay	Conv Layers	BN Layers
1	0.01	0.9	0.00001	3	2
2	**0.005**	**0.85**	**0.00005**	**2**	**1**
3	0.02	0.92	0.0001	4	3
4	0.015	0.88	0.00001	3	2
5	0.01	0.95	0.00002	5	4
6	0.008	0.87	0.00003	3	2
7	0.025	0.9	0.00005	4	3
8	0.012	0.93	0.00002	2	2
9	0.007	0.89	0.00001	4	3
10	0.001	0.91	0.00004	3	2

The results discussed earlier in this section are derived from run 2, which is the best performing run among the 10 runs. However, it is important to note that in runs 5 and 6, domain 6 shows a notable drop in accuracy. This may indicate that certain hyperparameter settings struggle with specific domain characteristics, highlighting the need for careful tuning to ensure robust performance across all domains. The combination of hyperparameters in run 2 allowed for fast convergence and minimal overfitting, balancing both learning rate and regularization. With a learning rate of 0.005, momentum of 0.85, and a relatively simpler model architecture with 2 convolutional layers and 1 batch normalization layer, this configuration optimized the learning process across all domains. With these hyperparameter settings, run 2 achieved the highest domain accuracy and F1-scores across most domains, demonstrating its effectiveness in domain-adaptive learning.

To illustrate the effect of hyperparameter tuning on model performance, Figure 5.3 displays the accuracy of each run across different domains, while Figure 5.4 highlights the performance of run 2 compared to the mean accuracy across all 10 runs. Run 2 consistently achieved the highest performance, demonstrating the importance of tuning key parameters such as the learning rate and the number of layers. In particular, hyperparameter adjustments allowed run 2 to outperform the mean accuracy across all domains, as shown by the blue line exceeding the red mean accuracy line.

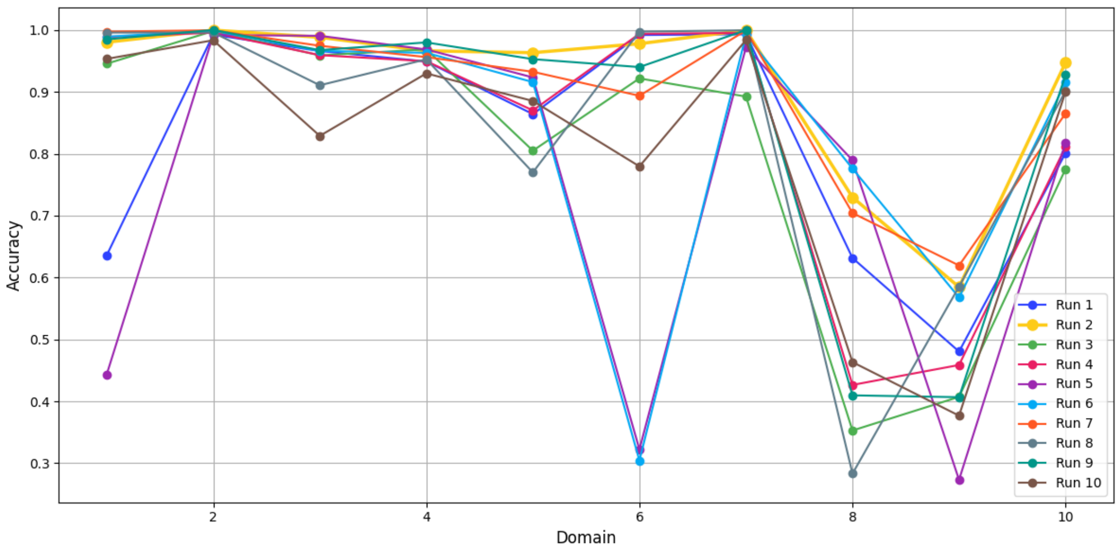


Figure 5.3: Performance across runs and domains. Run 2 is highlighted for its superior results.

The shaded region in Figure 5.4 represents the 95% confidence interval, showcasing variability across runs, while run 2 stays consistently within the upper bounds in all domains. This demonstrates not only the stability of run 2's configuration, but also its ability to generalize better in the presence of noise and complexity, especially in challenging domains like 8 and 9. These results underscore that hyperparameter tuning is critical for maximizing performance in fault diagnosis tasks, particularly when dealing with noisy and complex domains.

