Towards robust real-time condition monitoring and fault diagnosis for railway assets

vorgelegt von
Dipl.-Ing.-
Dachuan Shi

an der Fakultät V – Verkehrs- und Maschinensysteme
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr.-Ing.habil. Sandra Klinge
Gutachter: Prof. Dr.-Ing. Markus Hecht
Gutachter: Prof. Dr.-Ing. Thomas B. Siefer

Tag der wissenschaftlichen Aussprache: 3. November 2022

Berlin 2023

Acknowledgments

My research work involved in the present cumulative dissertation is funded by the EU Shift2Rail projects Assets4Rail (Grand number: 826250) and INNOWAG (Grand number: 730863) under Horizon 2020 Framework Programme. The research data related to vibration monitoring on freight wagons come from the previous projects of Chair of Rail Vehicles TUB.

I would like to thank my supervisor Prof. Dr.-Ing. Markus Hecht for the great support of research resources and the freedom to pursue my research interests. Prof. Hecht has a "railway Wikipedia" in his mind, having extraordinary knowledge and understanding of everything within the railway system. His expertise helped me and guided me to become a railway engineer. I also thank Prof. Dr.-Ing. Thomas B. Siefer from TU Braunschweig for the appraisal of my dissertation.

I thank my student assistant Marco Gillwald for the technical support. The research work related to data processing and machine learning is a trial-and-error process. Marco can always implement my ideas and support me in experiments and error analysis. I thank Shiping Dongfang and Yunguang Ye for the insightful discussions and technical support related to vehicle dynamics. Shiping has not only extraordinary expertise in railway dynamics but also broad knowledge and great passion about the entire railway system. Every time I need help with railways, I can always rely on Shiping, who gives me a very detailed explanation in an easy-to-understand way. Yunguang has not only supported me with his strong expertise of vehicle dynamics in the project tasks but also for the scientific publications. I have learned a lot from his impressive skill and knowledge of scientific writing. Also, a great thank goes to Qiuyong Tian and Yichang Zhou not only for professional exchange but also for the great help in daily work and in life. Furthermore, I thank my dear colleagues at Chair of Rail Vehicle, Jenny Böhm, Yasmin Baumgärtel, Carl Culemann, Ulrich Deghela, Matthias Gülker, Gernoth Götz, Aline Hosang, Thilo Hanisch, Dirk Itzeck, Daniel Jobstfinke, Harald Jäkel, Gökhan Katmer, Philipp Krause, Mirko Leiste, Florian Peche, Márton Pálinkó, Max Schischkoff, and Jonas Vuitton. I really enjoyed the relaxed working atmosphere with them, learned a lot of professional knowledge from them, and appreciated their reliability and responsibility towards work. Especially, Jonas, Aline and Thilo have helped me a lot in my project work.

I also thank the project partners, Daniel Rupp, Klaus Gradischek, Eldar Sabanovic, Luca Rizzetto, Roberto Oliverio, Gintautas Bureika, Stefano Ricci, for smooth cooperation and realizing the field tests in Lithuania and Italy, where the research data in my third publication come from.

Last, but most importantly, I would like to take this opportunity to express my greatest thanks to my wife Qian Ruan. Without her, I may not find my interest in the current research topic. Without her, I cannot learn more about myself and improve myself. She always encourages me, when I am struggling with self-doubt. She makes my life happy and full of fun. Also, I would like to thank my parents. Without their support, I would never have had the opportunity to explore the world and start my studies and work in Germany.

Abstract

A paradigm shift towards condition-based and predictive maintenance (CBM and PM) is undergoing in the railway system. This will increase the maintenance efficiency and ultimately increase the reliability and availability of railway assets. As an essential part of CBM and PM, condition monitoring along with intelligent data processing algorithms determines the up-to-date asset conditions to support maintenance decision-making.

Condition monitoring in railway applications usually requires real-time data processing for fault diagnosis. As train drivers or infrastructure operators should be immediately informed, once severe failures are detected. As many railway assets have not been electrified, the onsite infrastructure for power supply and data communication is absent. This results in further challenges for data processing with regard to power consumption and computational complexity. Furthermore, the operating conditions of railway assets vary in a large range. Condition variations reflect in the monitoring data and cause the distribution shift, which may induce the robustness problem of diagnostic models.

To cope with these problems encountered in practice, we have conducted extensive research towards robust real-time condition monitoring and fault diagnosis for railway assets. We propose to use lightweight convolutional neural networks (LCNN) to realize real-time capability. During the model training, data augmentation is introduced for robustness enhancement. This general data processing procedure is demonstrated in two distinct railway applications. They have been described in three scientific publications, which constitute the present cumulative dissertation.

