Monitoring and optimization of the railway system

Maria Francisca Sousa de Garcez Osório e Silva

Master's thesis

Supervisor at FEUP: Prof. Gil Gonçalves
Supervisor at GECAD: Prof. Goreti Marreiros

Master in Industrial Engineering and Management

2024-06-27

Resumo

Esta tese aborda a necessidade crítica de melhorar a detecção de irregularidades nos trilhos e deformações nas rodas nos sistemas ferroviários por meio da integração de modelos avançados de inteligência artificial e tecnologias inovadoras. O objetivo principal é desenvolver um modelo que melhore eficazmente a segurança e a eficiência nas operações ferroviárias, aproveitando Variational Autoencoders (VAEs) para uma compressão robusta de dados e aumento semântico, juntamente com um classificador sofisticado e um buffer de memória para experience replay.

Para atingir este objetivo, a tese investiga as técnicas mais comuns para aquisição de dados e como extrair conhecimento a partir dos sinais de sensores com algoritmos de aprendizagem automática e técnicas eficazes de pré-processamento para séries temporais. Por fim, a tese investiga as principais oportunidades e desafios na ampliação do uso da inteligência artificial para manutenção preditiva em sistemas ferroviários e propõe estratégias para superar esses desafios. Estas questões foram abordadas através de uma revisão bibliográfica abrangente e do desenvolvimento de novos algoritmos computacionais eficientes para abordar as lacunas identificadas na literatura.

A metodologia de investigação engloba, então, uma pesquisa detalhada da manutenção preditiva, com foco específico nas capacidades de processamento em tempo real dos modelos desenvolvidos, com vista a uma aplicação prática e realista. Implementando uma combinação de estratégias de continual learning e experience replay, o modelo é projetado para se adaptar ao longo do tempo, melhorando sua precisão preditiva e capacidades de generalização em várias condições operacionais.

O conjunto de dados utilizado abrange uma ampla gama de cenários simulados que refletem interações dinâmicas realistas entre coimboios e ferrovias. Nesta tese é desenvolvido um modelo de deteção de anomalias robusto, envisionado para uma implementação prática. É demonstrado que o recurso a embeddings aquando da extração de featuras, assim como o aumento semântico, melhora a capacidade de deteção de anomalias. Por fim, explora o potencial da integração de continual learning, que contribui para o combate à degeneração a longo prazo dos modelos, por esquecimento de instâncias.

Abstract

This thesis addresses the critical need to improve the detection of irregularities wheel out-of-roundness detection in railway systems through the integration of advanced artificial intelligence models and innovative technologies. The main goal is to develop a model that effectively enhances safety and efficiency in railway operations, leveraging Variational Autoencoders (VAEs) for robust data compression and semantic augmentation, along with a sophisticated classifier and a memory buffer for experience replay.

To achieve this goal, the thesis investigates the most common techniques for data acquisition and how to extract knowledge from sensor signals with machine learning algorithms and effective preprocessing techniques for time series. Finally, the thesis explores the main opportunities and challenges in expanding the use of artificial intelligence for predictive maintenance in railway systems and proposes strategies to overcome these challenges. These issues were addressed through a comprehensive literature review and the development of new efficient computational algorithms to address the gaps identified in the literature.

The research methodology encompasses a detailed exploration of predictive maintenance, with a specific focus on the real-time processing capabilities of the developed models, aiming for practical and realistic application. By implementing a combination of continual learning strategies and experience replay, the model is designed to adapt over time, improving its predictive accuracy and generalization capabilities across various operational conditions.

The dataset used encompasses a wide range of simulated scenarios that reflect realistic dynamic interactions between trains and railways. In this thesis, a robust anomaly detection model is developed, envisioned for practical implementation. It is demonstrated that the use of embeddings during feature extraction, as well as semantic augmentation, enhances the anomaly detection capability. Lastly, it explores the potential of integrating continual learning, which contributes to combating the long-term degradation of models due to forgetting instances.

Acknowledgments

First and foremost, I extend my deepest gratitude to the entire GECAD team for their exceptional support and guidance over the past few months. Special thanks to Lara, João and Diogo, for the most needed moral support.

I am profoundly grateful to Professor Goreti Marreiros, who has been an incredible role model and a constant source of support and inspiration, and thank her for always making the time for me.

I thank my colleague Afonso Lourenço, for the guidance he has given me, whose insights were fundamental in shaping this thesis.

I owe a great debt of gratitude to my supervisor, Professor Gil Gonçalves, and Professor João Reis, for their criticism, encouragement and, mostly, patience. Without their help and dedication this thesis would certainly not have been possible.

To my friends Adelaide and Rita: thank you for being my pillars over the last five years. You are the reason I stand here today, celebrating beside you. Thank you for not giving up on me

To Margarida, Luisa, Sílvia, Carolina, Inês, Bernardo, Manuel, Mário, and Daniel: thank you for all the cherished moments that make the end of this journey feel less like a celebration. Thank you for all the candles you lit for my success.

To Beatriz, Rafael, Afonso e Nuno, who were always there for me, I cannot thank you enough.

To my grandparents, mother, father, Carolina, Manu and Filipe, for always believing in me unconditionally.

And lastly, for my grandfather, the person I wish to dedicate this work to, for always believing I'd be here today, even when I didn't. Thank you for everything.

"The future belongs to those who believe in the beauty of their dreams."

Eleanor Roosevelt

Contents

Acronyms and Symbols

AI	Artificial Intelligence
CL	Continual Learning
OL	Online Learning
PdM	Predictive Maintenance
NN	Neural Network
DNN	Deep Neural Network
PCA	Principal Component Analysis
TACL	Task-Agnostic Continual Learning
VAE	Variational Autoencoder
SAE	Stacked Autoencoder
RL	Representational Learning
ER	Experience Replay
DER	Dark Experience Replay
ID	In-Distribution
OOD	Out-of-Distribution
AUC-ROC	Area Under the Receiver Operating Characteristic Curve
TPR	True Positive Rate
FW	Foward Transfer
IM	Intransigence Measure
KGR	Knowledge Gain Ratio
BWT	Backward Transfer
FM	Forgetting Measure
ReLU	Rectified Linear Unit
KL	Kullback–Leibler

List of Figures

List of Tables

Chapter 1

Introduction

Thesis Statement This thesis was conducted at GECAD (Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development) and aims to demonstrate that integrating advanced AI models with novel technologies can significantly enhance fault detection in railway systems.

By leveraging Variational Autoencoders (VAEs) for data compression and semantic augmentation, coupled with a robust classifier and memory buffer for experience replay, this research hypothesizes that real-time, accurate anomaly detection can be achieved, leading to improved safety and efficiency in railway operations.

1.1 Project framework and motivation

Modern transportation heavily relies on railways, efficiently moving people and goods across vast distances. Safety and reliability are paramount for these systems, demanding sophisticated maintenance strategies to minimize operational costs and ensure passenger and worker well-being. This project is focused in improving this process in order to add value to both parties.

Firstly, the primary focus of this research is to address what is believed to be the major weaknesses in current railway fault detection methodologies.

A critical shortcoming is their limited real-time capabilities. For effective real-time fault detection and response, the ability to efficiently process continuous data streams is paramount. Unfortunately, current methodologies often struggle in this regard, hindering their ability to provide timely insights for preventative maintenance and safety interventions.

Many current models fail when encountering new and unseen instances, lacking the ability to generalize effectively across varied data conditions. This limitation reduces the model's reliability and robustness in real-world applications.

There is a notable deficiency in the continual adaptation of models. Instead of evolving with new data, many models remain static and fail to incorporate new information to improve performance over time. This results in outdated models that do not reflect current operational conditions.

Also, incorporating historical data into the retraining process is essential for refining model accuracy over time. Extracting meaningful insights from these time series data, which is often characterized by autocorrelation and noise, which is still a significant challenge. Regardless, many models still do not effectively utilize a memory buffer for experience replay, in order to adapt to these historical relations.

1.2 Research Focus

Railway systems are critical infrastructures that require high standards of safety and efficiency. The maintenance of these systems poses significant challenges due to the complexity and scale of operations, involving numerous trains, tracks, and associated components.

Current methodologies in railway maintenance primarily rely on reactive strategies, where faults are detected and addressed after they have already impacted the system. This approach, which includes techniques such as mathematical and physical simulations, ultrasonic sensors, and various signal analysis methods, often leads to delays in maintenance actions and increased operational costs. The challenge is further exacerbated by the nature of time-series data from sensors, which are characterized by high autocorrelation and noise, making it difficult to extract meaningful information using traditional techniques.

Recent advancements have demonstrated the potential of latent representations in transforming raw time-series data into more manageable forms, thereby enhancing the detection of subtle patterns indicative of potential failures. However, many of these approaches assume that data follow an in-distribution (ID) or out-of-distribution (OOD) distribution, which is not reflective of most real-world scenarios where data naturally follow sequential and non-stationary distributions. Moreover, traditional batch machine learning approaches fail to handle the sheer volume of data in real-time, leading to outdated models that do not integrate new information efficiently.

Online Continual Learning (OCL) presents a promising paradigm shift by allowing algorithms to learn and update continuously from a stream of incoming data. This approach is particularly crucial in environments with computational resource constraints, where the end of the data stream is indefinite, and the frequency of data arrival limits the available computational resources. However, developing OCL methods that are resource-efficient and capable of maintaining performance across different domains without catastrophic forgetting remains a significant challenge.

For example, neural networks, while powerful, are often under-constrained, allowing them to learn a wide range of functions to minimize a loss function, which can lead to learning representations that do not generalize well to new, unseen data. This flexibility can result in overfitting, where the model captures noise or irrelevant patterns in the training data, thus performing poorly on

new instances. This issue is exacerbated in continual learning, where models must integrate new information while retaining past knowledge, a challenge that static models typically overlook.

By enforcing inductive biases (with methods like experience replay that creates a bias towards past data) and proper assumptions into the learning algorithms, we aim to reduce the search space, ease the learning process for algorithms like this, and improve the model's adaptability to real-world scenarios. This solution is designed to be resource-efficient, reducing the computational overhead during both training and inference phases, and ensuring sustainability in terms of memory usage and parameter growth.

While incorporating the assumption of limited data storage into learning algorithms streamlines processing, it introduces a potential bias. This bias may hinder performance in scenarios where problems deviate from the expected compositional structure. It's crucial to acknowledge that many of these assumptions, such as the inability to store data or the limitation of resources, might be unnecessary or even detrimental in practice. For instance, low-frequency data arrival might allow for storage without significant drawbacks. Similarly, the cost-effectiveness of resource acquisition may negate the need to prioritize limited resources within the learning algorithm itself.

The goal is to balance the bias so as to nudge the path to the correct solution, but not so much as to provoke "overfitting" or deviated results, carefully managing the memory mechanisms to achieve the best results.

1.3 Project goals

The primary goal of this project was to develop an advanced model aimed at optimizing fault detection in railways. The definition of done for this project included the following criteria:

- an Artificial Intelligence model that achieves at least 95% accuracy in detecting and classifying faults in railway systems, including components of trains and tracks;
- Latent space integration (feature extraction component in model that leverages embeddings);
- Leveraging innovative technology (solution implemented novel technologies in a way that has not been previously applied in the railway environment, ensuring a cutting-edge approach to fault detection);
- Ensuring predictive maintenance (effectively detect anomalies related to both the train and the tracks, ensuring comprehensive fault coverage);
- Data stream processing (deploying a system that handles continuous data streams, allowing for real-time fault detection and responses in under 5 seconds);
- Time series data (able to handle the available data from various sensors).

The proposed solution involves several key components: processing a multiple input time series data streams, using a Variational Autoencoder (VAE) to encode the data into embeddings, improve the model's understanding with additional semantic, perform effective anomaly detection through a binary classifier, and incorporating a memory buffer that retains both new and significant historical data.

A standout feature of this solution is its strong emphasis on real-world application. By integrating advanced techniques with practical considerations, the proposed model offers a unique approach to addressing the complexities of railway maintenance. This real-world application motivation is a unique selling point, distinguishing it from other theoretical models.

The proposed model promises to deliver a robust and advanced fault detection system capable of real-time, accurate anomaly detection, ultimately revolutionizing railway maintenance practices.

1.4 Approach adopted for the project

Throughout the realization of the project, a structured 6-step process based on the CRISP-DM (Cross Industry Standard Process for Data Mining) framework was followed. This well-known and widely used framework efficiently and effectively guided the data mining projects. The process steps are illustrated in Figure 1.1.

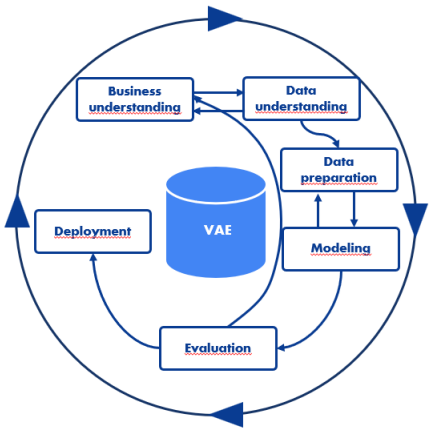


Figure 1.1: CRISP-DM Framework

The initial phase of the project involved a comprehensive analysis and understanding of the railway maintenance operations. The objectives were to comprehend the company's key operations, details of the problems the project aimed to address, objectives to be achieved throughout the project's execution, and expected results.

With the objectives well-defined and a clear understanding of the business, the next phase involved understanding and preparing the available data. A significant portion of the project's time was dedicated to this phase to obtain essential data for building the model and analyzing the results, which is explained in detail in Chapter 3.

Data pre-processing was a critical step that involved transforming, organizing and detailing the data into a usable format for modeling. Various pre-processing techniques, such as downsampling and feature extraction, were applied in order to make the usage of the dataset viable in computational costs, time and performance.

The modeling phase involved building several models, specifically Variational Autoencoders (VAEs) or Stacked Autoencoders (SAEs), and different algorithms for classifiers, each based on different approaches and techniques. The models underwent thorough evaluation based on their performance and domain knowledge, allowing for the validation of their effectiveness.

Once built, the models were evaluated for their performance in terms of accuracy, precision, recall, and F1 score. Additionally, the models were validated through domain knowledge to ensure their practical applicability in the railway maintenance context. The best-performing model was selected for deployment, based on its ability to effectively detect anomalies and optimize fault detection.

The results obtained from the deployment were analyzed, and conclusions were drawn regarding the model's effectiveness in improving railway maintenance practices. The insights gained from this phase provided valuable feedback for further refining and enhancing the model, concluding our projects developments as well.

1.5 Dissertation structure

This dissertation presents a comprehensive investigation into the application of advanced machine learning techniques for predictive maintenance (PdM) within railway systems.

Chapter two explores the current research and achievements relevant to the development of the proposed project both in the practical implementation environment (railways) and the theoretical context (continual learning and semantics).

The third chapter focuses extensively on the data and elaborates on dynamic interactions between trains and tracks and discusses the numerical modeling techniques used to simulate these interactions.

Chapter four defines the methodologies used in this model and how they were leverage to add more value to this project.

In chapter five, the focus shifts towards the practical application of the discussed methodologies in real-world scenarios.

Finally section six discusses findings, delves into future research, and contributions to both PdM and machine learning.

Chapter 2

Literature Review

This chapter establishes the theoretical foundation necessary for comprehending the current landscape of machine learning applications, particularly within the domain of railway condition monitoring.

Section 2.1 lays the groundwork by outlining the fundamental questions that spurred this comprehensive literature review.

Section 2.2 delves into the state-of-the-art for time series data mining techniques employed in railway condition monitoring. This section critically analyzes the key differences amongst these approaches.

Section 2.3 shifts the focus to continual learning, exploring its relevance and potential applications.