5.5 Performance Analysis

To further assess the statistical significance of the performance differences across the domains, we applied the Friedman test, a nonparametric test designed to detect differences in performance when the same models are evaluated under different conditions. The result, with a p-value of 0.00092, indicates that there are statistically significant differences between domains. A p-value below 0.05, as in our case, suggests that the variation in performance is not random, but related to differences in model learning between domains.

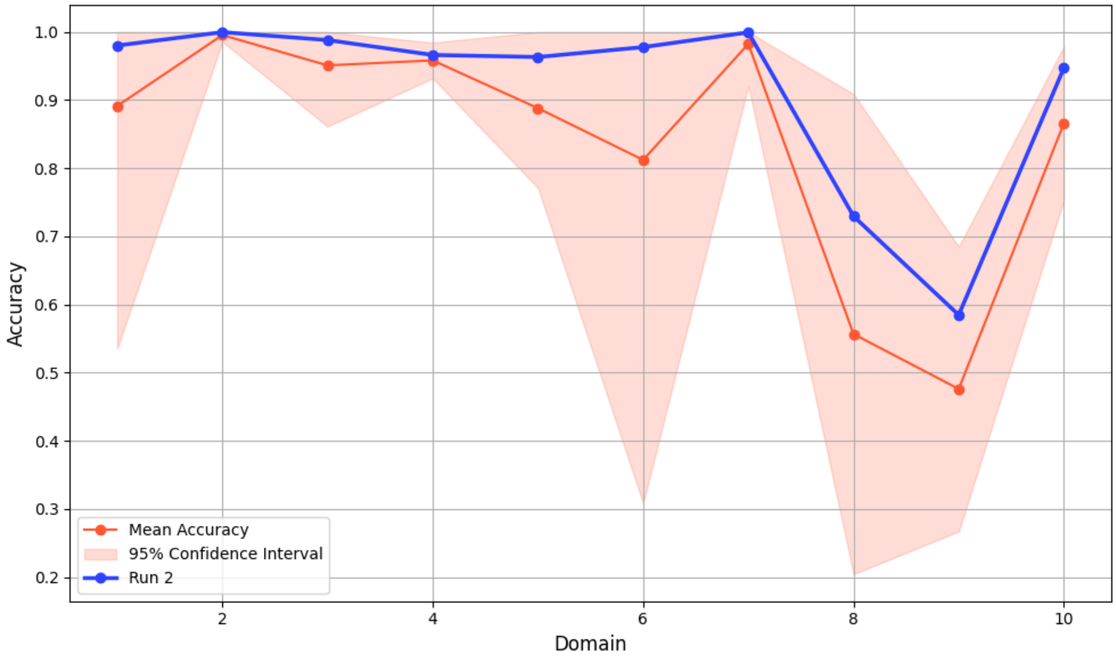


Figure 5.4: Performance of run 2 compared to the mean performance.

Following the Friedman test, we performed the Nemenyi post-hoc test, which allows pairwise comparison between the domains to pinpoint where these significant differences exist. The Nemenyi test compares each domain with the others, and the resulting p-values are displayed in the heatmap illustrated in Figure 5.5. Cells in the heatmap with values closer to 0 indicate that the difference in performance between those specific domains is statistically significant, while values closer to 1 signify that there is no significant difference between those domains. For example, domain pairs such as (1, 9) and (2, 1) show significant differences with p-values close to 0, highlighting substantial variation in performance between these domain pairs.