The first application deals with wheel flat detection (WFD) for vibration monitoring on freight wagons, which supports wagon maintenance. As freight wagons are not electrified, the algorithm should be executed in real time on embedded systems powered by batteries, which have limited computation power. In the first paper, we propose to automatically search a one-dimensional (1D) LCNN for real-time WFD with the optimal tradeoff between computational complexity and detection accuracy. In the second paper, the robustness problems induced by the variation of vehicle speeds, monitored wagons and track conditions are investigated. A novel data augmentation framework, incorporating multibody dynamic simulation for physical modeling and fast weighted feature-space averaging to augment simulation data, is proposed for robustness enhancement.

The second application concerns track geometry monitoring, supporting track maintenance. Our work published in the third paper is the first attempt to use a deep learning based computer vision solution for track geometry monitoring. Virtual point tracking for real-time target-less dynamic displacement measurement is proposed to track the lateral movement of the wheel on the rail, in order to calculate track alignment. It is mainly realized by a 2D LCNN for virtual point detection within each video frame, achieving frame rates of above 30 frames per second on edge devices. In addition, data augmentation based on image corruption is applied to enhance the robustness against different weather conditions and contaminations.

Real-time requirements and robustness problems are the general topics of condition monitoring. The proposed methods for real-time data processing and robustness enhancement are not confined to the two exemplary applications. They can be adapted into similar scenarios. The potential for further improvement are discussed.

Zusammenfassung

Im Eisenbahnsystem vollzieht sich ein Paradigmenwechsel zu zustandsbasierter und prädiktiver Instandhaltung. Dies zielt die Erhöhung der Wartungseffizienz und letztendlich der Zuverlässigkeit und Verfügbarkeit von Eisenbahnanlagen ab. Als wesentlicher Bestandteil ermittelt die Zustandsüberwachung mit intelligenten Datenverarbeitungsalgorithmen den aktuellen Zustand von Anlagen zur Unterstützung effizienter Instandhaltungsentscheidungen.

Die Zustandsüberwachung erfordert in der Regel die Echtzeitverarbeitung der erfassten Daten zur Fehlerdiagnose. Da die Lokführer oder die Infrastrukturbetreiber sofort informiert werden sollten, sobald schwerwiegende Störungen festgestellt werden. Viele Bahnanlagen sind noch nicht elektrifiziert, die Infrastruktur für Stromversorgung und Datenkommunikation fehlt. Daraus ergeben sich weitere Herausforderungen für die Datenverarbeitung hinsichtlich Stromverbrauch und Rechenaufwand. Darüber hinaus variieren die Betriebsbedingungen von Eisenbahnanlagen in einem großen Bereich. Variationen von Betriebsbedingungen verursachen deren Verteilungsverschiebung der Überwachungsdaten, was das Robustheitsproblem der Diagnosealgorithmen hervorrufen kann.

Um diese Probleme zu bewältigen, wurden umfangreiche Untersuchungen zur Realisierung robuster Echtzeit-Zustandsüberwachung und Fehlerdiagnose für Eisenbahnanlagen durchgeführt. Ein auf leichten faltenden neuronalen Netzen (LCNN) basierender Ansatz wird zur Datenverarbeitung in Echtzeit vorgestellt. Dazu werden Datenvermehrungstechniken zur Verbesserung der Robustheit eingesetzt. Dieses allgemeine Datenverarbeitungsverfahren wird in zwei Eisenbahnanwendungen demonstriert. Sie wurden in drei wissenschaftlichen Publikationen beschrieben, die die vorliegende kumulative Dissertation darstellen.

Die erste Anwendung befasst sich mit der Flachstellenerkennung an Güterwagen. Da Güterwagen nicht elektrifiziert sind, sollte der Algorithmus in Echtzeit auf batteriebetriebenen eingebetteten Systemen ausgeführt werden, die über eine begrenzte Rechenleistung verfügen. In der ersten Publikation wurde ein eindimensionales (1D) LCNN zur Flachstellenerkennung hinsichtlich des optimalen Kompromiss zwischen Rechenkomplexität und Erkennungsgenauigkeit entwickelt. In der zweiten Publikation wurde die Robustheit der Flachstellenerkennung durch die Variation von Fahrgeschwindigkeiten, überwachten Güterwagen und befahrenen Strecken untersucht. Zur Verstärkung der Robustheit wurde eine Datenvermehrungsmethode entwickelt, die Mehrkörpersimulation und die gewichtete Merkmalsraummittelung zur Erweiterung der Simulationsdaten umfasst.