2.1 Research Questions

With this central research questions in mind, the following research questions were posed:

1. How can embeddings improve feature extraction and will it improve fault detection in railway systems?
2. Would predictive maintenance suffer improvement if applied in an online continual learning scenario, and how so?
3. What are the benefits of integrating semantic augmentation into the latent representation of data for anomaly detection and generalization?
4. How does the incorporation of a memory buffer for experience replay impact the accuracy and robustness of the fault detection model over time?
5. What are the practical challenges and limitations in implementing this advanced fault detection model in a real-world railway environment?

These questions were addressed in the form of a comprehensive literature review.

2.2 Predictive Maintenance Railways

Predictive maintenance (PdM) represents a significant leap forward in railway maintenance. Unlike traditional methods that involve scheduled checks and repairs, PdM leverages advanced data analysis and machine learning to predict potential equipment failures before they occur. This proactive approach not only enhances safety but also optimizes maintenance resources by reducing unnecessary repairs and downtime, leading to significant cost reductions.

Current maintenance methodologies primarily rely on detecting and fixing problems after they've already impacted the system. One example is leveraging mathematical and physical simulations (Liang et al. (2013)), or using an ultrasonic sensor system (Dwyer-Joyce et al. (2013)), using axle box acceleration (ABA) measurements (Molodova et al. (2016)), extracting features using spectral kurtosis analysis from the frequency domain of signal data (Gómez et al. (2018)) or envelope spectrum analysis from vibration data (Gonçalves et al. (2023)), or with short-time Fourier transform (STFT) and anomaly detection based on locality sensitive hashing technique (LSHAD). (Lourenço et al. (2023)) , or similar (Mosleh et al. (2021a), Lederman et al. (2017), Schenkendorf et al. (2016), Salvador et al. (2016), Mosleh et al. (2020b), Mosleh et al. (2021b), Liang et al. (2015), Molodova et al. (2014)).

However, the integration of time-series data from sensors installed on trains and tracks presents a unique set of challenges and opportunities. The primary difficulty lies in extracting meaningful information from large datasets characterized by autocorrelation and noise. Traditional techniques such as the previous ones often struggle to interpret this data effectively, leading to delayed or inaccurate maintenance actions.

To address these limitations, recent advancements have shifted towards utilizing latent representations. These sophisticated analytical techniques transform raw time-series data into a more manageable form, enhancing the detection of subtle patterns and anomalies indicative of potential failures. Latent representations simplify the data, making it easier for algorithms to perform accurate and timely predictions.

One approach utilizes stacked sparse autoencoders (T. Jorge and Cury (2024)), or using Hidden Markov Models that employ hidden states (Vrignat et al. (2015), Kamlu and Laxmi (2019)), a Bayesian method like Markov chain Monte Carlo (MCMC) (Lam et al. (2017), Falamarzi et al. (2019)),a bivariate Gamma process (Sophie Mercier and Roussignol (2012), Higgins and Liu (2017)), a Autoregressive Moving Average (ARMA) model (J. N. Costa and Andrade (2022)) , a Principal Component Analysis (PCA) (Lasisi and Attoh-Okine (2018), Sharma et al. (2018), Lourenço et al. (2023)) or even a mix use of various feature extraction methods, such as auto-regressive (AR), auto-regressive exogenous (ARX), principal component analysis (PCA), and continuous wavelet transform (CWT) (Mohammadi et al. (2023)), and similars (Bai et al. (2015), Bai

et al. (2015)).

The limitations of offline learning become apparent when dealing with large datasets. While it offers the benefit of unrestricted access to all data from the current task, this approach necessitates additional storage equivalent to the entire dataset's size. This can be impractical for large-scale applications.

In contrast, traditional batch machine learning, where all data is processed simultaneously, struggles with the growing volume of data in real-time scenarios. These methods are not only computationally expensive due to the need to access all data at once, but they also fail to continuously update models with new information. Instead, they require complete model retraining from scratch upon encountering new data.

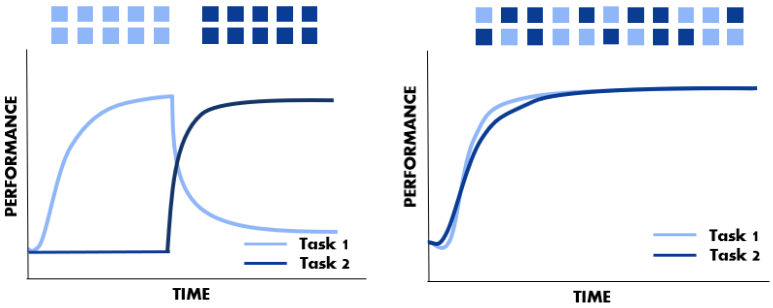
This model focuses on analyzing vibration signals through wavelet packet analysis to detect damage, but is designed for continual processing and learning of the data as it is collected (Wei et al. (2017)) , a system based on a parallelogram mechanism that involves real-time monitoring and analysis (Gao et al. (2019)), with a system using Fiber Bragg Grating (FBG) strain gauge arrays to continuously monitor the condition with immediacy (Liu and Ni (2018)), with a hybrid algorithm that combines online and offline methods, where the online component involves an iterative search used for predicting flange contact points dynamically during the simulation (Sugiyama et al. (2009)), or a hidden Markov model (HMM), to isolate individual wheel behaviour (Lourenço et al. (2023)).

In the online setup, algorithms continuously learn and update from a stream of incoming data without the need for large-scale historical datasets. This adaptation not only saves resources but also enables real-time predictive capabilities, crucial for immediate maintenance decisions and actions.

As the railway industry continues to evolve, the shift towards more dynamic, real-time predictive maintenance methods is becoming increasingly necessary. These developments promise to revolutionize how railway systems operate, ensuring higher levels of safety and efficiency while reducing costs and resource usage. The ongoing research and adaptation of online predictive tools represent a significant step forward in the pursuit of more resilient and reliable railway operations. Figure 2.1 presents this premise in an illustrative image.

From the Figure we conclude that when trained on a second task, the network rapidly forgets the knowledge it learned from the first task. However, training on both tasks in an interleaved way demonstrates that the network has the capacity to learn both tasks. This suggests forgetting isn't due to a limitation in the model itself, but rather how it's trained.

Some work has already been done in this subject, such as an adaptive deep learning framework designed to detect defects in high-speed railway that uses latent space and actively selects and annotates samples to update the training set Gao et al. (2022) , or a statistical monitoring system for railway with ongoing data collection and analysis, where the model continuously updates its



(a) Sequential Training (b) Interleaved Training

Figure 2.1: Schematic illustration of catastrophic forgetting

parameters based on new data Hofer and Bach (2015) , or a few-shot learning model that extracts features from input images and encodes them into a latent space and employs a double-loop continual learning method, which simulates working memory and long-term memory processes Li et al. (2023) , or a model that utilizes a Transformer-based architecture with a graph encoder and temporal decoder to capture complex spatial and temporal dependencies and adapts to new speed patterns Liu et al. (2024).

A summarization of the related work can be found in the following graph:

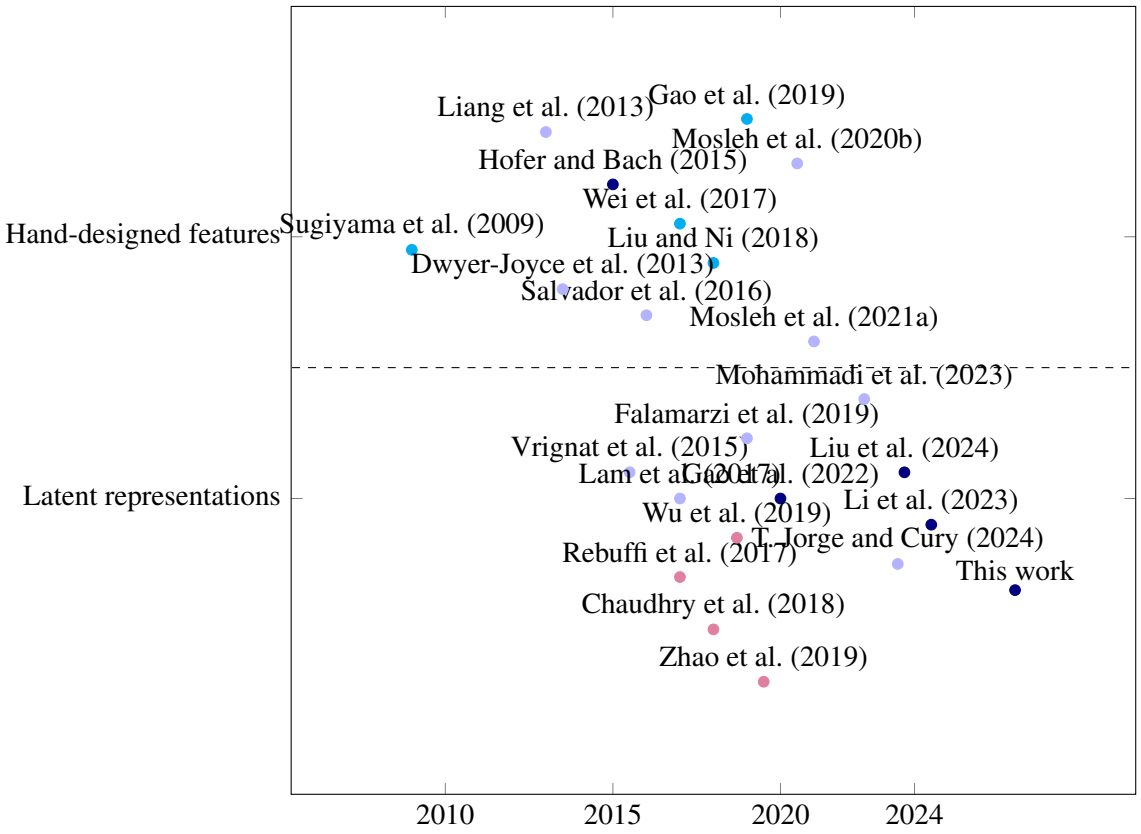


Figure 2.2: Overview of related work: Offline/Batch processing (light blue), Online processing (blue), Offline Continual Learning (purple) Online Continual Learning (dark blue)

2.3 Continual Learning

Continual learning, also known as lifelong learning, refers to the ability of a machine learning model to continuously learn and adapt over time by acquiring new knowledge while retaining previously learned experiences. This approach seeks to mimic the human ability to learn incrementally and cumulatively, without forgetting prior knowledge when exposed to new information.

Problem Definition Continual learning addresses the challenge of the models' ability to learn from a continuous stream of information, without forgetting previously seen tasks.

This issue is denominated catastrophic forgetting. Catastrophic forgetting is the tendency of an artificial neural network to forget previously learned information abruptly and drastically after learning new information. Upon contact with new tasks or data, the model will adjust its weights so as to maximize its performance and minimize errors for that specific task. However, this can significantly alter or even erase knowledge representations of prior tasks, meaning that after seeing task 1, if task 2 is substantially different, the model will be adapted for task 2, losing its capacity for task 1.

This is a fundamental challenge in the development of artificial intelligence: the stability-plasticity dilemma. This dilemma describes the inherent trade-off between the model's ability to keep learning new information (plasticity) and retain previous knowledge (stability). The solution to catastrophic forgetting is not optimizing the stability, and therefore the ability to maintain high performance on old tasks even as it encounters new ones, since this would lead to the deprecation of the plasticity, hence the old tasks' weights would be rigidly fixed and not be flexible enough to accommodate new and unseen tasks.

To better understand and mitigate these problems, continual learning (also referred to as incremental learning or lifelong learning), aims to develop algorithms that never stop learning, but rather update their parameters throughout their lifecycle. To address this, different scenarios of continual learning are described below.

Table 2.1: Comparison of different continual learning paradigms.

	Task labels	**Boundaries**	**Classes**	**Problem**
Class-Incremental Learning	No	Yes	New	Unique
Task-Incremental Learning	Yes	Yes	New	Partitioned
Domain-Incremental Learning	No	Yes	Same	Unique
Task-agnostic	**No**	**No**	**Any**	**Partitioned**
(Task-free)	No	No	Any	Unique

Task-Incremental Learning Task-incremental learning (also known as Task-IL) focuses on a machine learning model's ability to learn new tasks sequentially without forgetting how to perform previous ones. The defining characteristic of this scenario is the assumption that the tasks are introduced one after the other, and it is always clear to the algorithm which task is currently being performed. For this learning process, distinguishable, well-defined tasks (such as explicit labels and task boundaries) are required. Because of this, it is also possible to have task-specific parameters in this case.

A task can represent any type of problem that needs to be solved by the model and there is no limit to the number of type of task that can be presented to the model. However, this flexibility is very limited because the model will not be able to identify the task at hand, instead it will always be fed the task identity (definition of the problem to be solved) and the task boundary (knowledge about task transition).

Several promising approaches have emerged, a breakdown of some key methods include a model that incorporates task-specific distillation loss to minimize the discrepancy between responses in old and updated networks (Shmelkov et al. (2017)), or that generates pseudo-data from previous tasks and combining it with new task data during training, allowing the model to retain knowledge from previous tasks while learning new tasks sequentially (Shin et al. (2017)), or even uses dynamic network expansion, where the model introduces new neurons and connections to handle new tasks, ensuring that the performance on previously learned tasks is not degraded (Yoon et al. (2018)).

Domain-Incremental Learning Domain-incremental learning (also known as Domain-IL) focuses on a machine learning model's ability to adapt to new domains of data while maintaining performance on previously encountered domains. Domain-IL no longer deals with different tasks, but instead considers the possibility of shifts of input data distribution within the same task for which it has been designed. The model is trained sequentially on data from different domains, from which there are intrinsic unique characteristics, even though the task remains consistent.

This scenario works without explicit labels indicating the domain shift or task identity, while never forgetting previous learned domains, but requires availability of knowledge of when a shift occurs. So, even though these models solve a single problem and focuses on the same classes throughout the entire process, there needs to be a method of knowing domain shifts, in order to further detect what type of shift occurred.

Recent wok in this area is for example a model that learns to perform the same task over multiple distributions or environments, with an EWC-augmented loss function, which helps in retaining the knowledge of previously seen domains (Seff et al. (2017)), or a conditional GAN trained on distinct data distributions over time, and employs a variant of EWC to prevent forgetting by protecting critical parameters (Zenke et al. (2017)), or a model that dynamically adjusts the importance of network parameters, with a parameter importance estimation technique based on the Fisher Information Matrix and dynamically expands the network capacity by adding new neurons and connections when necessary (Adel et al. (2020)).

Class-Incremental Learning Class-incremental learning (also known as Class-IL) focuses on a machine learning model's ability to discriminate between a growing number of objects or classes. In this scenario, there is only one unique problem at hand – classification. However, this problem could be partitioned in different tasks that tackle specific areas, therefore are likely to include specific classes that represent specific tasks.

Class-IL should be able to distinguish between classes within an episode, but also be able to identify which task a sample belongs to. These models do not require task identity - that should be inferred by distinguishing between classes from different episodes - but only task boundary. There is not any type of restriction on the classes as well, the models should be prepared to face unprecedented classes, already seen classes and even mix classes that haven't been observed together.

Recent work has been done, mainly, combining exemplar storage and distillation loss, so the model stores examplers for each class and uses them, the loss function is augmented with a distillation term to retain previous knowledge as well (Rebuffi et al. (2017)), on a gradient episodic memory, that stores a subset of examples from previous tasks (classes) and a gradient-based approach to ensure that the learning of new tasks does not interfere significantly with the performance on old tasks (Lopez-Paz and Ranzato (2022)), or, as we have already seen, an approach with Elastic Weight Consolidation to protect important parameter in order to handle new classes without forgetting previous ones (Hu et al. (2019)).

New Instances and Classes New instances and classes scenario of continual learning (also known as NIC) is a method proposed by (Parisi and Lomonaco (2020)) that bridges the scenario of Domain-IL and Class-IL, where data associated with a new task may belong to both known and entirely new classes. This differs from Class-IL since that scenario implies that 1) new tasks introduce only new classes, and 2) task boundaries are available. Due to the nonexistence of both factors, a major focus of NIC is novelty detection (Lomonaco et al. (2020), Mainsant et al. (2022)).