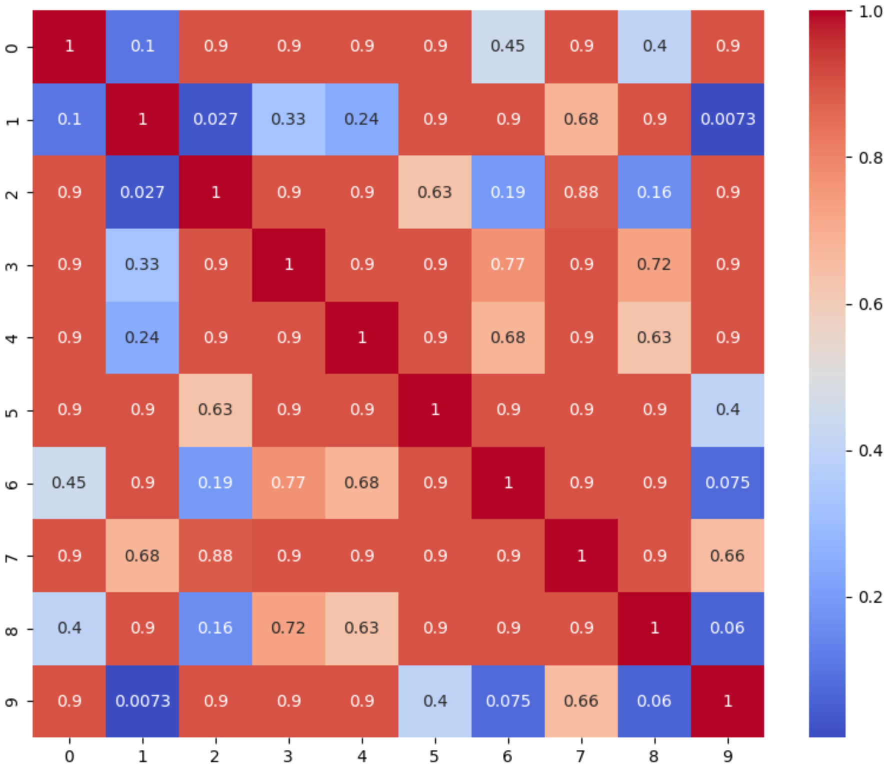


Figure 5.5: Nemenyi Post-Hoc Test Results

Chapter 6

Conclusions

The primary objective of this thesis was to explore the application of continual learning (CL) to predictive maintenance (PdM) in railway systems, focusing on non-stationary environments. This investigation aimed to address the key challenges in improving fault diagnosis accuracy under changing operational conditions, such as changing train speeds, loads, and environmental factors.

In Chapter 2, a comprehensive literature review was conducted to examine the state of the art in CL for fault diagnosis in railway systems. Following the PRISMA guidelines for systematic reviews, relevant studies were meticulously selected and analyzed to ensure a thorough investigation of the challenges and opportunities in applying CL to PdM. The review was guided by predefined inclusion and exclusion criteria, ensuring that only studies aligned with the research focus were considered. Queries were formulated to search multiple databases, targeting papers addressing key aspects of CL, such as non-stationary environments, catastrophic forgetting, and model adaptability in fault diagnosis. This structured approach allowed the identification of significant research gaps, which informed the direction of the proposed model development. The review aimed to address specific research questions related to improving CL models for PdM in railway systems:

- (RQ1) How can continual learning (CL) improve predictive maintenance (PdM) in railway systems by addressing the challenges of non-stationary environments?
- (RQ2) What are the most effective continual learning (CL) methods for minimizing catastrophic forgetting while maintaining model adaptability in fault diagnosis for industrial systems?
- (RQ3) How do differences in domain similarity and complexity affect the performance of continual learning (CL) models in railway predictive maintenance (PdM)?

Chapter 3 focused on the data generation process, where the dataset was developed in collaboration with CONSTRUCT-LESE using Vehicle-Structure Interaction (VSI) software to simulate train-track dynamics under various operational conditions. The numerical modeling accounted for different wheel imperfections, including flats and polygonization, which are common sources of damage in railway systems. The simulations incorporated various factors, such as varying train speeds, loads, and track irregularities, to represent real-world scenarios. In addition, train segmentation was performed to structure the data into distinct domains based on operational differences such as speed, load distribution, and seasonal conditions. Although the generated dataset provided valuable insight into fault diagnosis, it was limited in terms of diversity, particularly with respect to the range of train types and operational variations. This constraint affected the comprehensive evaluation of the model's scalability and adaptability across different scenarios. Expanding the dataset to include a wider spectrum of operational conditions is essential for future work to enhance the robustness and applicability of the model to real-world railway systems.