Die zweite Anwendung betrifft die Gleisgeometrieüberwachung. Eine auf 2D-LCNN basierender Methode zur Erkennung und Verfolgung virtueller Punkten wurde entwickelt und in der dritten Publikation beschrieben. Diese Methode ermöglicht die Echtzeitmessung seitlicher Bewegung der Räder auf den Schienen mit der Abtastfrequenz von über 30 Bildern pro Sekunde. Darüber hinaus wurde eine Datenvermehrungsmethode basierend auf Bildverfälschung zur Verstärkung der Robustheit gegenüber unterschiedlichen Wetterbedingungen und Verschutzungen angewendet.

Echtzeitfähigkeit und Robustheit sind die allgemeinen Themen der Zustandsüberwachung. Die entwickelten Methoden können in ähnliche Szenarien zur Echtzeit-Datenverarbeitung und Robustheitsverbesserung verwendet werden. In vorliegender Dissertation wird dazu das Verbesserungspotenzial der entwickelten Methoden diskutiert.

List of Contents

List of Abbreviations

1D	One-Dimensional
AE	autoencoder
bACC	balanced Accuracy
BOGP	Bayesian Optimization with Gaussian Process
CAE	Convolutional Autoencoder
CBM	Condition based Maintenance
CCD	Charge-Coupled Device
C-DCGAN	Conditional-Deep Convolutional Generative Adversarial Network
CNN	Convolutional Neural Network
CV	Computer Vision
CWT	Continuous Wavelet Transform
DBA	dynamic time warping Barycenter averaging
DCGAN	deep convolutional generative adversarial network
DL	Deep Learning
DNN	deep neural network
DSC	Depthwise Separable Convolution
DSCN	Deep Separable Convolutional Network
DTW	dynamic time warping
DTWA	dynamic time warping alignment
EHS	Energy Harvesting System
EMD	Empirical Mode Decomposition
ERM	Empirical Risk Minimization
ES	Embedded System
FCF	Fault Characteristic Frequencies
FCNN	Fully Connected Neural Network
FEM	finite element method
FLOP	Floating-Point Operation
FMECA	Failure Mode, Effects and Criticality Analysis
fps	frames per second
FWFSA	Fast Weighted Feature-Space Averaging
GAN	Generative Adversarial Network
GAP	Global Average Pooling
GBDT	Gradient Boosting Decision Tree
HPE	Human Pose Estimation

HS	Hard-Swish function
HT	Hilbert-transform
IMU	Inertial Measurement Unit
IMV	Infranord Measurement Vehicles
IS2	Maintenance Level II for Wheelsets
KL	Kullback-Leblier
KPI	Key Performance Indicator
LCNN	Lightweight Convolutional Neural Network
LDR-CNN	Lightweight Deep Residual Convolutional Neural Network
LDWR	Wheels' Lateral Displacement on the Rail
MAC	Multiply–Accumulate
MBS	Multibody Dynamic Simulation
MFD	Machine Fault Diagnosis
ML	Machine Learning
MMD	Maximum Mean Discrepancy
MSE	Mean Squared Error
NDT	Nondestructive Testing
NL	Nonlinearity
NN	Neural Network
OCMS	Onboard condition monitoring systems
PM	Predictive Maintenance
RASF	Reality-Augmented Simulation Faulty
RKHS	Reproducing Kernel Hilbert Space
RMSE	Root Mean Squared Error
RoI	Region of Interest
RUL	Remaining Useful Life
SAE	Stacked AutoEncoder
SE	Squeeze and Excitation
SIM	Simulation
SOTA	State of The Art
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
TG	Track Geometry
TGMS	Track Geometry Measurement System
TQI	Track Quality Indicator
TRV	Track Recording Vehicle

t-SNE	t-distributed Stochastic Neighbor Embedding
UGMS	Unattended Geometry Measurement System
WDBA	Weighted Dynamic Time Warping Barycenter Averaging
WF	Wheel Flat
WFD	Wheel Flat Detection
WILD	Wheel Impact Load Detector
WPT	Wavelet Packet Transform
WT	Wavelet Transform
WTMS	Wayside Train Monitoring System

General

e.g.	exempli gratia
eq.	equation
etc.	et cetera
i.e.	id est
incl.	including

Symbols (e.g. for mathematical equations)

λ	irregularity wavelength
ACC_C^f	clean accuracy of a diagnostic model
$ACC_{v,s}^f$	impaired accuracy of a diagnostic model f under the v-th condition variation with the s-th severity
C	the number of classes
F	feature map
I	input images
J	number of the channels
K	size of the convolutional filters
L	chord length
mRB^f	mean robustness of a diagnostic model f
M	number of convolutional filters
N	array length of the input feature map
P_{r1}	rail reference point for the definition of lateral alignment
P_{r2}	rail reference point on the outer rail edge
P_w	wheel reference point on the wheel flange