NIC surges from the inflexibility and rigidity felt towards the previous mentioned scenarios, which lacked the acknowledgement of the dynamic nature of the data. The key challenge is the balance between the learning of new classes and the retention of knowledge about previous ones without specific indicators of new classes.

Some interesting work in this area is a model that leverages Elastic Weight Consolidation (EWC) to selectively slow down learning on weight that are important for previous tasks that does not rely on task identity (Kirkpatrick et al. (2017)),

Task-Agnostic Learning Task-agnostic learning (also known as task-free or boundary-agnostic) is a scenario of continual learning where the model is designed to learn and make predictions without explicit information about the addressed task. These models can handle a variety of tasks without needing modifications or task identity and boundaries. They are responsible to automatically identify task changes, therefore an important focus of these types of models is the novelty detection methods.

These are ambitious models, aiming to achieve human-like flexibility and adaptability, battling this often too strong assumption of task label and boundary availability and therefore, preparing them to more real-world settings.

Since this area is rapidly expanding, there are a lot of recent relevant work done such as the usage of a model without explicit labels on tasks, with a Hidden-Mode Markov Decision Process, where the hidden states follow a non-backtracking chain, with a Recurrent Neural Network and a Replay-based Recurrent Reinforcement Learning (Caccia et al. (2023)), or a model that introduces a task-agnostic meta-learning approach to prevent the initial model from over-performing on certain tasks (Jamal et al. (2018)), or a model that proposes a paradigm for task-agnostic exploration that combines agent-centric and environment-centric components (Parisi et al. (2021)).

Chapter 3

Data

In the realm of railway wheel defect detection, the cornerstone of a data-driven approach lies in the quality and comprehensiveness of the available data. While physical-based models seem enticing, the cost of inducing faults in real-world railway systems renders this approach impractical.

Fortunately, the past two decades have witnessed a surge in research delving into the intricate mechanisms of wheel out-of-roundness formation (Johansson and Andersson (2005)). This newfound understanding has paved the way for the development of reliable fault induction techniques within simulation environments (Lourenço et al. (2022)). This numerical modeling approach surpasses the limitations of experimental measurements often plagued by external factors such as environmental disturbances, measurement errors, and electromagnetic interference.

As this research forms part of a collaborative project with the Laboratory of Seismic and Structural Engineering at the Polytechnic University of Porto's Research Unit CONSTRUCT (Department of Civil Engineering), we are fortunate to have access to data generator software courtesy of their generous collaboration.

3.1 Train-track dynamic interaction

To investigate train-track dynamic interaction, validated numerical simulations were employed. These simulations leveraged VSI software, previously described and validated by the authors in a prior publication(Montenegro et al. (2015)). The VSI model established a coupling between the train and track through a sophisticated 3D wheel-rail contact model. This model incorporates Hertzian contact theory to compute normal contact forces, while the USETAB routine assesses tangential forces arising from rolling friction creep phenomena (Hertz (1882), Kalker (1996)). Notably, the VSI software operates within the MATLAB® environment, enabling the import of structural matrices from both the train and track structure. These structures were previously modeled independently using a finite element (FE) package.

The track structure within the simulations was meticulously represented using beam elements for rails and sleepers. Additionally, spring-dashpot elements were employed to capture the behavior of flexible layers, such as ballast and fasteners/pads. Finally, mass point elements were included to account for the mass of the ballast. The train model, constructed within ANSYS®, utilized a multibody formulation. This formulation incorporates spring-dashpot elements to simulate the flexibility of primary and secondary suspensions, rigid beams to represent the vehicle's rigid body movements, and mass point elements positioned at the center of gravity of each body (carbody, bogies, and wheelsets) to capture their mass and inertial effects. A visual depiction of this numerical modeling approach is presented in Figure 3.1.

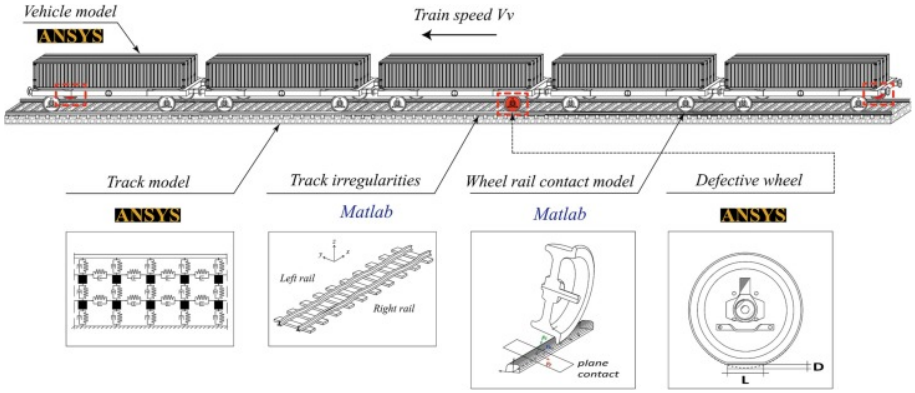


Figure 3.1: Train-track dynamic interaction model with ANSYS® and Matlab: the wheels marked in red were defective in the damaged scenarios, namely the first and last on the right side, and the sixth on the left side.

3.2 Numerical wayside modelling

There is an accelerometer and strain gauge installed along the track for the out-of-roundness detection system. To assess and confirm the methodology outlined, extensive 3D simulations based on the train-track interaction model were conducted. The virtual wayside monitoring system includes measurements taken from 24 distinct locations positioned on the rail between two sleepers. Figure 3.2 exemplifies 4 sensors installed along the stretch of the track in midspan location.

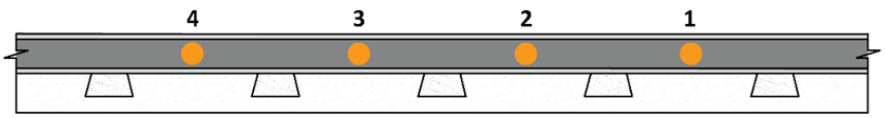


Figure 3.2: Virtual wayside monitoring system: sensor placement

To test and validate the automatic out-of-roundness diagnosis, virtual simulations were carried out with the baseline and damaged wheels. Only after successful validation has the model gained the potential to identify different types of trains and the damage that the wheels may have, and it is now able to cope with being used with real experimental data, since in real conditions the wheels are usually not completely smooth, and they often have imperfections (i.e., flats and polygonization).

This is an important factor due to the extreme variations that these small sized out-of-roundness irregularities can cause with variations in contact forces and vibrations on either track or train components. When looking closely at Figure 3.1, there are three wheels that are considered faulty. Two intervals of flat lengths L_w regarding wheels flats were considered, named as L1 and L2. To define the lower and upper limits of the flat lengths for each interval L1 and L2, the uniform distributions U (25, 50) mm, and U (50,100) mm were used, correspondingly. For the depth of the wheel flat D_w the following Equation 3.1 is used to calculate(Zhai et al. (2001)):

$$D_w = \frac{L_w^2}{16R_w} \tag{3.1}$$

where R_w is the wheel's radius. The vertical profile deviation of the wheel flat is defined as Equation 3.2:

$$Z = -\frac{D_w}{2}\left(1-\cos\left(\frac{2\pi x_w}{L_w}\right)\right)\cdot H\left(x_w-(2\pi R_w-L_w)\right), \quad 0\le x\le 2\pi R \tag{3.2}$$

In which H corresponds to the Heaviside function and x_w represents the coordinate aligned with the longitudinal direction of the track. Figure 3.3 exemplifies the effect that different wheel flat insensitivity has.

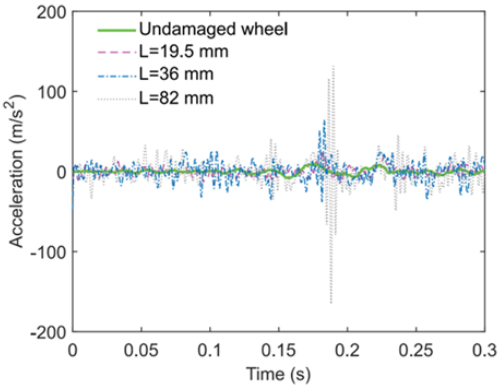


Figure 3.3: Acceleration time series for different wheel flat lengths (L_w): 0, 19.5, 36 and 82 mm

For wheel polygonization, a periodic radial tread irregularity around the wheel circumference was considered by varying wavelengths λ as a function of the harmonic order θ and wheel radius, represented in Equation 3.3:

$$\lambda = \frac{2\pi R_w}{\theta}, \quad \theta = 1,2,3,\ldots,n \tag{3.3}$$

The selected wheel profiles were characterized by the wavelengths in the first 20 harmonics, with the 6th to 8th harmonic orders being dominant. Different irregularity wheel profiles were gener-

ated based on the sum of sine functions $H = 20$ as follows in Equation 3.4:

$$w(x_w) = \sum_{\theta=1}^{H} A_\theta \sin\left(\frac{2\pi}{\lambda} x_w + \phi_\theta\right) \tag{3.4}$$

where A_θ is the amplitude of the sine function for each wavelength, calculated by Equation 3.5:

$$A_\theta = \sqrt{2} \cdot 10^{\frac{L_w}{20}} \cdot w_{\text{ref}} \tag{3.5}$$

with $w_{\text{ref}} = 1\,\mu\text{m}$.

The wheel irregularity level L_w values were selected based on the irregularity spectrum in Figure 3.4, produced with measurement values of four wheels with polygonal damage (Cai et al. (2019)). Considering phase angles of the sine functions that are uniformly and randomly distributed between 0 and 2π, several wheel irregularity profiles were generated with Equation 4 to obtain different damage severities between 0.8 mm and 1.2 mm.

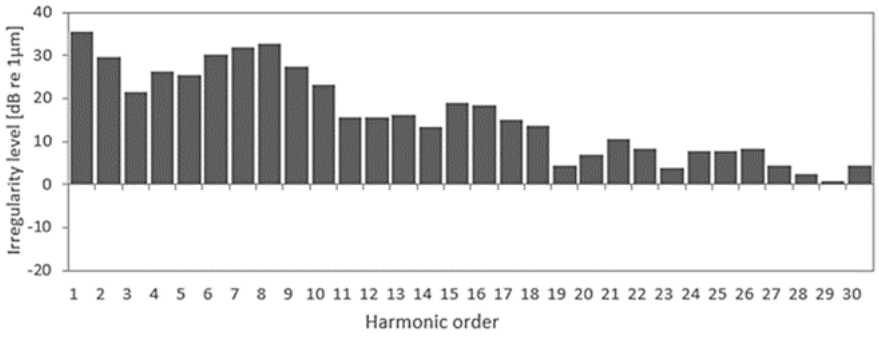


Figure 3.4: Wheel irregularity amplitude spectra (L_w) and harmonic order () distribution

In real circumstances, the rails will show small imperfections which affect the contact force values. Consequently, 10 unevenness profiles, demonstrated in Figure 3.5, appeared due to wavelengths between 1 m and 30 m that cover the D1 wavelength interval.

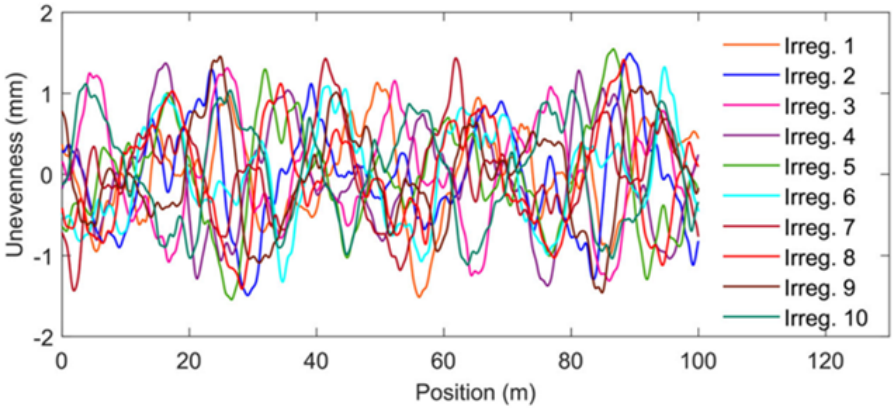


Figure 3.5: Simulation of 10 unevenness rail profiles for wavelengths between 1 m to 30 m

The amplitude of the unevenness profile varied between -2 mm and 2 mm, for a total simulation length of 100 m. These were based on the 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 those of wheel flat and polygonization. Thus, the exited frequencies due to the track are much lower than those of a defective wheel. More details about the generation of unevenness profiles can be found elsewhere (Mosleh et al. (2020a)). By reproducing the responses of the rail considering several speeds, wheel profiles, and rail irregularities, different time-series samples were simulated to capture a range of operating conditions (Araliya Mosleh and Calçada (2023)). The dynamic analyses were carried out for two passing vehicles.

3.3 Dataset

The simulated data is comprised on a set of two types of trains: the Laagrss vehicle and the Alfa vehicle. The former is designed for hauling freight, typically carrying heavier loads at slower speed. In contrast, Alfa is a train built for passenger transport, prioritizing speed and lighter weight.

Specific information about the available scenarios can be found in the Tables 3.1 and 3.1, which characterize the various operational conditions, load states, defect types, and measurement parameters for both baseline and damage scenarios.

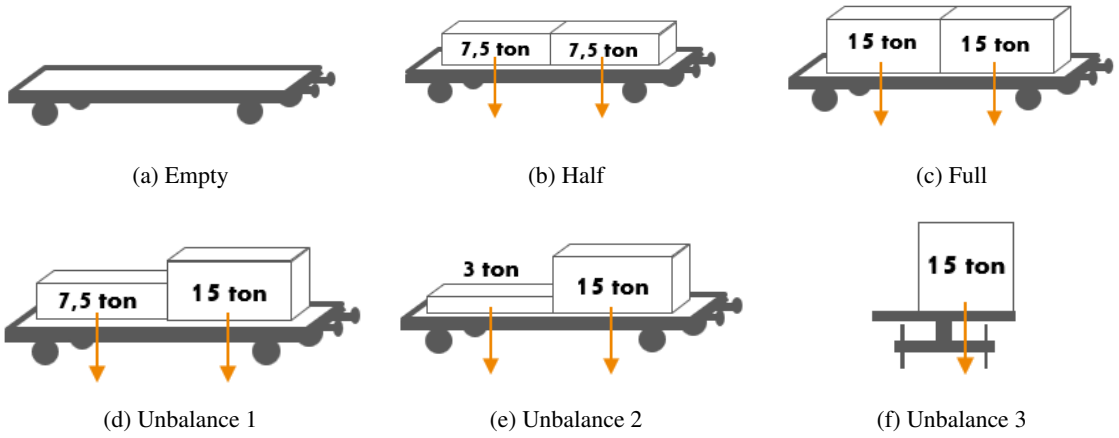
Table 3.1: Comparison of Laagrss Baseline and Damage Scenarios

	Baseline scenarios	**Damage scenarios**	
		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

The six different options for train load follow schemes in Figure 3.6, respectively:

Table 3.2: Comparison of Alfa Baseline and Damage Scenarios

	Baseline scenarios	**Damage scenarios**	
		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



(a) Empty (b) Half (c) Full

(d) Unbalance 1 (e) Unbalance 2 (f) Unbalance 3

Figure 3.6: Load schemes for trains

From this dataset, the chosen semantic information was the train load, the train speed and the number of wheels, that were captured by the signal, because this was information with a low cost for obtaining, and holds a lot of representative power.

3.4 Scenarios

Most implementations operate under the assumption that data follow an in-distribution (ID) pattern or even an out-of-distribution (OOD) pattern. However, in real-world scenarios, this is seldom the case. Typically, data follow a natural distribution, and it is crucial to preserve this sequence, even though it might not always optimize the model's performance.

To genuinely embed the continual learning aspect of real-world applications, the data must be carefully segmented and identified. In this project, ensuring the model's capability to handle different data domains is paramount. Consequently, various scenarios were created to represent possible temporal evolutions that respect the natural orderly sequence of time.