Chapter 4 presented the proposed methodology for fault diagnosis in railway wheels using a CL framework. The process was structured into four key stages: data collection, signal processing, convolutional neural networks (CNNs) training with CL, and anomaly detection. The data, initially captured as time series signals, were transformed into images using the Markov transition field (MTF), enhancing the detection of complex patterns within the data. The model framework used an ensemble of smaller CNNs, each tailored to specific domains, and each CNN operated independently to prevent interference between domains. Parameter isolation techniques were applied to preserve the integrity of previously learned knowledge, addressing the challenge of catastrophic forgetting. The system was designed to incrementally improve its fault detection capabilities as new damage scenarios were introduced, enhancing the scalability and adaptability of PdM systems in railway networks. This structured approach provided a comprehensive methodology for managing fault diagnosis in dynamic and evolving environments.

Chapter 5 provided a detailed analysis of the performance of the proposed fault diagnosis model in multiple domains, focusing on key metrics such as forward transfer, backward transfer, and average domain accuracy. The model's ability to handle various train-track interactions, speeds, and loads was evaluated through extensive experimentation, with results indicating that the CL approach significantly outperforms the isolated model. The CL model achieved an average domain accuracy of 91.37%, compared to 56.51% for the isolated model, demonstrating its superior capacity to retain knowledge and generalize in domains. Forward transfer, measuring the model's ability to leverage prior knowledge when learning new tasks, reached a value of 0.748, highlighting the strength of knowledge sharing between domains. The model also exhibited minimal forgetting, with a backward transfer score of nearly zero (0.00024), confirming its ability to retain previously acquired knowledge without performance degradation. Following 10 experimental runs, the highest performance was achieved with a well-balanced configuration of learning rate and momentum, highlighting the importance of hyperparameter optimization. The precision, recall and F1-scores further emphasized the robustness of the model, particularly in domains 1, 2, and 7, where the F1-scores reached 1.00, indicating optimal fault detection. However, domains 8 and 9 posed significant challenges due to increased noise and complexity, resulting in lower F1-scores of 0.81 and 0.76, respectively. The statistical significance of the results was verified using the Friedman test (p-value of 0.00092), while Nemenyi post-hoc analysis highlighted domain-specific performance disparities. These findings underscore the effectiveness of the model in adapting to various operational conditions while also pinpointing areas for improvement in more complex domains.

Looking ahead, future work should prioritize improving the dataset by incorporating a wider range in types, speeds, and operational scenarios to better reflect real-world conditions. This expansion would enable more comprehensive testing of the model's adaptability and robustness across diverse environments. Furthermore, integrating sustainable AI practices is critical to minimize the environmental impact of large-scale machine learning models, which is increasingly important given the growing focus on reducing carbon footprints in AI research. Another promising avenue for exploration is the implementation of early exit networks, which would allow training to stop in specific domains once a predefined accuracy threshold is reached. This threshold-based approach can optimize computational efficiency, significantly reducing resource usage while maintaining high model performance. Such improvements would be particularly valuable in resources-constrained environments such as PdM in railways, where the balance in computational resources with practical performance is crucial. By implementing these strategies, future work can further enhance the scalability, efficiency and practical application of CL models in industrial systems.

Bibliography

Alemi, Alireza, Francesco Corman, and Gabriel Lodewijks (2017). *Condition monitoring approaches for the detection of railway wheel defects*. doi: 10.1177/0954409716656218.

Alvear-Sandoval, Ricardo F., José L. Sancho-Gómez, and Aníbal R. Figueiras-Vidal (2019). *On improving CNNs performance: The case of MNIST*. doi: 10.1016/j.inffus.2018.12.005.

Barke, D. W. and W. K. Chiu (2005). "A review of the effects of out-of-round wheels on track and vehicle components". In: *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 219 (3). issn: 20413017. doi: 10.1243/095440905X8853.

Beaulieu, Shawn et al. (2020). "Learning to continually learn". In: *Frontiers in Artificial Intelligence and Applications* 325. issn: 18798314. doi: 10.3233/FAIA200193.