$p(x)$	marginal distribution
$p(x, y)$	joint probability distribution
$p(x\|y)$	conditional distribution
$RB_{v,s}^{f}$	robustness of a diagnostic model f under the v-th condition variation with the s-th severity
R	wheel radius
S	the number of variation severities
s	scaling factor controling signal-to-noise ratio
V	the number of variation types
w_i	weight of the i-th time series sample
W	weights of convolutional filters
$\bar{x}_0$	initial average time series
X_s	a set of time series samples in the source domain
X_t	a set of time series samples in the target domain
$\bar{x}$	averaged time series sample
x_f	longitudinal distance along the wheel flat
x_i	i-th time series sample
$x_{s,i}^{r}$	i-th time series sample in the real-world source domain
$y_{s,i}^{r}$	label of the i-th time series sample in the real-world source domain
Y	irregularity amplitude
z	relative displacement measured by the onboard transducer

List of Publications

Index	Publication and my contribution	Status
1	Dachuan Shi, Yunguang Ye, Marco Gillwald, Markus Hecht (2021), Designing a lightweight 1D convolutional neural network with Bayesian optimization for wheel flat detection using carbody accelerations, International Journal of Rail Transportation, 9:4, 311-341, https://doi.org/10.1080/23248378.2020.1795942	Published
	First author and corresponding author: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Project administration	
2	Dachuan Shi, Yunguang Ye, Marco Gillwald, Markus Hecht, (2022), Robustness enhancement of machine fault diagnostic models for railway applications through data augmentation, Mechanical Systems and Signal Processing, Volume 164, 108217, ISSN 0888-3270, https://doi.org/10.1016/j.ymssp.2021.108217.	Published
	First author and corresponding author: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Project administration	
3	Dachuan Shi, Eldar Šabanovič, Luca Rizzetto, Viktor Skrickij, Roberto Oliverio, Nadia Kaviani, Yunguang Ye, Gintautas Bureika, Stefano Ricci, Markus Hecht, (2022), Deep learning based virtual point tracking for real-time target-less dynamic displacement measurement in railway applications, Mechanical Systems and Signal Processing, Volume 166, 108482, https://doi.org/10.1016/j.ymssp.2021.108482.	Published
	First author and corresponding author: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization	

Contributor Roles Taxonomy is used to depict the contributions. The terms are explained below.

Term	Definition
Conceptualization	Ideas; formulation or evolution of overarching research goals and aims
Methodology	Development or design of methodology; creation of models
Software	Programming, software development; designing computer programs; implementation of the computer code and supporting algorithms; testing of existing code components
Validation	Verification, whether as a part of the activity or separate, of the overall replication/ reproducibility of results/experiments and other research outputs
Formal analysis	Application of statistical, mathematical, computational, or other formal techniques to analyze or synthesize study data
Investigation	Conducting a research and investigation process, specifically performing the experiments, or data/evidence collection
Resources	Provision of study materials, reagents, materials, patients, laboratory samples, animals, instrumentation, computing resources, or other analysis tools
Data Curation	Management activities to annotate (produce metadata), scrub data and maintain research data (including software code, where it is necessary for interpreting the data itself) for initial use and later reuse
Writing - Original Draft	Preparation, creation and/or presentation of the published work, specifically writing the initial draft (including substantive translation)
Writing - Review & Editing	Preparation, creation and/or presentation of the published work by those from the original research group, specifically critical review, commentary or revision – including pre-or postpublication stages
Visualization	Preparation, creation and/or presentation of the published work, specifically visualization/ data presentation
Supervision	Oversight and leadership responsibility for the research activity planning and execution, including mentorship external to the core team
Project administration	Management and coordination responsibility for the research activity planning and execution
Funding acquisition	Acquisition of the financial support for the project leading to this publication

1. Introduction

Traditionally, conditions of railway assets are assessed by inspections within the maintenance process. The inspection interval and content limit the availability of up-to-date information on the condition of the individual assets. Although today's maintenance system ensures the safe operation of the railway system without this information, it can hardly achieve an optimal trade-off between maintenance efforts and assets' reliability as well as availability. Within the wave of digitalization, condition monitoring is becoming an essential part of maintenance in the railway system. From the monitoring data, the useful information is extracted by intelligent algorithms to represent the up-to-date asset conditions. This offers the opportunity and feasibility for sustainable optimization of maintenance, increasing the efficiency and quality of the railway system. Being motivated by this ultimate objective, we investigate data processing towards robust real-time condition monitoring and fault diagnosis for railway assets.