These divisions are seasonal and recurrent, aimed at thoroughly testing the model's ability to retain past knowledge and manage new information. The scenarios are as follows and are represented in Figure 3.7:

- **Domain 1**: This period is characterized by higher speed trains to minimize traffic disruptions during peak seasons.

- **Domain 2**: This period features slower speed trains, operating in a less congested railway setting, typical of off-peak seasons.

- **Domain 3**: During this period, there is a high flow of commerce and transport, characterized by high-speed trains with fully loaded wagons, representing summer time.

- **Domain 4**: This period is the inverse of Domain 3, featuring the slowest speeds and never completely filled wagons, indicative of a slow flow of transport and goods, akin to winter time.

- **Domain 5**: In this scenario, every train adheres to the speed limit according to its load scheme, with lighter trains traveling at higher speeds and heavily loaded trains at lower speeds. This scenario represents an exceptional event, such as the implementation of a new government regulation.

- **Domain 6**: This period features 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 scenarios ensure a comprehensive assessment of the model's capability to handle real-world continual learning challenges, preserving the natural data sequence while managing different operational conditions effectively.

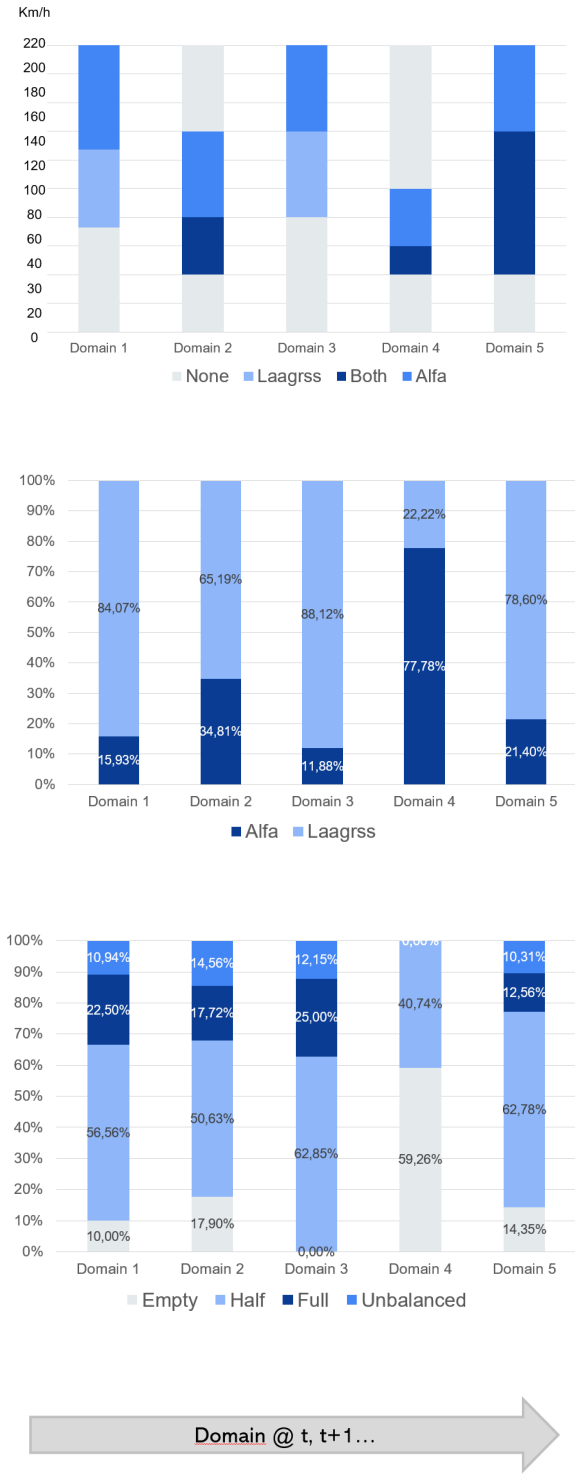


Figure 3.7: Sample partitioning throughout scenarios

In Chapter 3, the simulation setup and data acquisition processes crucial for modeling railway track and wheel interactions. The diverse scenarios crafted for the dataset underscored the potential complexities in railway dynamics, providing a robust foundation for the subsequent analysis. The chapter successfully established a comprehensive dataset that reflected a wide spectrum of realistic

operational conditions, thereby setting the stage for applying advanced AI techniques to enhance fault detection capabilities in railway systems.

Chapter 4

Proposed Approach

This section is dedicated to exploring a variety of data analysis techniques and tools that are pivotal in the context of railway systems maintenance. The proposed model operates by:

1. Downsampling and semantic pre-processing
2. Processing multiple input data streams with the VAE. It encodes this data into a compressed, latent representation, capturing the underlying structure
3. The latent representation is augmented with semantic
4. This enriched representation is then fed into a classifier, which outputs a binary classification (normal or anomaly). Anomaly detection with the latent representation. This enriched representation is then fed into a classifier, which outputs a binary classification (normal or anomaly). Anomaly detection with the latent representation
5. In parallel, the system incorporates a memory buffer that enables experience replay by incorporating both new and significant historical data (including anomalies) into the retraining process, the model's accuracy is refined over time.

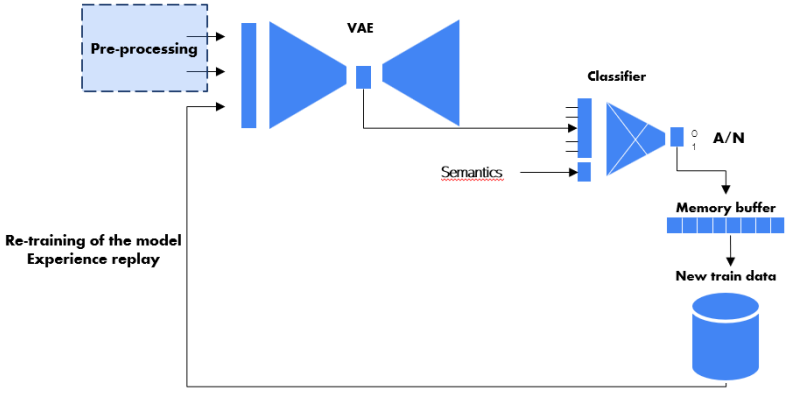


Figure 4.1: Architecture of proposed model

4.1 Pre-processing

4.1.1 Hilbert Transformation - Downsampling

The Hilbert transform is a fundamental tool in signal processing, such as those generated by moving trains, used primarily to derive the analytic signal from a real-valued signal. The analytic signal is a complex signal whose real part is the original real signal and whose imaginary part is the Hilbert transform of the original signal. This transformation provides a way to analyze signals in the frequency domain and is particularly useful for amplitude and phase modulation processes.

The mathematical formula for the Hilbert transformation is represented in Equation 4.1:

$$H\{x(t)\} = x(t) * \frac{1}{\pi t} = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(\tau)}{t-\tau} d\tau \tag{4.1}$$

4.1.2 Peak Detection - Semantics

To identify the peaks in the strain gauge data, we utilize the *scipy.signal.find_peaks* function. This function locates the local maxima in a signal array, which correspond to the positions where each wheel exerts maximum strain on the sensor. The algorithm can be described mathematically as follows:

Given a discrete signal $x[n]$, a peak is detected at position n_i if the following condition holds:

$$x[n_i] > x[n_i - 1] \quad \text{and} \quad x[n_i] > x[n_i + 1] \tag{4.2}$$

Additionally, the function can be configured to detect peaks that are higher than their immediate neighbors by a specified prominence. The prominence of a peak measures how much the peak stands out due to its intrinsic height and its location relative to other peaks. Mathematically, the prominence P of a peak at position n_i is defined as:

$$P(n_i) = x[n_i] - \max\left(\min(x[L], x[R])\right) \tag{4.3}$$

where L and R are the indices of the left and right bases of the peak, respectively.

By applying the peak detection algorithm to the strain gauge data, we can identify not only the positions of the wheels, but also the total number of wheels of each vehicle.

4.2 Representation Learning

4.2.1 Variational Autoencoders

Variational Autoencoders (VAEs) are a powerful tool for anomaly detection in complex datasets. They excel at learning the underlying structure of data and identifying deviations from that struc-

ture, making them well-suited for tasks where anomalies can be subtle or context-dependent (Wiewel and Yang (2019)).

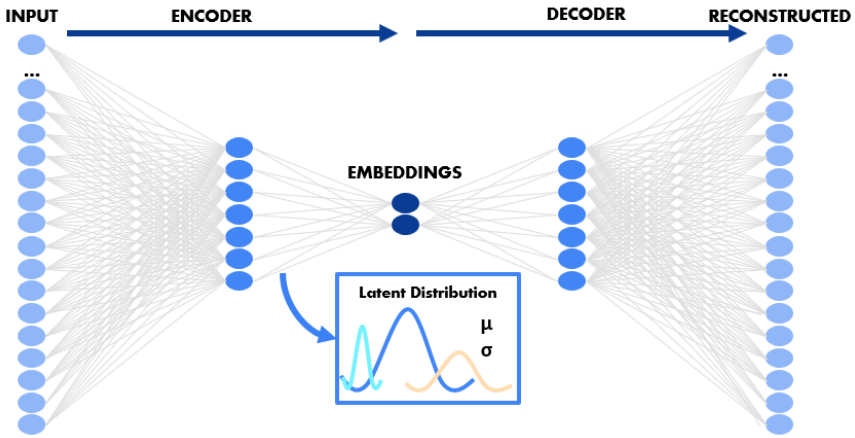


Figure 4.2: Architecture of a Variational Autoencoder

VAE's structure comprises of two main components: an encoder and a decoder. The input data is fed to the encoder, which will map this data into a latent distribution (a compressed, low-dimensional representation of the data) and then reconstructed by the decoder. The model is trained to understand how the data is distributed, minimizing the reconstruction loss (difference between reconstructed data and original data) and maximizing the likelihood between the learned latent distribution and a prior distribution, given by the Equation 4.4:

$$\text{ELBO} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)] - D_{\text{KL}}(q_\phi(z \mid x) \parallel p(z)) \tag{4.4}$$

Rep Learning perfect in this case These architectures are used mainly in representational learning, also known as feature learning, that is a set of techniques in machine learning that allow a system to automatically discover the representations needed for feature detection or classification from raw data. Learned representations are more effective than hand-designed features, since they capture complex patterns and interactions in the data, even in high-dimensional data. RL is also very useful to enhance generalization (using knowledge from seen instances on unseen ones is critical but difficult, requiring a better grasp of the underlying structure), to mitigate overfitting (otherwise, the model will perform well only in seen data, but the goal is for a more consistent performance across the data, creating a more robust model) and for anomaly detection (since a lot of anomalies can be dependent on context in data, and learning intricate patterns will help the model to better identify anomalies and faults).

Specific use These models will be used in order transform am initial signal of 834 data points into a more representative sample of just 20 points, that is more rich and informative of the signal's underlying structure and characteristics.

4.2.2 Stacked Autoencoders

Stacked Autoencoders (SAEs) are a powerful deep learning model particularly well-suited for representation learning and dimensionality reduction. Unlike traditional autoencoders, which consist of a single encoder-decoder pair, SAEs leverage multiple layers of encoders and decoders stacked on top of one another, allowing for the learning of hierarchical features from the data. This makes SAEs particularly effective at capturing complex, high-dimensional patterns in railway maintenance data.

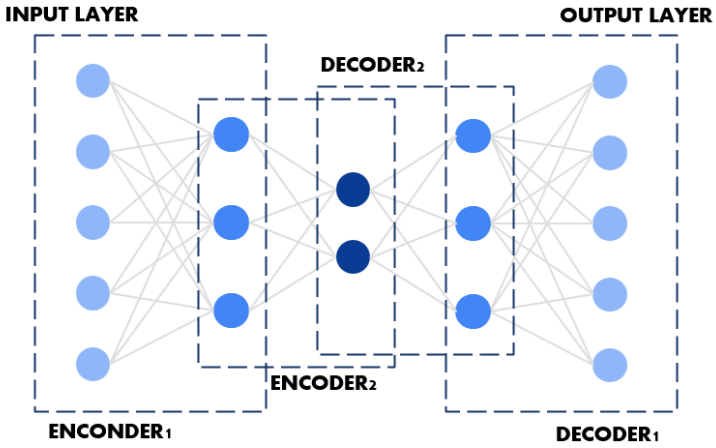


Figure 4.3: Architecture of a Stacked Autoencoder

Each encoder layer transforms the input into a progressively lower-dimensional representation. This hierarchy of encoders extracts increasingly abstract features from the data, which are crucial for capturing the underlying structure and patterns. Each decoder layer aims to reconstruct the original input from its encoded representation. By minimizing the reconstruction error, the model ensures that the learned representations preserve the essential characteristics of the input data.

The training process of SAEs typically involves training individually each layer is in an unsupervised manner. The first layer is trained as a standalone autoencoder, then its output is used as the input for the next layer, and so on, which helps initializing the weights of the deep network effectively.

The reconstruction loss used to train the SAE can be represented as follows:

$$L_{recon} = \frac{1}{N}\sum_{i=1}^{N} |x_i - \hat{x}_i|^2 \tag{4.5}$$

where x_i is the original input, $\hat{x}_i$ is the reconstructed input, and N is the number of samples.

The primary advantage of using SAEs in this context is their ability to create a compact, informative representation of the input data, which can be used for various downstream tasks such as

anomaly detection and predictive maintenance. The learned features are often more robust and discriminative compared to those extracted using traditional methods, leading to improved performance in detecting anomalies and identifying potential faults.

VS VAE In the context of this project, if the focus is on robust feature extraction and simplicity, an SAE might be more appropriate. However, if the goal is to leverage the probabilistic nature of the model for better anomaly detection and uncertainty quantification, a VAE would be a better choice. Given the complex nature of railway fault detection and the potential benefits of a probabilistic approach, the VAE could offer more advantages, especially in identifying rare and subtle faults.

4.3 Anomaly detection/ classification

why classifier

The model leverages classifiers on learned embeddings, as opposed to decoder-based reconstruction errors, for anomaly detection.

Embeddings, which are concise, lower-dimensional representations of the input data, capture the most salient features necessary for distinguishing normal from anomalous behavior. Classifiers can quickly evaluate these embeddings to identify anomalies. This approach significantly reduces computational overhead compared to decoder-based methods, which require the complex process of reconstructing input data from embeddings. By focusing on embeddings, classifiers can leverage the rich, informative features extracted by the model, facilitating more accurate and scalable anomaly detection.

why not decoder

Furthermore, classifiers on embeddings are less susceptible to noise and irrelevant variations in the data, which can adversely affect decoder-based approaches. Decoders aim to recreate the original data, making them sensitive to minor perturbations that may not be relevant to anomaly detection. This sensitivity can lead to higher false-positive rates, where normal data is incorrectly identified as anomalous due to minor reconstruction errors. In contrast, classifiers on embeddings emphasize the most critical features for the task at hand, enhancing robustness against such noise.

4.4 Semantics

Traditional anomaly detection methods often rely solely on statistical properties like data values or frequencies. However, incorporating semantics, the meaning and contextual relationships within the data, can significantly improve anomaly detection performance.

Unlike purely statistical approaches that flag outliers based on deviation from the norm, semantic anomaly detection delves deeper. It considers what the data represents and how it interacts with other data points within the same context, and allows to establish more meaningful and rich

connections. This allows for a more nuanced understanding of anomalies, leading to fewer false positives.

For example, in a purely statistical approach, a high wheel acceleration reading might be flagged as an anomaly for all trains. However, with semantics, the system can consider the specific train type. An acceleration normal for a lightweight train (Train A) could be a severe anomaly for a heavier train (Train B).

This way, semantics are (Raz et al. (2002)) learning techniques that infer these invariants. Even though semantic information is not free and sometimes hard to obtain, the cost is relatively low, specially when compared to the increase in performance that they produce.