Bouvier, Victor (Dec. 2021). "Towards Adaptive Learning with Invariant Representations". Theses. Université Paris-Saclay. url: https://theses.hal.science/tel-03663398.

Cai, Wubin et al. (2019). "Experimental and numerical investigation into formation of metro wheel polygonalization". In: *Shock and Vibration* 2019. issn: 10709622. doi: 10.1155/2019/1538273.

Chauhan, Rahul, Kamal Kumar Ghanshala, and R. C. Joshi (2018). "Convolutional Neural Network (CNN) for Image Detection and Recognition". In: *ICSCCC 2018 - 1st International Conference on Secure Cyber Computing and Communications*. doi: 10.1109/ICSCCC.2018.8703316.

Chong, See Yenn, Jung Ryul Lee, and Hye Jin Shin (2010). *A review of health and operation monitoring technologies for trains*. doi: 10.12989/sss.2010.6.9.1079.

Debayle, Johan, Nima Hatami, and Yann Gavet (2018). "Classification of time-series images using deep convolutional neural networks". In: doi: 10.1117/12.2309486.

Du, Simon S. et al. (2021). "FEW-SHOT LEARNING VIA LEARNING THE REPRESENTATION, PROVABLY". In: *ICLR 2021 - 9th International Conference on Learning Representations*.

Fawaz, Hassan Ismail et al. (2019). "Deep learning for time series classification: a review". In: *Data Mining and Knowledge Discovery* 33 (4). issn: 1573756X. doi: 10.1007/s10618-019-00619-1.

Fröhling, R. D. and G. Hettasch (2010). "Wheel - Rail interface management: A rolling stock perspective". In: vol. 224. doi: 10.1243/09544097JRRT339.

Gonçalves, Vítor et al. (2023). "Wheel Out-of-Roundness Detection Using an Envelope Spectrum Analysis". In: *Sensors* 23 (4). issn: 14248220. doi: 10.3390/s23042138.

Guedes, António et al. (2023). "Detection of Wheel Polygonization Based on Wayside Monitoring and Artificial Intelligence". In: *Sensors* 23 (4). issn: 14248220. doi: 10.3390/s23042188.

Hadsell, Raia et al. (2020). *Embracing Change: Continual Learning in Deep Neural Networks*. doi: 10.1016/j.tics.2020.09.004.

Han, Bin et al. (2021). “A new bearing fault diagnosis method based on capsule network and markov transition field/gramian angular field”. In: *Sensors* 21 (22). issn: 14248220. doi: 10.3390/s21227762.

Hertz, Heinrich (1882). “Ueber die Berührung fester elastischer Körper”. In: *Journal fur die Reine und Angewandte Mathematik* 1882 (92). issn: 14355345. doi: 10.1515/crll.1882.92.156.

Hidirov, Shuhratjon and Hakan Guler (2019). “Reliability, availability and maintainability analyses for railway infrastructure management”. In: *Structure and Infrastructure Engineering* 15 (9). issn: 17448980. doi: 10.1080/15732479.2019.1615964.

Hurtado, Julio et al. (2023). *Continual learning for predictive maintenance: Overview and challenges*. doi: 10.1016/j.iswa.2023.200251.

Ioffe, Sergey and Christian Szegedy (2015). “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *32nd International Conference on Machine Learning, ICML 2015*. Vol. 1.

Jardine, Andrew K.S., Daming Lin, and Dragan Banjevic (2006). *A review on machinery diagnostics and prognostics implementing condition-based maintenance*. doi: 10.1016/j.ymssp.2005.09.012.

Jiang, Xuefeng et al. (2019). “CapsNet, CNN, FCN: Comparative performance evaluation for image classification”. In: *International Journal of Machine Learning and Computing* 9 (6). issn: 20103700. doi: 10.18178/ijmlc.2019.9.6.881.

Kalker, JJ (1996). *Book of tables for the Hertzian creep-force law*. Faculty of Technical Mathematics and Informatics, Delft University of Technology.

Kirkpatrick, James et al. (2017). “Overcoming catastrophic forgetting in neural networks”. In: *Proceedings of the National Academy of Sciences of the United States of America* 114 (13). issn: 10916490. doi: 10.1073/pnas.1611835114.