In the following, common inspection measures within the maintenance process for rolling stock and tracks are presented as the research background. Their limitations are the main driver of railway condition monitoring applications. However, they face specific challenges related to real-time data processing and diagnostic robustness, which are the main focus of our research. These challenges will be explained in detail. Finally, the structure of the present dissertation is introduced.

1.1. Inspection and maintenance of railway assets

Maintenance is defined as the combination of all technical, administrative and managerial actions during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function [1]. Inspection and monitoring are the parts of technical maintenance actions. In current practice in the railway system, inspection is the most common measure to assess the asset conditions within maintenance processes. It is worth noting that inspection is a conformity examination, aiming to determine whether the asset conforms with the requirements of safe operating. It does not determine the exact asset conditions. The inspection tasks and intervals heavily depend on the target assets. For instance, in the maintenance process of railway freight wagons, technical inspections are carried out before every train journey and wagon handover, after new train composition as well as loading and unloading. These activities usually take place every day. The technical inspections aim to assess the operating safety and rail worthiness of wagons, identify any failures, and take appropriate maintenance decisions [2]. In practice, the qualified inspectors walk along the train on both sides to examine each wagon primarily based on their vision, hearing and cognitive capabilities. Simultaneously, the inspectors conduct diverse tasks for train preparation, brake testing, documentation, communication, and preparation of transport paper. This leads to the fact that the inspectors have to assess the wagon conditions and identify the component failures in a very short time, typically one minute per axle [3]. However, there are more than a hundred failure modes on freight wagons according to the GCU failure category. Despite remarkable cognitive capabilities of inspectors, the probability of failure detection relies on their physical and mental status as well as the visibility of the components and failures, which could be affected by failure locations, weather conditions, darkness, and

contamination. In the case that a failure is detected, a wagon damage report should be filled, on which the failure is described with a pre-defined failure code and additional textual descriptions. The corresponding maintenance measure according to the failure severity has to be executed immediately. In the worst case, the defective wagon must be detached from the train and dispatched to the workshop for repair. This maintenance measure directly delays the train departure and increases the cost of train composition.

The above-described inspection processes take place during the normal operation of rail freight transport. The successful detection of a failure triggers unscheduled maintenance, which corresponds to the strategy of corrective maintenance. It ensures operation safety. However, it disrupts normal operation. To reduce failure occurrences and unplanned downtimes as much as possible, corrective maintenance is combined with preventive maintenance. Preventive maintenance requires maintenance actions scheduled at fixed intervals. The intervals are usually defined in terms of time or mileage for different levels of scheduled maintenance based on the experience and the historical maintenance. For instance, light maintenance may take place every several months, which includes the technical inspection, lubrication, cleaning, replacement of wear components such as brake blocks and measurement of wheel profiles. Maintenance Level II for wheelsets (IS2) occurs typically every two or three years, in which wheelsets are dismounted for reprofiling and nondestructive testing (NDT). The wagon revision occurs every 6-8 years, in which the entire wagons are dissembled for examination. Due to the large interval of scheduled maintenance, the focus of the inspection is more on the wear status and the fatigue failures, rather than abrupt failures which are mostly detected in unscheduled maintenance.

The combination of corrective and preventive maintenance constitutes today's maintenance strategy for railway assets. It is not confined to rolling stock maintenance, but also applied to railway infrastructure maintenance. For track maintenance, inspections include regular human visual inspections, scheduled track geometry inspections with track recording vehicles (TRV), scheduled NDT of rails, scheduled running dynamics inspections, etc. Human visual inspections aim to detect irregularities that can be visually observed. They are carried out more frequently than inspections with specific measurement systems. The latter focuses more on wear status and fatigue failures. For instance, track geometry inspections are performed every several months depending on the maximum line speeds of the inspected track sections. The parameters of track geometry and rail profile are accurately measured and compared to the thresholds.

Although the current maintenance strategy has been well implemented in practice and can ensure safe railway operation, it has the following limitations in maintenance efficiency.

- Failure detection during normal operation relies on frequent visual inspections. More frequent inspections and a longer duration of a single inspection can certainly increase the probability of failure detection and the reliability of the assets. However, it increases higher labor costs and may also decrease the availability of the assets.
- Failure detection by visual inspections usually triggers unplanned downtimes, decreasing assets' availability.
- Visual inspections cannot assess the exact conditions of the assets. The inspection results are not recorded if no failures are detected. Consequently, maintenance records merely

contain records of failure events. In other words, visual inspections cannot help avoid failure occurrences or predict wear-out points.