In fact, there has been research done in the effectiveness of multi-task learning with auxiliary self-supervised objectives. Experimental results are indicative that such not only lead to improved anomaly detection, but also improvements in generalization (Ahmed and Courville (2020)).

Semantic understanding empowers the model to generalize its knowledge to unseen data. By learning the underlying relationships between features, the model can predict anomalies in scenarios (Train C) it hasn't encountered during training. It essentially leverages the "global coherence" of semantic factors to make informed decisions (Cai et al. (2019)).

In this particular case, the chosen semantic information was the train load, the train speed and the number of wheels, that were captured by the signal. Since the train load is stored as a categorical variable, they were transformed beforehand into a numerical labeled array.

4.5 Experience Replay

Generator models such as VAEs are used in pseudo-rehearsal based methods, a technique in machine learning where instead of retaining a physical dataset of past examples (exemplars buffer), a generative model is used to synthesize new data samples. This method allows the system to "rehearse" past knowledge using the generated samples, reinforcing old information without the need to store actual past data.

Domain incremental learning challenges models to adapt across dynamically changing domains without forgetting. The objective is to minimize the total risk across all domains, as seen in Equation 4.6:

$$L^*(\theta) = L_C(\theta) + L_{C_{1:t-1}}(\theta) = \mathbb{E}_{(x,y)\sim D_C}[\ell(y,h(x;\theta))] + \sum_{i=1}^{t-1} \mathbb{E}_{(x,y)\sim D_{C_i}}[\ell(y,h(x;\theta))] \tag{4.6}$$

To address catastrophic forgetting, a surrogate objective is utilized, balancing learning from new data and maintaining performance on past data, represented in Equation 4.7:

$$L(\theta) = L_C(\theta) + \beta \cdot J_{\text{replay}}(\theta) = L_C(\theta) + \beta \cdot \mathbb{E}_{(x',y')\sim M}[\ell'(y',h(x';\theta))] \tag{4.7}$$

Experience replay is a related concept used primarily in reinforcement learning, where a model re-encounters previous experiences, commonly stored in a replay buffer, during training. By repeatedly sampling from this buffer, the model breaks the temporal correlation that characterizes streaming and sequential data and helps in stabilizing the learning and improving the convergence of the model.

In the context of this project, experience replay will be responsible to dull the bias towards recent data, where only the most recent railway information is sufficient to maintain satisfactory performance on anomaly detection. By generating new samples that reflect past data, the model can continuously reinforce its understanding of "normal" train behavior and stabilize this concept. This ensures the model remains effective even with evolving data streams.

Label Strategy Label strategy refers to the method used for handling and utilizing labels during training in continual learning when considering the buffer maintenance and rehearsal strategies.

Whether the memory buffer holds the true label or a predicted label by the model depends on the context in which it is inserted and its goal. Basic experience replay could be implemented and, therefore, true labels are stored and used, which ensures accurate supervision during rehearsal but may not always be feasible in real-time applications.

Another option is dark experience replay (DER) that incorporates knowledge distillation - uses the outputs of the model (soft labels) on past data to guide the learning of new tasks. This helps retain knowledge by preserving the model's responses to old tasks. Some implementations of DER involve adversarial examples to make the model more robust to changes and retain previous knowledge.

We experimented with both these strategy to conclude which one is more appropriate and performs best in our model.

Buffer Maintenance Policy Buffer maintenance policy refers to the rules and strategies used to manage a fixed-size memory buffer in order to ensure that the buffer contains the most informative samples.

The mechanism used is "reservoir sampling", a technique for selecting a random sample of k items from a stream of n items.

The probability that item m remains after processing all n items in reservoir sampling is given by the product:

$$P(\text{item } m \text{ remains}) = \left(1 - \frac{k}{m+1}\right) \times \left(1 - \frac{k}{m+2}\right) \times \cdots \times \left(1 - \frac{k}{n}\right)$$

Simplifying this, we find:

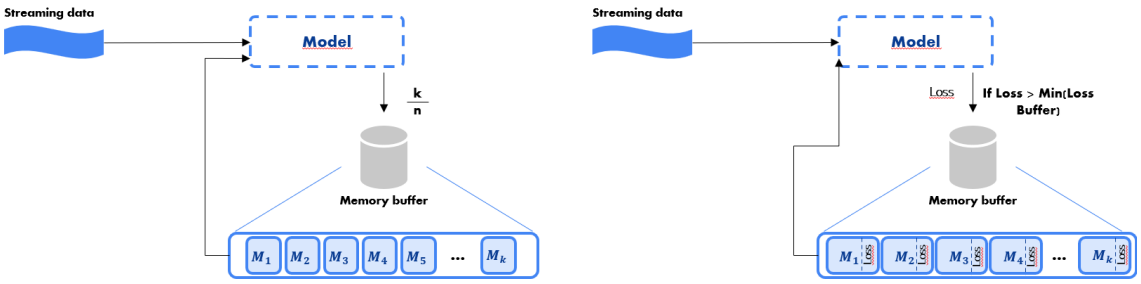
$$P(\text{item } m \text{ remains}) = \frac{k}{n}$$

Therefore, ensuring that each item in the stream has an equal probability of $\frac{k}{n}$ of being included in the reservoir, achieving an unbiased random sampling from the stream.

Additionally, a loss-based sampling strategy complements reservoir sampling by prioritizing the retention of data points based on their contribution to the learning loss. This method targets the preservation of "hard" examples that are typically more informative and challenging for the model, thus ensuring that critical features of the data are not overlooked as the model updates. This strategy can be mathematically described in Equation 4.8 as:

$$D_{\text{retain}} = \{(x,y) \mid \ell(f(x;\theta),y) > \tau\} \tag{4.8}$$

where D_{retain} includes data points (x,y) whose loss $\ell(f(x;\theta),y)$ exceeds a threshold τ. This method ensures that "hard" examples, which provide significant learning value, are preferentially kept in the memory buffer.



(a) Reservoir Sampler

(b) Loss-Based Sampler

Figure 4.4: Architectures of Experience Replay

Rehearsal Strategy Rehearsal strategy characterizes the rules in which the stored data in the buffer is periodically trained.

Traditional methods include uniform balanced sampling, which ensures that the buffer maintains a representative subset of the entire data distribution, and minimal rehearsal strategies, where only a minimal subset of data is rehearsed to save computational resources.

Advanced strategies such as Maximum Loss, Minimal Margin, Minimal Logit-Distance, and Minimal Confidence aim to prioritize samples based on their potential impact on model performance, focusing on those that either exhibit high loss or are difficult to classify.

Loss-based approaches like Maximally Interfered Retrieval (MIR) perform virtual updates on the model with new data batches to identify and rehearse old samples that are most interfered with, thus emphasizing the retention of crucial information.

Since there are not enough different distributions to justify a complex and sophisticated strategy, the strategy employed in our model is a simple, uniform use of the maximum amount of samples stored in the memory.

Chapter 4 explored the approaches for anomaly detection in railway systems, leveraging the strength of Variational Autoencoders and sophisticated classification techniques. The different methodologies such as semantic augmentation and experience replay implemented into the anomaly detection framework are particularly innovative. Next chapter will delve on how these approaches were manipulated in order to provide the best results possible.

Chapter 5

Discussion and Validation

5.1 Fault Diagnosis

This section will cover the strategic implementation of the analytical tools discussed in Section 2, focusing on their application within real-world scenarios.

Evaluation Metrics In this section, we employ a suite of metrics to rigorously evaluate the performance of our predictive models, ensuring their effectiveness in fault diagnosis within railway systems. The metrics used are defined as follows:

- **Accuracy**: Defined as the ratio of correctly predicted instances to the total instances, accuracy is mathematically represented as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{5.1}$$

- **Precision**: This metric indicates the ratio of true positive predictions to the total positive predictions, critical for minimizing false positives:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{5.2}$$

- **Recall (Sensitivity)**: Recall measures the model's ability to detect all relevant instances effectively:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{5.3}$$

- **F1 Score**: The F1 Score is the harmonic mean of precision and recall, providing a balanced measure of a model's precision and recall:

$$F1 \text{ Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.4}$$

- **Area Under the Receiver Operating Characteristic Curve (AUC-ROC)**: This metric assesses the model's discrimination capability at various threshold settings, represented by:

$$\text{AUC-ROC} = \int_0^1 \text{TPR}(t)\, dt \tag{5.5}$$

where $\text{TPR}(t)$ is the true positive rate at threshold t.

These metrics collectively provide a comprehensive evaluation of the diagnostic models, ensuring that they meet both the statistical and practical requirements necessary for effective implementation in railway maintenance operations and will be used throughout the experiments made in this section.

Continual Learning Metrics In the context of continual learning, it is essential to evaluate the model's performance over time, particularly how it learns new tasks and retains knowledge from previous tasks.

- **Forward Transfer (FWT)**: measures the influence of learning new tasks on the performance of previous tasks:

$$\text{FWT} = \frac{1}{N-1} \sum_{i=1}^{N-1} (R_{0,i} - R_{i,i}) \tag{5.6}$$

where $R_{0,i}$ is the initial performance on task i, and $R_{i,i}$ is the final performance on task i. A higher FWT indicates that learning new tasks has positively impacted previous tasks.

- **Intransigence Measure (IM)**: evaluates the difficulty a model has in learning new tasks.

$$\text{IM} = \frac{1}{N} \sum_{i=1}^{N} \left(R_{\text{joint},i} - R_{\text{continual},i} \right) \tag{5.7}$$

where $R_{\text{joint},i}$ is the performance of the model trained jointly on all tasks, and $R_{\text{continual},i}$ is the performance of the model trained continually. Lower IM values indicate better performance in learning new tasks.

- **Knoowledge Gain Ratio (KGR)**: proportion of knowledge retained from initial training to continual learning. Formally:

$$\text{KGR} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{R_{\text{continual},i}}{R_{\text{initial},i}} \right) \tag{5.8}$$

where $R_{\text{initial},i}$ is the initial performance on task i, and $R_{\text{continual},i}$ is the final performance after continual learning on task i. Higher KGR values indicate better retention and integration of knowledge.

- **Backward Transfer (BWT)**: measures the influence of learning new tasks on the performance of old tasks.

$$\text{BWT} = \frac{1}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (R_{j,i} - R_{i,i}) \tag{5.9}$$

  where $R_{j,i}$ is the performance on task i after learning task j, and $R_{i,i}$ is the initial performance on task i. Positive BWT values indicate that learning new tasks has improved the performance on previous tasks.

- **Forgetting Measure (FM)**: quantifies the maximum forgetting experienced by a model. It is the maximum performance drop observed on any task during continual learning.

$$\text{FM} = \max_{i \in \{1,\ldots,N-1\}} \left(\max_{j \in \{i+1,\ldots,N\}} (R_{j,i} - R_{i,i}) \right) \tag{5.10}$$

  where $R_{j,i}$ is the performance on task i after learning task j, and $R_{i,i}$ is the initial performance on task i. Lower FM values indicate less forgetting and better retention of knowledge over time.

5.1.1 Pre-Processing

Signal Reduction The raw sensor data collected from railway accelerometers and strain gauges consists of time series representing acceleration points and strain measurements during train passages. The sensor starts recording upon detecting a new train and stops when the movement ceases.

Consequently, the data acquisition time varies depending on the train's speed. Slower trains generate longer signals with more data points, while faster trains produce compressed signals. The longest recorded acceleration signal has a staggering 166,749 data points, making it computationally expensive to work with directly.

Three variations of the Hilbert transform were explored for dimensionality reduction:

- Original Hilbert Transform with Point Selection: This baseline approach applies the standard Hilbert transform to the raw signal and then selects a specific nth point (e.g., peak amplitude) from the analytical signal for further analysis.

- Hilbert Transform with Moving Average: This variation incorporates a moving average filter on the analytical signal before selecting the nth point.

- Hilbert Transform with Gaussian Filter: Similar to the moving average approach, a Gaussian filter is applied.

All of these transformations are represented in Figure 5.1, where it's possible to see how the representativeness of the signal is affected by each transformation.

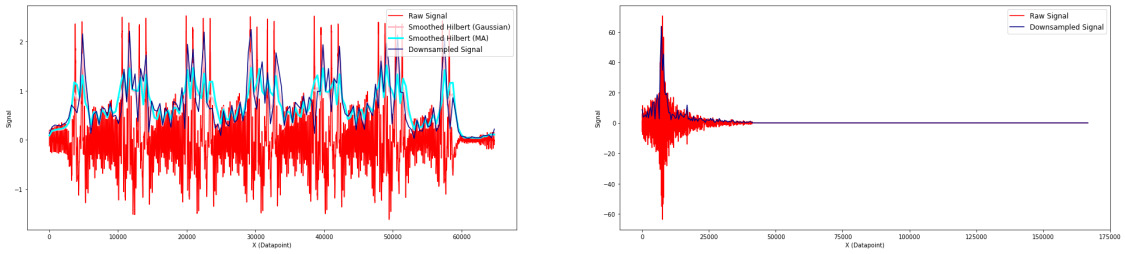


Figure 5.1: Original signal of acceleration with 166749 points in red, downsampled signal with 417 points in blue

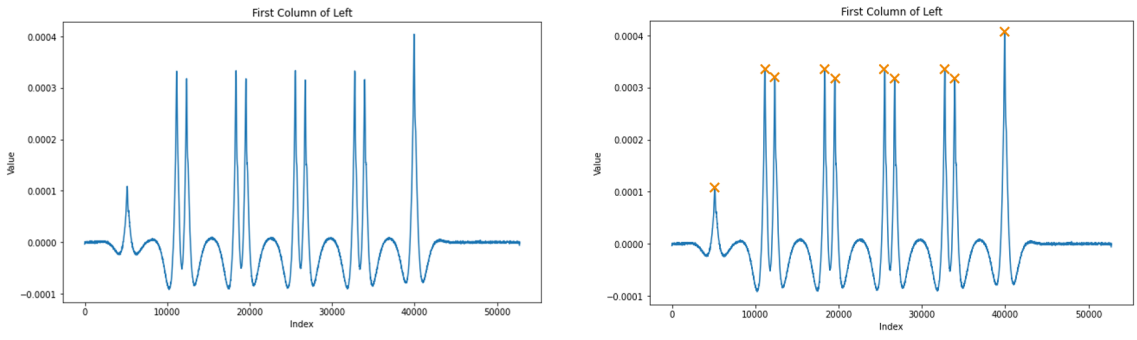
The moving average and the gaussian filter tended to oversoothe the sharp curves of the signal as, making the original Hilbert with a simple selection of the nth point the most representative downsampling of the signal, with n = 400.

Semantics

To obtain the necessary semantics from the dataset, further pre-processing was done, now in the form of an algorithm implemented to extract intrinsic information effectively.

In this case, the factor was in determining the number of wheels on a train in each instance. This was accomplished using data gathered by strain gauges, which measure the strain caused by the weight of the wheels as they pass over the sensor. The raw sensor data is illustrated in Figure 5.2a, showcasing the fluctuations in strain measurements as wheels traverse the sensor.

To identify individual wheels, we applied the *scipy.signal.find_peaks* algorithm to the strain data. The resulting array of detected peaks, represented by the x-axis values, is shown in Figure 5.2b.



(a) Original strain signal

(b) Strain signal with flagged wheel

Figure 5.2: Impact of the algorithm, each cross represents a different wheel registered

By identifying these peaks, we can accurately count the number of wheels on the train, which is crucial for subsequent analysis. This pre-processing step ensures that the model is equipped with precise and relevant information, improving its overall performance and reliability.

5.1.2 Feature Extraction - Embeddings

The next step is to identify what are the most representative features in a single signal, since these are the key elements to be fed into the anomaly classifier, and will improve the performance if handled correctly.

The encoder has one fully-connected linear layer with a ReLU (Rectified Linear Unit) activation function each. The layer has an output dimension of 256. A final fully-connected linear layer with an output dimension of 40 (doubled the latent space dimension). This layer outputs a vector containing both the mean and logarithm of variance.