Knoblauch, Jeremias, Hisham Husain, and Tom Diethe (2020). “Optimal Continual Learning has Perfect Memory and is NP-HARD”. In: *37th International Conference on Machine Learning, ICML 2020*. Vol. PartF168147-7.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2017). “ImageNet classification with deep convolutional neural networks”. In: *Communications of the ACM* 60 (6). issn: 15577317. doi: 10.1145/3065386.

Lagnebäck, Robert (2007). “Evaluation of wayside condition monitoring technologies for condition-based maintenance of railway vehicles”. In: *Licentiate thesis / Luleå University of Technology*.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). “Deep Learning”. In: *Nature 521*, pp. 436–444. doi: 10.1038/nature14539.

LeCun, Yann, Léon Bottou, et al. (1998). “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86 (11). issn: 00189219. doi: 10.1109/5.726791.

LeCun, Yann, Corinna Cortes, and CJ Burges (2010). “MNIST handwritten digit database”. In: *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2.

Li, Zhizhong and Derek Hoiem (2018). “Learning without Forgetting”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40 (12). issn: 19393539. doi: 10.1109/TPAMI.2017.2773081.

Lin, Guoliang, Hanlu Chu, and Hanjiang Lai (2022). “Towards Better Plasticity-Stability Trade-off in Incremental Learning: A Simple Linear Connector”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2022-June. issn: 10636919. doi: 10.1109/CVPR52688.2022.00019.

Lopez-Paz, David and Marc'Aurelio Ranzato (2017). "Gradient episodic memory for continual learning". In: *Advances in Neural Information Processing Systems*. Vol. 2017-December.

Lourenço, Afonso, Carolina Ferraz, Jorge Meira, et al. (2023). "Automated green machine learning for condition-based maintenance". In: doi: 10.14428/esann/2023.es2023-85.

Lourenço, Afonso, Carolina Ferraz, Diogo Ribeiro, et al. (2023). "Adaptive time series representation for out-of-round railway wheels fault diagnosis in wayside monitoring". In: *Engineering Failure Analysis* 152. issn: 13506307. doi: 10.1016/j.engfailanal.2023.107433.

Lourenço, Afonso, Diogo Ribeiro, et al. (2024). "Time series data mining for railway wheel and track monitoring: a survey". In: *Neural Computing and Applications* 36. doi: 10.1007/s00521-024-10138-w.

Maschler, Benjamin, Thi Thu Huong Pham, and Michael Weyrich (2021). "Regularization-based Continual Learning for Anomaly Detection in Discrete Manufacturing". In: *Procedia CIRP* 104. issn: 22128271. doi: 10.1016/j.procir.2021.11.076.

Maschler, Benjamin, Sophia Tatiyosyan, and Michael Weyrich (2022). "Regularization-based Continual Learning for Fault Prediction in Lithium-Ion Batteries". In: *Procedia CIRP* 112. issn: 22128271. doi: 10.1016/j.procir.2022.09.091.

Maschler, Benjamin, Hannes Vietz, et al. (2020). "Continual Learning of Fault Prediction for Turbofan Engines using Deep Learning with Elastic Weight Consolidation". In: *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA* 2020-September. issn: 19460759. doi: 10.1109/ETFA46521.2020.9211903.

Mirzadeh, Seyed Iman et al. (2020). "Understanding the role of training regimes in continual learning". In: *Advances in Neural Information Processing Systems* 2020-December. issn: 10495258.

Mohammadi, Mohammadreza et al. (2023). "An Unsupervised Learning Approach for Wayside Train Wheel Flat Detection". In: *Sensors* 23 (4). issn: 14248220. doi: 10.3390/s23041910.

Molęda, Marek et al. (2023). *From Corrective to Predictive Maintenance—A Review of Maintenance Approaches for the Power Industry*. doi: 10.3390/s23135970.

Montenegro, P. A. et al. (2015). "Wheel-rail contact formulation for analyzing the lateral train-structure dynamic interaction". In: *Computers and Structures* 152. issn: 00457949. doi: 10.1016/j.compstruc.2015.01.004.