- Scheduled instrumented inspections provide a comprehensive assessment of asset conditions. However, it cannot be performed frequently due to high costs and time-consuming. Low-frequency data can hardly be used for accurate wear prediction.
- Planning scheduled maintenance is guided by the standards or regulatory requirements, operating and maintenance experience, historical maintenance records, and results of scheduled inspections. Due to a lack of up-to-date information related to the conditions of the individual assets, the scheduled maintenance may cause waste in the life cycles of spare parts or be planned too late to prevent failure occurrences.

In a summary, the main obstacle is the lack of a measure that can quantify the up-to-date conditions of the assets without interruption of normal operation. Condition monitoring techniques along with intelligent data processing algorithms can make up for this gap. On the one hand, condition monitoring systems can detect abrupt failures at early stages during normal operation. The train drivers or infrastructure operators are informed about failure detection so that they can take steps to avoid safety hazards and meanwhile also arrange maintenance actions on time. In this way, the unscheduled maintenance is turned into emergently scheduled maintenance. On the other hand, the cumulated monitoring data over time can be used for wear prediction to support the planning of scheduled maintenance, especially in which the worn components are replaced or reprofiled. This not only saves the life cycle of spare parts but also enables the optimization of spare parts inventory management.

With more and more systems of condition monitoring and data analytics being employed in practice, a paradigm shift from corrective and preventive maintenance to condition based and predictive maintenance (CBM and PM) is undergoing in the railway system, as it also does in other industrial and transportation sectors. CBM is defined as a type of preventive maintenance in EN 13306 [1], which includes assessment of assets' conditions, analysis, and possible follow-up maintenance actions. Although condition assessment is also conducted in today's maintenance processes by visual inspections and scheduled instrumented inspections, it cannot change the fact that maintenance is scheduled at fixed intervals rather than based on assets' conditions. The involvement of condition monitoring is the keystone of the actual CBM, since it fundamentally increases the quantity and quality of data related to asset conditions. This data facilitates advanced data analytics methods for early detection of abrupt failures and tracking of degradation progress. With the accumulation of monitoring data over time, it becomes possible to predict failures and wear-out as well as estimate the remaining lifetime of assets, which is the key part of PM.

1.2. Challenges and outline

CBM and PM require intelligent algorithms to determine the asset conditions for the subsequent fault diagnosis and degradation prognosis. Apart from asset conditions, planning and making decisions for maintenance should also take into account the available maintenance resources, the economic aspects, the operation plans, etc. It is an optimization

problem with the objective of achieving the highest reliability and availability of railway assets and meanwhile the lowest life cycle costs.

The present dissertation focuses on data processing of monitoring data. Development of condition monitoring hardware and maintenance optimization are out of scope. To be more specific, we aim to tackle two challenges of data processing encountered in practice.

First, condition monitoring in railway applications usually requires real-time processing of acquired data. Raw monitoring data, which occupies a large space, can hardly be transmitted directly to the cloud for subsequent processing. This is particularly true in railway applications. Many railway assets have not been electrified. The onsite infrastructure for power supply and data communication is absent. For instance, onboard monitoring systems on rolling stock have to transmit data via 2G or 4G mobile network. 4G network is not always available along the track. The data may have to be transmitted by 2G network with speeds of up to 5 kB/s. It is impossible to upload a large amount of data with such low transmission rates. Monitoring systems installed on infrastructure face the same problem, since data cables are not standard equipment installed along the track. Moreover, real-time fault diagnosis is highly desired, since the relevant operators should be informed immediately to take measures to prevent safety hazards in the event of severe failures. From the maintenance perspective, the entities in charge of maintenance (ECM) should also be informed immediately to plan maintenance actions in time. Furthermore, data processing algorithms for condition monitoring are commonly executed on edge devices such as embedded electronic systems, rather than on powerful workstations. The embedded systems have very limited computation power. This results in further challenges for data processing with regard to computational complexity.

Second, operating conditions of railway assets in the wild outdoor environment vary in a large range. Variations of operating conditions reflect in monitoring data and cause distribution shifts thereof, which may induce robustness problems of diagnosis algorithms. For instance, vehicle speeds and track conditions significantly affect the results of onboard vibration monitoring. For optical monitoring, weather conditions are usually one of the decisive factors. When condition monitoring systems are deployed on different objectives, which even belong to the same type of assets, calibration of hardware and algorithm settings is often necessary to ensure meaningful monitoring results. These robustness problems have not been well studied in the area of machine fault diagnosis (MFD).