The next step was to perform a hyperparameter optimization. The goal of this optimization was to minimize the reconstruction loss in a validation dataset (20% of the input data). This was achieved through the variation of 5 parameters (learning rate, latent space dimension, number of epochs, batch size and the KL weight in the loss function) throughout 50 studies.

The optimal scenario found was with a learning rate of 0.0005, latent space of 20, 150 epochs, a batch size of 64 and a weight of 1.412 for the KL Divergence. This model achieved a minimal loss of 683.01, whereas the average loss was 2702.43, therefore resulting in a improvement of 74.73%.

Visualizing the latent space helps in assessing how well the model captures the underlying structure of the data. From the visualization in Figure 5.3, it appears that the model has identified distinct clusters, which is a positive indicator of its ability to differentiate between various operational scenarios or fault conditions in the railway dataset.

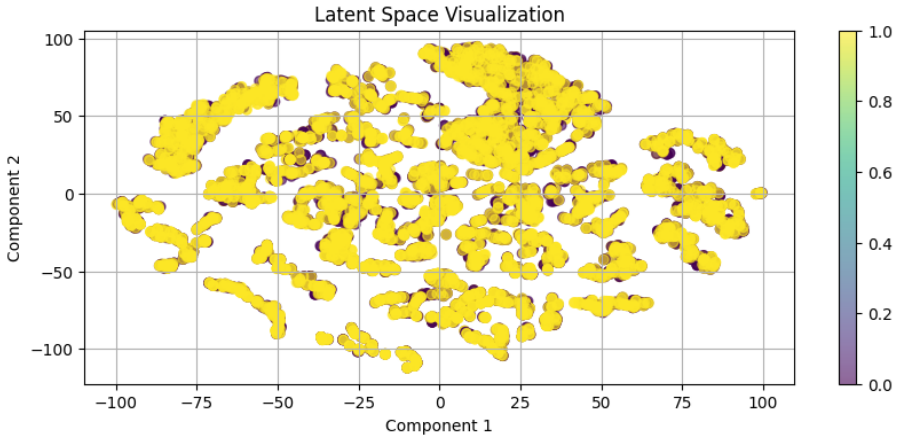


Figure 5.3: Latent Space Visualization

The performance of the improved Variational Autoencoder was further analyzed by observing in a small amount of samples if the reconstructed signal generated by the VAE, when compared to the original signal, held the same representativeness in the data. Some of these samples are the ones in Figure 5.4.

Further on, another experiment was conducted, this time to evaluate the structure of the VAE, if it is the most appropriate and the one that proves the most beneficial to the classifier's end result. For

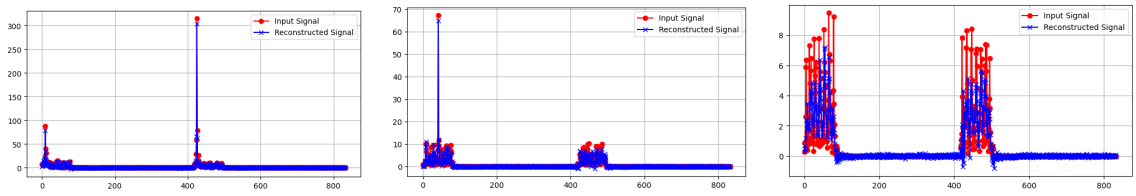
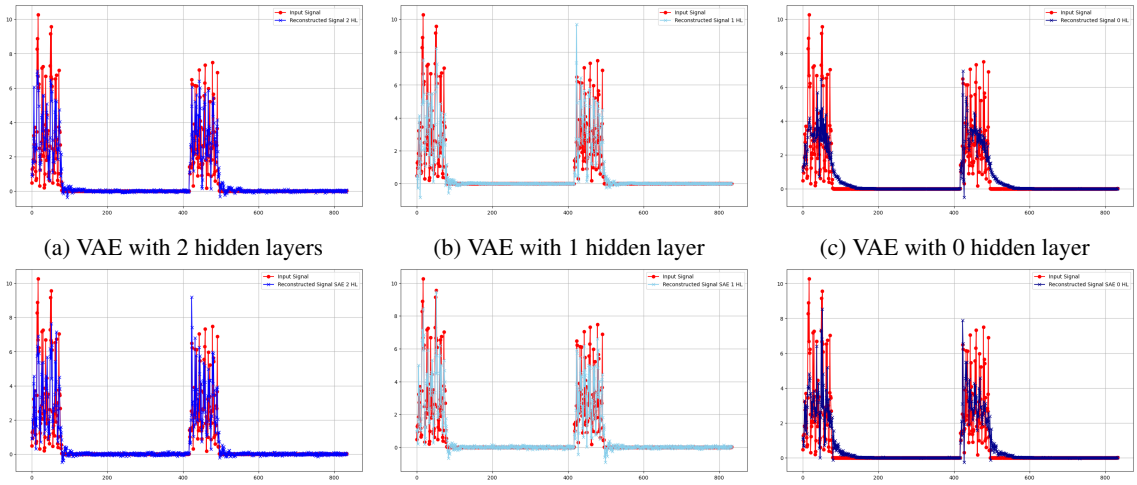


Figure 5.4: Examples of reconstructed signals by VAE

that, we fixed the hyperparameters as the one established in the last experiment as the parameters for all VAEs tested, and the classifier used for predicting testing was optimized XGBosst (more information in Section 5.1.3).

We tested with a VAE with 2 hidden layers (each layer has an output of 256, 64 and 40, respectively), a VAE with 1 hidden layer (each layer has an output of 256 and 40, respectively) and a VAE with 0 hidden layer (output of the initial layer is directly the double of the latent dimension). We also tested a SAE with the exact same architectures as the VAE. For visualization purposes, the reconstructions are presented in Figure 5.5.



(a) VAE with 2 hidden layers (b) VAE with 1 hidden layer (c) VAE with 0 hidden layer

(d) SAE with 2 hidden layers (e) SAE with 1 hidden layer (f) SAE with 0 hidden layer

Figure 5.5: Reconstructed signals by three different VAEs and three different SAEs

The difference between one hidden layer and two is very faint in both types of autoencoder, which shows potential for model (b), since it is more computationally cheap without compromising the performance of the model.

Each of these models created a different dataset that would be the input for the classifier. A baseline experiment was added in order to prove the added value that the feature extraction via embeddings bring. The baseline scenario had a simple feature extraction that instead of learning the latent space, it relies on straightforward statistical and mathematical operations, like mean,

standard, deviation, range, skewness, kurtosis, frequency domain, Fast Fourier Transform (FFT), dominant frequencies, spectral energy, among others. The results are displayed in Table 5.1:

Table 5.1: Classifier performance with different architectures of VAE

	VAE with 2 HL	**VAE with 1 HL**	**VAE with no HL**	**Feature Extractor**	**SAE with 2 HL**	**SAE with 1 HL**	**SAE with no HL**
Accuracy	99%	99%	95%	72%	99%	98%	97%
Recall	99%	99%	94%	74%	99%	98%	97%
Precision	99%	99%	91%	72%	99%	99%	95%
F1 Score	99%	98%	95%	75%	99%	98%	97%

Having estabished that latent spaces are inherently better at creating meaningful relationships in time series data, we then focused on the VAE architectures. Here, the balance between model complexity and performance favored a single hidden layer architecture. The observed correlation between reconstructed data representativeness and prediction performance suggests the suitability of a 1-layer VAE for further investigation.

After all this verification process was completed, it was proved that the VAE was representing the dataset with good quality. Therefore, it was possible to advance to the next step, where a new dataset would be created, this time completely made of embeddings generated from the previous dataset.

After generation, the dataset consisted of instances with 40 data points each. When used, semantics were concatenated to these 40 data points.

5.1.3 Anomaly Detection

In the literature, Neural Networks have been prominently suggested and reviewed as a major algorithm for anomaly detection. This preference is due to their under-constrained nature, which makes them particularly effective in environments with significant variability. Consequently, our initial experiments employed Neural Networks.

The neural network used in the initial experiments was a Binary Classifier with a single Linear layer, designed to output the probability of an instance being an anomaly (anomaly equals 1).

These initial tests were conducted before implementing downsampling and semantic integration. At this stage, the raw signal data consisted of thousands of data points. The embeddings derived from this raw data were not highly significant yet were essential for the process. This necessity arose because feeding such large arrays directly into the classifier would have been computationally infeasible.

During this phase, the pipeline incorporated two classifiers. The first classifier aimed to identify the vehicle, while the second was responsible for anomaly detection. We evaluated two versions of

this pipeline: Version A, this version treated the two classifiers independently, with no interaction between the vehicle identifier and the anomaly detector; version B, where the pipeline followed a sequential process, the output label from the vehicle classifier was provided as an additional input to the anomaly detector, potentially enhancing the anomaly detection performance by leveraging the vehicle identification information.

Table 5.2: Neural Networks performance in Train Classifier and Anomaly Detector

	Version A		**Version B**	
	Anomaly Detector	**Train Classifier**	**Anomaly Detector**	**Train Classifier**
Accuracy	48%	78%	63%	78%
Recall	48%	78%	63%	78%
Precision	23%	83%	52%	83%
F1 Score	31%	77%	56%	77%

However, it very quickly became clear that the current model was very poor, doing mostly guesswork and, on top of that, taking close to 100 minutes to train. This would be unsustainable in the long term. The downsampling methodology was, then, applied (Section 5.1.1). Now that tests became very computationally and timely cheap, it was possible to compare different algorithms and find the most adequate for our process.

In order to choose the algorithm for the classifier, a test was performed using five different algorithms: Neural networks, XGBoost, Logistic Regression, KNeighbors and Decision Trees. The conditions of the test were as followed: all instances of the dataset were used for every experiment, divided by 75% training and 25% test, semantics were not integrated in the experiment and the metrics registered were Accuracy, Recall, Precision and F1 Score.

The results were as followed in Table 5.3 and Table 5.4:

Table 5.3: Classifier Algorithm Experiment without Semantics

	Neural Networks	**XGBoost**	**Logistic Regression**	**KNeighbors**	**Decision Trees**
Accuracy	62%	51%	**52%**	50%	51%
Recall	62%	83%	**92%**	56%	51%
Precision	51%	52%	**53%**	52%	49%
F1 Score	55%	64%	**67%**	54%	45%

After weighting the performance of models with and without semantic information, the decision was to use XGBoost as the primary algorithm. Logistic Regression excelled without semantics, but its performance (relatively to the others) plummed when semantics were included. Decision

Table 5.4: Classifier Algorithm Experiment with Semantics

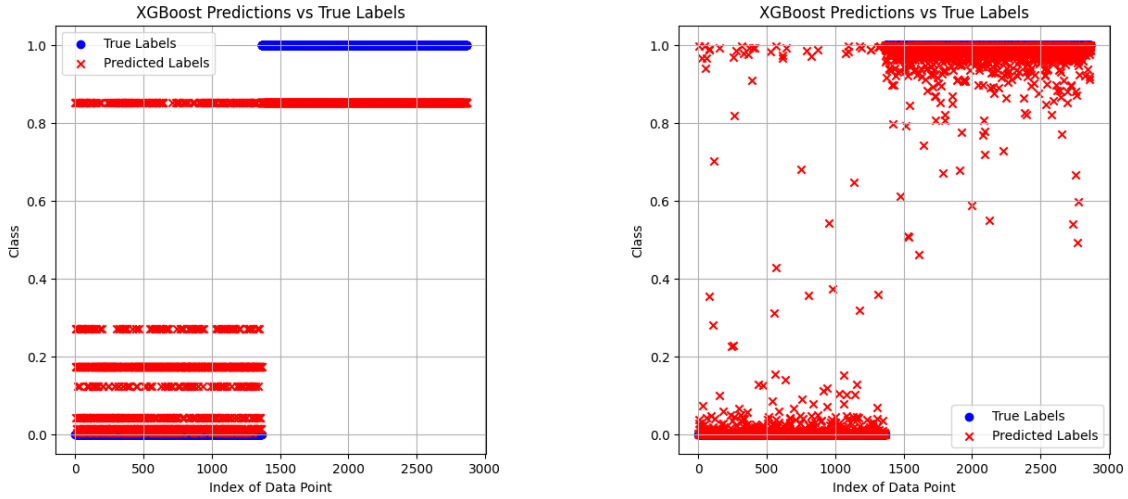
	Neural Networks	**XGBoost**	**Logistic Regression**	**KNeighbors**	**Decision Trees**
Accuracy	65%	**95%**	77%	88%	**95%**
Recall	65%	**100%**	92%	96%	95%
Precision	75%	91%	72%	83%	**95%**
F1 Score	55%	**95%**	80%	89%	**95%**

trees followed a similar pattern, although with reversed performance. Interestingly, XGBoost maintained consistent above-average performance throughout the experiment, suggesting strong potential.

The next step was to further optimize the XGBoost performance. In order to do this, a hyperparameter optimization search was performed, that varied a total of 6 parameters ('alpha', 'colsample_bytree', 'subsample', 'learning_rate', 'max_depth', 'min_child_weight'). After 50 iterations, the best hyperparameters were found, resulting in a improvement of 3.79% overall, as possible to verify in Table 5.5. Figure 5.6 portraits the improvement this process had: predictions are more agglomerated in the correct spot, clearly indicating more confident and accurate predictions.

Table 5.5: XGBoost Performance before and after Hyperparameter Optimization

	XGBoost	**XGBoost Optimized**
Accuracy	95%	**99%**
Recall	**100%**	99%
Precision	91%	**99%**
F1 Score	95%	**98%**



(a) non-optimized algorithm

(b) optimized algorithm

Figure 5.6: Predicted vs. True Labels XGBoost

To further evaluate the performance of our classification model, confusion matrix was used. This tool provides detailed breakdown of the model's predictions compared to the actual outcomes, allowing for a nuanced assessment of its accuracy. This is displayed in Figure 5.7 and indicates that the classifier is highly accurate and effective at detecting anomalies in the railway system, with a perfect recall rate (no false negatives) and a high precision rate (low false positives).

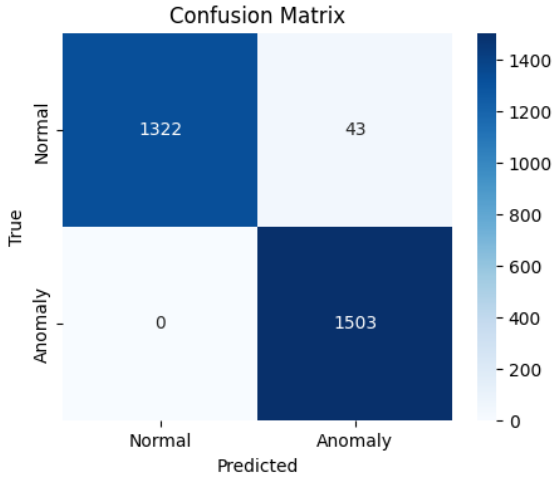


Figure 5.7: Confusion Matrix Classifier

5.1.4 Generalization - Zero Shot Learning

To assess the model's capacity for generalization to unseen data, we implemented a stratified train-test splitting approach that leverages train speeds. This technique guarantees that the training and testing datasets accurately represent the real-world distribution of operational train speeds. Notably, the model will be trained on instances encompassing the entire spectrum of speeds, with

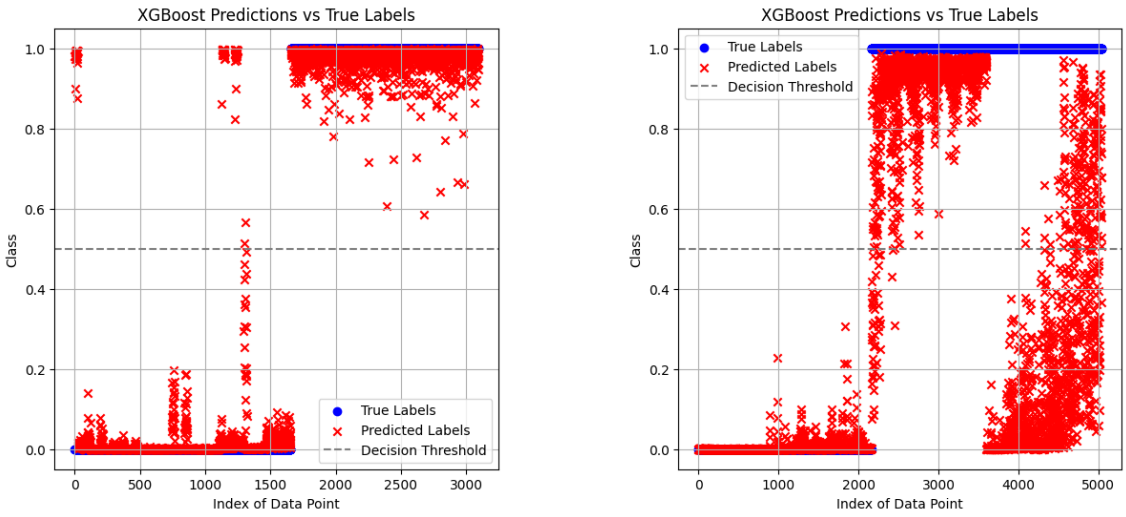
the exception of those belonging to domain 6. Domain 6 represents trains operating at more stable, mid-range velocities. By excluding this domain from training, we ensure the model acquires knowledge of both slower and faster operating regimes. This strategic exclusion empowers the model to effectively generalize its capabilities towards unseen data corresponding to both lower and higher speed domains.