Mosleh, Araliya, Pedro Alves Costa, and Rui Calçada (2020). "A new strategy to estimate static loads for the dynamic weighing in motion of railway vehicles". In: *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 234 (2). issn: 20413017. doi: 10.1177/0954409719838115.

Mosleh, Araliya, Pedro Aires Montenegro, et al. (2021). "Railway vehicle wheel flat detection with multiple records using spectral kurtosis analysis". In: *Applied Sciences (Switzerland)* 11 (9). issn: 20763417. doi: 10.3390/app11094002.

Parisi, German I. et al. (2019). *Continual lifelong learning with neural networks: A review*. doi: 10.1016/j.neunet.2019.01.012.

Patil, Aseem and Milind Rane (2021). "Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition". In: *Smart Innovation, Systems and Technologies*. Vol. 195. doi: 10.1007/978-981-15-7078-0_3.

Pech, Martin, Jaroslav Vrchota, and Jiří Bednář (2021). *Predictive maintenance and intelligent sensors in smart factory: Review*. doi: 10.3390/s21041470.

Ramesh, Rahul and Pratik Chaudhari (2022). "Model Zoo: A Growing "Brain" That Learns Continually". In: *ICLR 2022 - 10th International Conference on Learning Representations*.

Rolnick, David et al. (2019). "Experience replay for continual learning". In: *Advances in Neural Information Processing Systems*. Vol. 32.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (2013). "Learning Internal Representations by Error Propagation". In: *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence*. doi: 10.1016/B978-1-4832-1446-7.50035-2.

Schapire, Robert E. (2013). *Boosting: Foundations and Algorithms*. doi: 10.1108/03684921311295547.

Schmidhuber, Jürgen (2015). *Deep Learning in neural networks: An overview*. doi: 10.1016/j.neunet.2014.09.003.

Selcuk, Sule (2017). "Predictive maintenance, its implementation and latest trends". In: *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 231 (9). issn: 20412975. doi: 10.1177/0954405415601640.

Shin, Hanul et al. (2017). "Continual learning with deep generative replay". In: *Advances in Neural Information Processing Systems*. Vol. 2017-December.

Standard, British (2003). "Railway applications/Track-Track geometry quality". In: *BS EN*, pp. 13848–1.

Swanson, Laura (2001). "Linking maintenance strategies to performance". In: *International Journal of Production Economics* 70 (3). issn: 09255273. doi: 10.1016/S0925-5273(00)00067-0.

Tripuraneni, Nilesh, Michael I. Jordan, and Chi Jin (2020). "On the theory of transfer learning: The importance of task diversity". In: *Advances in Neural Information Processing Systems* 2020-December. issn: 10495258.

Waeyenbergh, Geert and Liliane Pintelon (2002). "A framework for maintenance concept development". In: *International Journal of Production Economics* 77 (3). issn: 09255273. doi: 10.1016/S0925-5273(01)00156-6.

Xie, Jiawei et al. (2020). *Systematic literature review on data-driven models for predictive maintenance of railway track: Implications in geotechnical engineering*. doi: 10.3390/geosciences10110425.

Yan, Jialin, Jiangming Kan, and Haifeng Luo (2022). "Rolling Bearing Fault Diagnosis Based on Markov Transition Field and Residual Network". In: *Sensors* 22 (10). issn: 14248220. doi: 10.3390/s22103936.

Zeiler, Matthew D. and Rob Fergus (2014). "Visualizing and understanding convolutional networks". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 8689 LNCS. doi: 10.1007/978-3-319-10590-1_53.

Zenke, Friedemann, Ben Poole, and Surya Ganguli (2017). "Continual learning through synaptic intelligence". In: *34th International Conference on Machine Learning, ICML 2017*. Vol. 8.

Zhai, W. M. et al. (2001). "Dynamic effects of vehicles on tracks in the case of raising train speeds". In: *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*. Vol. 215. doi: 10.1243/0954409011531459.

Zhao, Xiaoting et al. (2022). "Markov Transition Fields and Deep Learning-Based Event-Classification and Vibration-Frequency Measurement for ϕ-OTDR". In: *IEEE Sensors Journal* 22 (4). issn: 15581748. doi: 10.1109/JSEN.2021.3137006.