To cope with these challenges, we have conducted extensive research on data processing towards robust real-time condition monitoring and fault diagnosis for railway assets. We take advantage of deep learning techniques and propose lightweight convolutional neural networks (LCNN) to realize real-time data processing with low computational demands. During the training process, we propose a novel data augmentation framework for robustness enhancement. The proposed approach is demonstrated in two distinct railway applications. The first application deals with wheel flat detection (WFD) using onboard vibration monitoring, which supports the maintenance of railway freight wagons. The second application concerns track geometry monitoring based on computer vision, aiming to support track maintenance. On the one hand, we demonstrate that different tasks of condition monitoring share the same challenges, which can be addressed by the proposed general approach of data processing. On the other hand, we tackle the specific unsolved problems in

the two applications. The problems and the previous research are detailed in Section 2. Our solutions for the two applications have been described in three scientific publications, which constitute the present cumulative dissertation. Its outline is as follows.

In Section 2, the fundamentals of condition monitoring and fault diagnosis are first introduced. The topics related to the two railway applications are explained in more detail. For each application, the existing and emerging monitoring techniques are introduced. Afterward, the state-of-the-art data processing methods reported in academic publications are reviewed.

In Section 3, the research questions based on the actual needs and the gaps of the previous studies are proposed. The research questions will be answered in the three publications, which are the main part of this dissertation.

In Section 4-6, the three publications are presented. The first and second publications deal with the real-time capability and the robustness of the algorithm for WFD on freight wagons. The third publication addresses the real-time computer vision application for track geometry monitoring.

In Section 7, the main findings and limitations of our work are discussed. The potential improvement is proposed. In addition, the potential use of the developed methods for similar applications is envisaged.

In Section 8, the conclusions of the dissertation are drawn.

2. Scientific Background

This section begins with a general introduction of the relevant fundamentals. As our work contributes to rolling stock bogie monitoring and track geometry monitoring, we dive into these two topics, presenting their status quo in practice and the state of the art (SOTA) in academia.

2.1. Fundamentals

Condition monitoring aims to measure relevant parameters to represent asset conditions and to track their changes over time [1]. In the following, the general process and the involved techniques of condition monitoring are first introduced, followed by a detailed explanation of fault diagnosis.

2.1.1. Condition monitoring

The main concept behind condition monitoring is to use low-cost non-invasive sensors to measure the indicative parameters without interrupting normal operation and analyze the measurement data over time for condition assessment and prediction. Condition monitoring has been widely applied in different industrial and transportation sectors. A representative application refers to vibration monitoring of rotating machines such as electric motors, turbines, compressors, etc. Instrumenting accelerometers for vibration monitoring is much more advantageous than disassembling machines for inspection of their internal conditions. The vibration signatures of machines' healthy conditions are distinguishable from their faulty conditions with the help of frequency analysis. Other common monitoring techniques include lubricant analysis, acoustic emission, thermography, ultrasound and current signature analysis. Although their used sensing techniques and the corresponding data analytics methods are different, the general procedure of condition monitoring is the same as vibration monitoring. This procedure is standardized in the standard ISO 17359.

Figure 1 shows the adopted condition monitoring procedure within the CBM/PM process. A feasibility study should be conducted at first to ascertain and decide whether condition monitoring is worth doing for the specific assets. Meanwhile, the key performance indicators (KPI), such as availability, reliability and LCC, should be defined as the objectives of maintenance optimization. Afterward, the target asset is analyzed by breaking down its components and functions. A complex system may have tens or even hundreds of components, each of which may have various failure modes. Condition monitoring techniques can hardly cover all of the components and their failure modes. Therefore, the first step of condition monitoring is to identify the critical components, functions and/or failures modes. A standard method for criticality analysis is failure mode, effects and criticality analysis (FMECA), which evaluates the criticality of each failure mode with regard to its probability, detectability and severity. The high-ranking failure modes have priorities of condition monitoring. Once the targets are identified, the relevant physical parameters and the corresponding measurement methods are reviewed for the selection of the potential condition monitoring techniques. Monitoring systems should have high automation, high robustness, low investment costs and low requirements for deployment. The monitored physical parameters should be able to represent the conditions of the targets. The installation locations

should enable data acquisition without affecting assets' normal operation. Based on these requirements, suitable monitoring techniques are selected and deployed for data acquisition. The raw or processed data is transmitted to the cloud server, where the data is stored, further processed and visualized. The results of condition monitoring are fed into the maintenance program for planning maintenance activities. Finally, the evaluation of maintenance KPIs should be done and included in the annual maintenance report. In this way, the benefits of CBM/PM can be quantified.