A comparative experiment was conducted using a baseline model integrated within the identical environment to further testify the first experiment. In this one, both models were trained on the same dataset and evaluated on the same occluded test data. However, the baseline model treated unknown instances as the nearest neighbor imputation. This is, when encountering an unknown instance (e.g., a Laagrss vehicle at 90 km/h), the model classified it as the closest known instance within its existing knowledge base (e.g., a 80 km/h Laagrss vehicle).

The results of these experiments are presented in Table 5.6 and Figure 5.8.

Table 5.6: XGBoost Performance on Unseen Speed Distributions

	Baseline	**Generalized**
Accuracy	98%	82%
Recall	98%	82%
Precision	97%	86%
F1 Score	98%	81%



(a) Baseline generalization scenario

(b) Performance in unseen speed scenarios

Figure 5.8: XGBoost predictions with and without generalization

Our experiments proves that getting the most similar model does perform better than an attempt at generalizing based on known tasks. This is due to the fact this dataset includes a small amount of different tasks, which hinders the generalization capacity of the model. According to the literature, zero-shot learning will create better results as more complex and detailed is the dataset. Potentially,

if we were to increase the number of types of vehicles we would eventually find a break-even point where zero-shot is a more interesting approach than the baseline.

5.1.5 Experience Replay

This section describes the experimental conditions and methodologies employed to evaluate the forgetting phenomenon in our machine learning model. Three different strategies were employed to manage and update the training data: Baseline, Reservoir Sampling, and Loss-Based Sampling. The experiments aimed to investigate the model's performance over successive training sessions, capturing metrics such as accuracy, recall, precision, and F1 score.

Scenario number 1: Memory sampler of Reservoir Sampling of size 800. Samples are selected and stored based on their true labels.

Scenario number 2: Memory sampler of Reservoir Sampling of size 800. Samples are selected and stored based on the predicted labels from the model. This helps the model focus on challenging and informative samples, efficiently manage memory constraints, adapt to evolving data distributions, and ultimately improve robustness and generalization. This approach is particularly beneficial in real-world applications with imbalanced data, limited memory, and changing environments.

Scenario number 3: Loss-based memory sampler of size 800. Samples are selected and stored based on its associated loss value when compared to the true labels. This method aims to maintain a high model performance by keeping samples that are harder to predict correctly, thus minimizing forgetting and improving generalization.

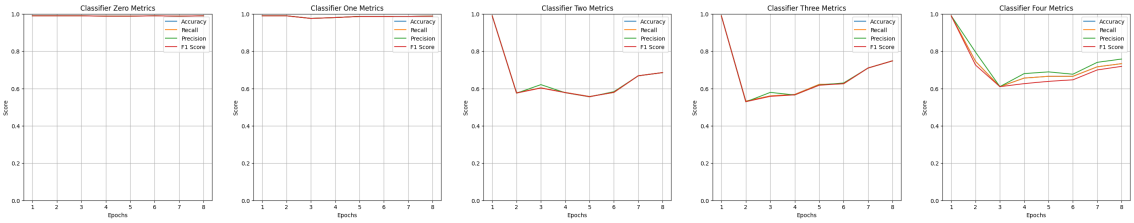
Scenario number 4: Loss-based memory sampler of size 800. Samples are selected and stored based on its associated loss value when compared to the true labels, but the predicted value is stored.

Scenario number 0: The model is retrained scratch using the entire dataset every time new data arrives, therefore, it is integrated in the same environment with new data flow.

The experiment employed a staged training approach to evaluate the model's susceptibility to catastrophic forgetting. The initial training phase exposed the model solely to winter scenarios, characterized by trains operating at very low speeds. Subsequent training sessions, spanning seven more iterations, progressively introduced scenarios with increasing speeds and varying cargo loads. These scenarios ranged from medium speeds with high cargo loads to faster speeds with low cargo loads. Notably, the winter scenario was never reintroduced after the initial training phase.

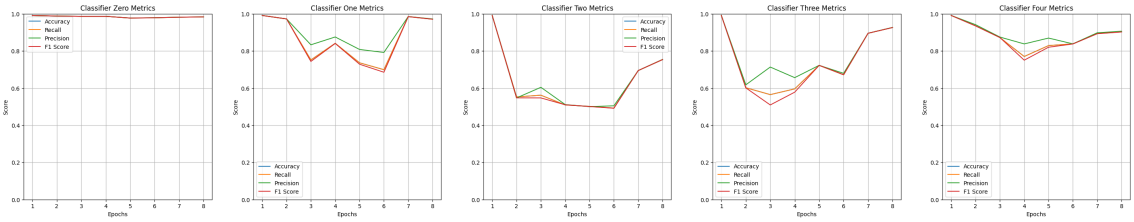
Following each training session, the model's performance on the initial winter scenario was evaluated to assess its ability to retain knowledge in the face of exposure to progressively more diverse training data. This design aimed to simulate real-world situations where systems encounter new

data distributions while potentially forgetting previously learned information. The results of performance throughout the different iterations and different scenarios are presented in Figures 5.9 and 5.10.



(a) Scenario 0 (b) Scenario 1 (c) Scenario 2 (d) Scenario 3 (e) Scenario 4

Figure 5.9: Memory size of 800



(a) Scenario 0 (b) Scenario 1 (c) Scenario 2 (d) Scenario 3 (e) Scenario 4

Figure 5.10: Memory size of 200

Our findings indicate that while Reservoir Sampling performance tends to degrade with reduced memory size due to its non-selective nature, Loss-Based Sampling methods exhibit superior performance under the same conditions. The Loss-Based Samplers leverage their ability to prioritize more significant samples, which helps in maintaining model performance even with limited memory. This adaptive sampling strategy ensures that the most informative and challenging samples are retained, thereby enhancing the model's learning efficiency and robustness in resource-constrained environments.

A subsequent experiment employed the same scenarios but adopted a different evaluation approach. Instead of measuring performance on a dedicated test set, this experiment focused on annotating the model's performance across all previously encountered tasks. This way, we could observe metrics such as FWT, IM, KGR, BWT, presented in Tables 5.7 and 5.8:

Table 5.7: Experience Replay impact on model performance with memory size 200

	Baseline (Zero)	**Scenario One**	**Scenario Two**	**Scenario Three**	**Scenario Four**
Forward Transfer	-0.0109	0.0934	0.1882	0.1492	0.3122
Backward Transfer	-0.0117	-0.0129	-0.1170	-0.0077	-0.1164
Intransigence Measure	0.0104	0.1086	0.1990	0.2192	0.3637
Knowledge Gain Ratio	3.1971	2.8668	2.5662	2.4983	2.0126

Table 5.8: Experience Replay impact on model performance with memory size 800

	Baseline (Zero)	**Scenario One**	**Scenario Two**	**Scenario Three**	**Scenario Four**
Forward Transfer	-0.0102	0.0377	0.0947	0.2427	0.1414
Backward Transfer	-0.0038	-0.0219	-0.1136	-0.0018	-0.1240
Intransigence Measure	0.0111	0.0628	0.1031	0.3038	0.1864
Knowledge Gain Ratio	3.2005	3.0277	2.8927	2.2169	2.6144

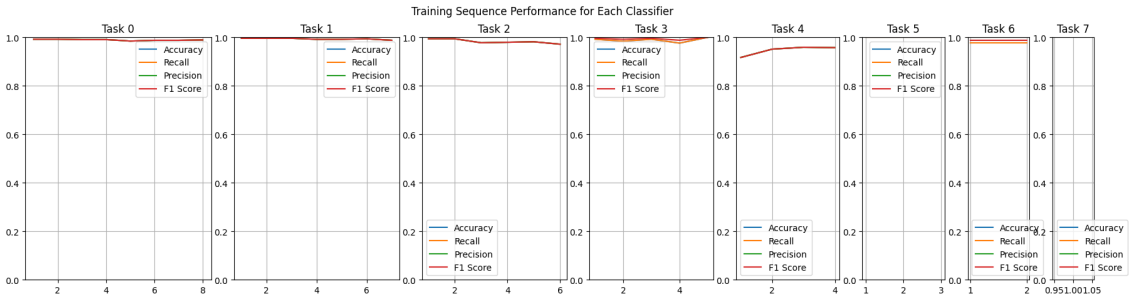


Figure 5.11: Performance throughout different tasks scenario baseline

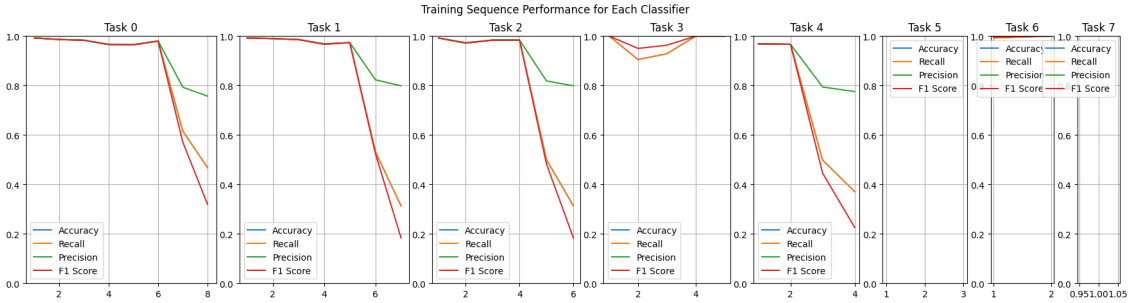


Figure 5.12: Performance throughout different tasks scenario zero

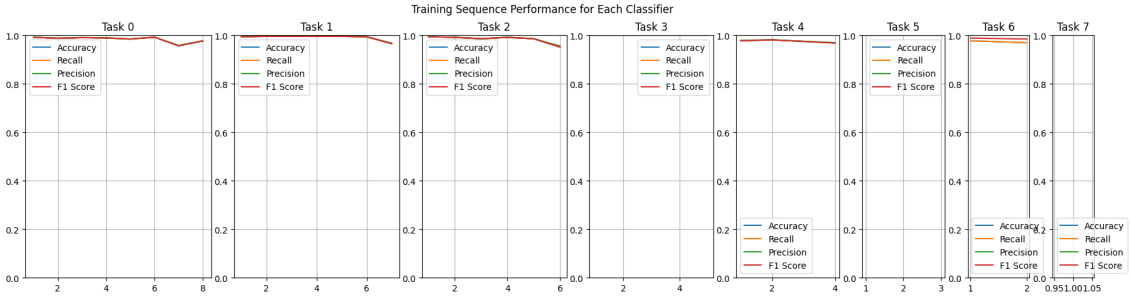


Figure 5.13: Performance throughout different tasks scenario one

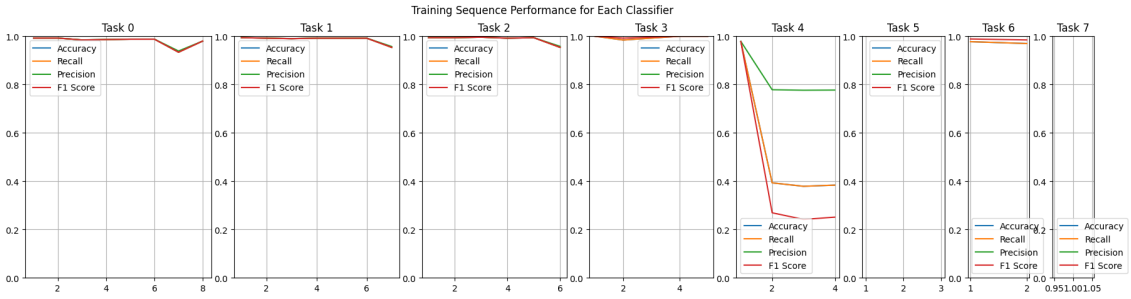


Figure 5.14: Performance throughout different tasks scenario two

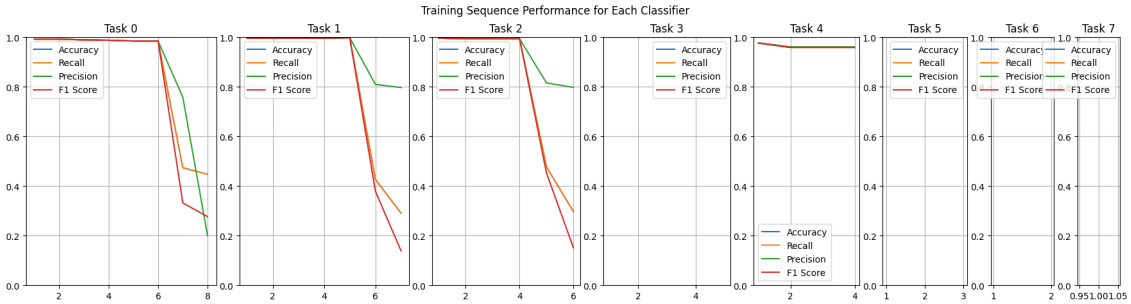


Figure 5.15: Performance throughout different tasks scenario three

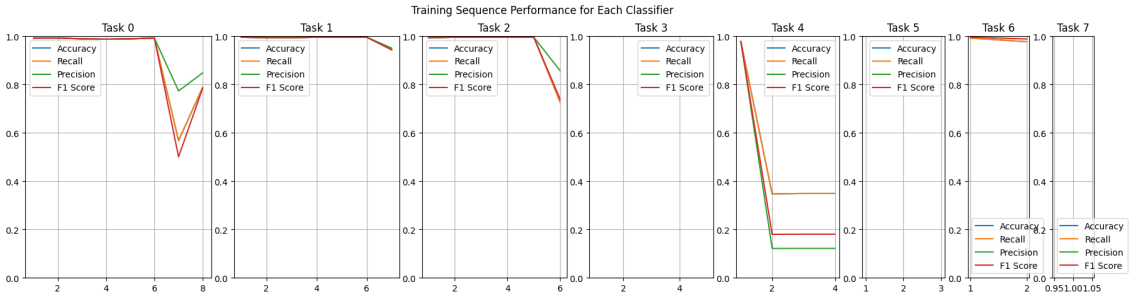


Figure 5.16: Performance throughout different tasks scenario four

These results are in agreement with the previous experiment. Increasing the memory size generally improves the model's ability to transfer knowledge forward and reduces the negative impact on

previously learned tasks. Scenarios Two and Four demonstrate significant forward learning but also highlight issues with forgetting, suggesting a need for targeted strategies to improve retention. Scenario Three benefits the most from a larger memory buffer, indicating a well-balanced learning and retention process.

In Chapter 5, the effectiveness of the proposed AI models was rigorously validated using a series of experiments and performance evaluations. The models demonstrated high accuracy and robustness in detecting anomalies, surpassing traditional methods, particularly in their ability to handle complex and noisy data. The discussion also highlighted the challenges encountered during the implementation, such as computational demands and data heterogeneity, and proposed solutions to mitigate these issues. This chapter not only confirmed the feasibility of the proposed models in a real-world setting but also reinforced the importance of continual learning and model adaptability in maintaining high performance over time.

Chapter 6

Conclusions and Project's prospects

The goals of the thesis were achieved with success. The baseline model (an anomaly detector that takes a processed signal, encodes it into embeddings, augments it with semantics and finally classifies it as either a normal instance or anomalous one) was constructed to be a strong and robust cornerstone for the experiments that took place upon the same model, such as the generalization and continual learning.

The main achievements from this project can be presented as:

- The proposed method begins with a novel signal segmentation technique that leverages the sensitivity of each type of sensor. The stable temporal structure of the strain gauge signal is used to segment both strain gauge and accelerometer signals. Feature extraction is primarily focused on the accelerometer, which is more sensitive to damage and less affected by non-linear operational and environmental factors.
- To enhance the model's understanding of the data, a Variational Autoencoder (VAE) was employed to encode the segmented time-series data into a latent space. This compressed latent representation captures the underlying structure of the data, facilitating more effective anomaly detection.
- The latent representation is augmented with semantic information, such as the number of wheels on a train, which is derived from the strain gauge data using peak detection algorithms. This enrichment of the latent space ensures that the model has a comprehensive understanding of the contextual information necessary for accurate fault detection..
- The methodology is designed to minimize the need for extensive sensor installations. By effectively utilizing data from a single accelerometer and a strain gauge, the approach reduces installation and maintenance costs without compromising data quality.
- To address the challenge of continual learning and generalization, the system incorporates an Experience Replay (ER) buffer. This buffer retains both new and significant historical

data, including anomalies, and integrates this information into the retraining process. This approach helps in refining the model's accuracy over time and ensures that it can adapt to new data while retaining knowledge of past events.

- In the final stage, the enriched latent representation was fed into a binary classifier for anomaly detection, which exhibited great performances overall.

This was an innovative approach that showed the potential to revolutionize railway maintenance practices, offering real-time, accurate anomaly detection, and ultimately enhancing the safety and efficiency of railway operations, designed with real-world applications in mind, for real-world applications.

The primary challenge encountered during the development of this thesis was related to the dataset. Ideally, the dataset should have encompassed a wider range of scenarios, different types of trains, and varied operational characteristics to better emulate real-world conditions. Additionally, it needed to be more consistent and uniform. However, simulating these diverse data instances is a complex and demanding task, as described in Chapter three, resulting in an incomplete dataset. This limitation hindered the ability to fully demonstrate the model's capacity and, consequently, the potential added value of the proposed solution.

Despite these challenges, the project has garnered attention and has been selected for further development under GECAD, with enhanced support from data managers. This continued development is expected to address the current limitations and significantly improve the project's outcomes.

Looking ahead, several interesting concepts could be explored to enhance the model's performance and generalizability. One such concept is the implementation of task-specific classifiers, which would have dedicated parameters for each data distribution. This approach could improve zero-shot learning and enhance the model's ability to generalize across different tasks.

Additionally, leveraging the generative capabilities of the Variational Autoencoder (VAE) presents another promising direction. By exploring new instances and classes generated by the VAE, it would be possible to enrich the dataset and further integrate these generated instances with zero-shot learning techniques. This could potentially lead to a more robust and versatile model capable of handling a wider variety of real-world scenarios.

6.1 Contributions

Online continual learning for railway wheels fault detection

Authors: Francisca Osório, Afonso Lourenço, Diogo Ribeiro and Goreti Marreiros

Ongoing work, to be submitted to Railway Engineering Science

Stress-based vehicle wheel detector and axle count: from strain gauges to fiber Bragg gratin

Authors: Francisca Osório, Afonso Lourenço, Diogo Ribeiro and Goreti Marreiros

Ongoing work, to be submitted to Railway Engineering Science

Bibliography

Adel, T., Zhao, H., and Turner, R. E. (2020). Continual learning with adaptive weights (claw).

Ahmed, F. and Courville, A. (2020). Detecting semantic anomalies. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3154–3162.

Araliya Mosleh, Andreia Meixedo, D. R. P. M. and Calçada, R. (2023). Early wheel flat detection: an automatic data-driven wavelet-based approach for railways. *Vehicle System Dynamics*, 61(6):1644–1673.

Bai, L., Liu, R., Sun, Q., Wang, F., and Xu, P. (2015). Markov-based model for the prediction of railway track irregularities. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 229(2):150–159.

Caccia, M., Mueller, J., Kim, T., Charlin, L., and Fakoor, R. (2023). Task-agnostic continual reinforcement learning: Gaining insights and overcoming challenges.

Cai, W., Chi, M., Tao, G., Wu, X., and Wen, Z. (2019). Experimental and numerical investigation into formation of metro wheel polygonalization. *Shock and Vibration*, 2019(1):1538273.

Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H. S. (2018). *Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence*, page 556–572. Springer International Publishing.

Dwyer-Joyce, R., Yao, C., Lewis, R., and Brunskill, H. (2013). An ultrasonic sensor for monitoring wheel flange/rail gauge corner contact. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 227:188–195.

Falamarzi, A., Moridpour, S., and Nazem, M. (2019). Development of a tram track degradation prediction model based on the acceleration data. *Structure and Infrastructure Engineering*, 15.

Gao, R., He, Q., and Feng, Q. (2019). Railway wheel flat detection system based on a parallelogram mechanism. *Sensors*, 19(16).

Gao, S., Kang, G., Yu, L., Zhang, D., Wei, X., and Zhan, D. (2022). Adaptive deep learning for high-speed railway catenary swivel clevis defects detection. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):1299–1310.

Gonçalves, V., Mosleh, A., Vale, C., and Montenegro, P. A. (2023). Wheel out-of-roundness detection using an envelope spectrum analysis. *Sensors*, 23(4).

Gómez, M. J., Corral, E., Castejón, C., and García-Prada, J. C. (2018). Effective crack detection in railway axles using vibration signals and wpt energy. *Sensors*, 18(5).

Hertz, H. (1882). Ueber die berührung fester elastischer körper. *Journal für die reine und angewandte Mathematik*, 1882(92):156–171.

Higgins, C. and Liu, X. (2017). Modeling of track geometry degradation and decisions on safety and maintenance: A literature review and possible future research directions. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 232:095440971772187.

Hofer, V. and Bach, H. (2015). Statistical monitoring for continual quality control of railway ballast. *Expert Systems with Applications*, 42(22):8557–8572.

Hu, W., Lin, Z., Liu, B., Tao, C., Tao, Z., Ma, J., Zhao, D., and Yan, R. (2019). Overcoming catastrophic forgetting via model adaptation. In *International Conference on Learning Representations*.

J. N. Costa, J. Ambrósio, D. F. and Andrade, A. R. (2022). A multivariate statistical representation of railway track irregularities using arma models. *Vehicle System Dynamics*, 60(7):2494–2510.

Jamal, M. A., Qi, G.-J., and Shah, M. (2018). Task-agnostic meta-learning for few-shot learning.

Johansson, A. and Andersson, C. (2005). Out-of-round railway wheels - a study of wheel polygonalization through simulation of three-dimensional wheel-rail interaction and wear. *Vehicle System Dynamics*, 43(8):539–559.

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

Kamlu, S. and Laxmi, V. (2019). Condition-based maintenance strategy for vehicles using hidden markov models. *Advances in Mechanical Engineering*, 11(1):1687814018806380.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Lam, H. F., Yang, J.-H., hu, Q., and Ng, C.-T. (2017). Railway ballast damage detection by markov chain monte carlo-based bayesian method. *Structural Health Monitoring: An International Journal*, 17:147592171771710.

Lasisi, A. and Attoh-Okine, N. (2018). Principal components analysis and track quality index: A machine learning approach. *Transportation Research Part C: Emerging Technologies*, 91:230–248.

Lederman, G., Chen, S., Garrett, J., Kovačević, J., Noh, H. Y., and Bielak, J. (2017). Track-monitoring from the dynamic response of an operational train. *Mechanical Systems and Signal Processing*, 87:1–16.

Li, Z., Shang, D., Ma, Z., Liu, Y., and Lu, Y. (2023). Memory mechanisms based few-shot continual learning railway obstacle detection. In *2023 China Automation Congress (CAC)*, pages 9372–9377.

Liang, B., Iwnicki, S., Ball, A., and Young, A. (2015). Adaptive noise cancelling and time-frequency techniques for rail surface defect detection. *Mechanical Systems and Signal Processing*, 54-55:41–51.

Liang, B., Iwnicki, S., Zhao, Y., and Crosbee, D. (2013). Railway wheel-flat and rail surface defect modelling and analysis by timeâ€“frequency techniques. *Vehicle System Dynamics*, 51.

Liu, C., He, S., Liu, H., Chen, J., and Dong, H. (2024). Windtrans: Transformer-based wind speed forecasting method for high-speed railway. *IEEE Transactions on Intelligent Transportation Systems*, 25(6):4947–4963.

Liu, X.-Z. and Ni, Y.-Q. (2018). J-2018-sss-wheel tread defect detection for high-speed trains using fbg-based online monitoring techniques. *Smart Structures and Systems*, 21:687–694.

Lomonaco, V., Maltoni, D., and Pellegrini, L. (2020). Rehearsal-free continual learning over small non-i.i.d. batches.

Lopez-Paz, D. and Ranzato, M. (2022). Gradient episodic memory for continual learning.

Lourenço, A., Meira, J., and Marreiros, G. (2023). Online adaptive learning for out-of-round railway wheels detection. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, SAC '23, page 418–421, New York, NY, USA. Association for Computing Machinery.

Lourenço, A., Fernandes, M., Canito, A., Almeida, A., and Marreiros, G. (2022). Using an explainable machine learning approach to minimize opportunistic maintenance interventions. In González-Briones, A., Almeida, A., Fernandez, A., El Bolock, A., Durães, D., Jordán, J., and Lopes, F., editors, *Highlights in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation. The PAAMS Collection*, pages 41–54, Cham. Springer International Publishing.

Lourenço, A., Ferraz, C., Ribeiro, D., Mosleh, A., Montenegro, P., Vale, C., Meixedo, A., and Marreiros, G. (2023). Adaptive time series representation for out-of-round railway wheels fault diagnosis in wayside monitoring. *Engineering Failure Analysis*, 152:107433.

Lourenço, A., Ferraz, C., Meira, J., Marreiros, G., Bolón-Canedo, V., and Alonso-Betanzos, A. (2023). Automated green machine learning for condition-based maintenance. pages 291–296.

Mainsant, M., Mermillod, M., Godin, C., and Reyboz, M. (2022). A study of the dream net model robustness across continual learning scenarios. In *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 824–833.

Mohammadi, M., Mosleh, A., Vale, C., Ribeiro, D., Montenegro, P., and Meixedo, A. (2023). An unsupervised learning approach for wayside train wheel flat detection. *Sensors*, 23(4).

Molodova, M., Li, Z., Núñez, A., and Dollevoet, R. (2014). Automatic detection of squats in railway infrastructure. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):1980–1990.

Molodova, M., Oregui, M., Núñez, A., Li, Z., and Dollevoet, R. (2016). Health condition monitoring of insulated joints based on axle box acceleration measurements. *Engineering Structures*, 123:225–235.

Montenegro, P., Neves, S., Calçada, R., Tanabe, M., and Sogabe, M. (2015). Wheel-rail contact formulation for analyzing the lateral trainâ€“structure dynamic interaction. *Computers Structures*, 152:200–214.

Mosleh, A., Costa, P. A., and Calçada, R. (2020a). A new strategy to estimate static loads for the dynamic weighing in motion of railway vehicles. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 234(2):183–200.

Mosleh, A., Montenegro, P. A., Alves Costa, P., and Calçada, R. (2020b). An approach for wheel flat detection of railway train wheels using envelope spectrum analysis. *Structure and Infrastructure Engineering*.

Mosleh, A., Montenegro, P. A., Costa, P. A., and Calçada, R. (2021a). Railway vehicle wheel flat detection with multiple records using spectral kurtosis analysis. *Applied Sciences (Switzerland)*, 11.

Mosleh, A., Montenegro, P. A., Costa, P. A., and Calçada, R. (2021b). Railway vehicle wheel flat detection with multiple records using spectral kurtosis analysis. *Applied Sciences*, 11(9).

Parisi, G. I. and Lomonaco, V. (2020). *Online Continual Learning on Sequences*, page 197–221. Springer International Publishing.

Parisi, S., Dean, V., Pathak, D., and Gupta, A. (2021). Interesting object, curious agent: Learning task-agnostic exploration.

Raz, O., Koopman, P., and Shaw, M. (2002). Semantic anomaly detection in online data sources. In *Proceedings of the 24th International Conference on Software Engineering*, ICSE '02, page 302–312, New York, NY, USA. Association for Computing Machinery.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. (2017). icarl: Incremental classifier and representation learning.

Salvador, P., Naranjo, V., Insa, R., and Teixeira, P. (2016). Axlebox accelerations: Their acquisition and time-frequency characterisation for railway track monitoring purposes. *Measurement*, 82:301–312.

Schenkendorf, R., Dutschk, B., Lüddecke, K., and Groos, J. (2016). Improved railway track irregularities classification by a model inversion approach.

Seff, A., Beatson, A., Suo, D., and Liu, H. (2017). Continual learning in generative adversarial nets.

Sharma, S., Cui, Y., He, Q., Mohammadi, R., and Li, Z. (2018). Data-driven optimization of railway maintenance for track geometry. *Transportation Research Part C: Emerging Technologies*, 90:34–58.

Shin, H., Lee, J. K., Kim, J., and Kim, J. (2017). Continual learning with deep generative replay.

Shmelkov, K., Schmid, C., and Alahari, K. (2017). Incremental learning of object detectors without catastrophic forgetting.

Sophie Mercier, C. M.-H. and Roussignol, M. (2012). Bivariate gamma wear processes for track geometry modelling, with application to intervention scheduling. *Structure and Infrastructure Engineering*, 8(4):357–366.

Sugiyama, H., Araki, K., and Suda, Y. (2009). On-line and off-line wheel/rail contact algorithm in the analysis of multibody railroad vehicle systems. *Journal of Mechanical Science and Technology*, 23:991–996.

T. Jorge, J. Magalhães, R. S. A. G. D. R. C. V. A. M. A. M. P. M. and Cury, A. (2024). Early identification of out-of-roundness damage wheels in railway freight vehicles using a wayside system and a stacked sparse autoencoder. *Vehicle System Dynamics*, 0(0):1–26.

Vrignat, P., Avila, M., Duculty, F., and Kratz, F. (2015). Failure event prediction using hidden markov model approaches. *IEEE Transactions on Reliability*, 64(3):1038–1048.

Wei, J., Liu, C., Ren, T., Liu, H., and Zhou, W. (2017). Online condition monitoring of a rail fastening system on high-speed railways based on wavelet packet analysis. *Sensors*, 17(2).

Wiewel, F. and Yang, B. (2019). Continual learning for anomaly detection with variational autoencoder. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3837–3841.

Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., and Fu, Y. (2019). Large scale incremental learning.

Yoon, J., Yang, E., Lee, J., and Hwang, S. J. (2018). Lifelong learning with dynamically expandable networks.

Zenke, F., Poole, B., and Ganguli, S. (2017). Continual learning through synaptic intelligence.

Zhai, W., Wang, Q., Lu, Z., and Wu, X. (2001). Dynamic effects of vehicles on tracks in the case of raising train speeds. *Proceedings of The Institution of Mechanical Engineers Part F-journal of Rail and Rapid Transit - PROC INST MECH ENG F-J RAIL R*, 215:125–135.

Zhao, B., Xiao, X., Gan, G., Zhang, B., and Xia, S. (2019). Maintaining discrimination and fairness in class incremental learning.