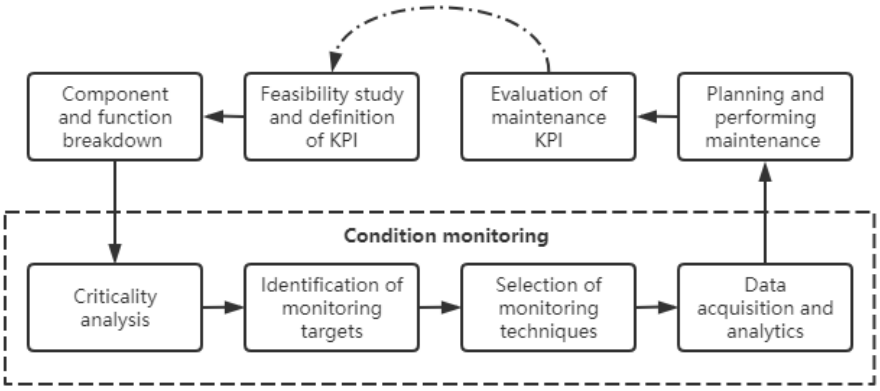


Figure 1 Condition monitoring within the CBM / PM process

2.1.2. Data processing for condition monitoring

Fault diagnosis methods broadly fall into two categories, namely the model-based and data-driven approaches. The model-based approach evaluates the residuals between the model and monitoring outputs. The commonly used methods for railway applications have been reviewed by Strano et al. [4] and Ngigi et al. [5], such as Kalman filter and particle filter for estimation of suspension and wheel-rail contact parameters. The model-based methods are able to estimate system outputs in faulty conditions. However, these methods do not give a decision boundary for fault diagnosis. It is not clear how large residuals can indicate an occurrence of failures. Also, modeling requires known parameters of vehicle components and track irregularities (as system noise). These parameters are not always available in practice. Therefore, these methods are mostly verified with simulation data for theoretical studies. They have been rarely applied in practice to process the monitoring data. In this work, we focus on the data-driven approach. Its general procedures are introduced as follows.

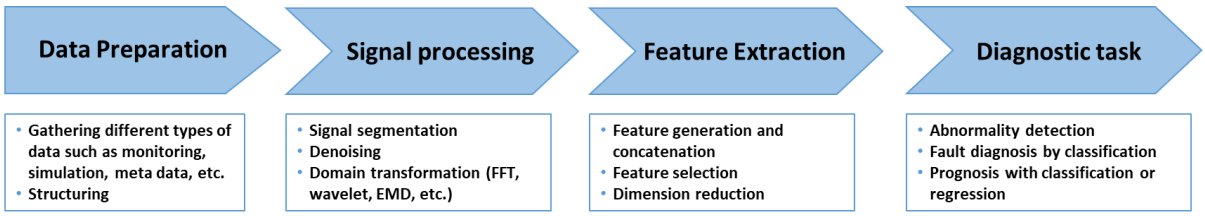


Figure 2 Procedures of data processing for condition monitoring

Data processing for condition monitoring has three main purposes: abnormality detection (or fault detection), diagnosis of failure modes, and degradation prognosis (or prediction of remaining useful lifetime). Abnormality detection aims to distinguish abnormal conditions from normal ones. Failure mode diagnosis aims to identify the exact failure modes.

Degradation prognosis aims to predict degradation or wear progress. Despite different purposes, the procedures for data processing are similar, as presented in Figure 2.

First, data is gathered from heterogeneous data sources and structured in the required format for the subsequent processing. Heterogeneous data have different formats such as numeric data, texts and videos. They should be pre-processed into a structured format for further use. Apart from the condition monitoring data, other useful data include simulation data generated by physical or mathematical models, weather data, and metadata related to the monitored assets as well as their operating environments. In this context, modeling and simulation are considered as a way to generate synthetic data that is complementary to monitoring data, rather than serve as the model-based diagnostic approach. Metadata and weather data provide the auxiliary information indicating the characteristics of the monitored assets and their operating conditions. In some cases, they can be directly used for fault diagnosis. For instance, the rotating speed and the geometric size of a bearing indicate the characteristic frequency of single-point failures such as spalling, which can be used as a diagnostic rule. In other cases, they merely indicate boundary conditions. For instance, the power consumption of railway point-operating machines is very different in summer and winter. Especially on snowy days, the resistance to move the switch rail is larger due to frozen ice and snow. The diagnostic models developed on monitoring data collected in summer are not directly applicable to the winter data.

Second, signal processing is employed to highlight diagnostic information while suppressing noisy information. The continuous monitoring data is usually divided into short segments. Digital filters can be applied to eliminate stochastic noise and filter out frequency ranges that are not interested. The waveform time series data such as vibration and acoustic signals are acquired in the time domain and can be transferred into the frequency spectrum by various algorithms such as fast Fourier transform (FFT) and wavelet transform (WT). Especially for diagnosis of rotating machines, patterns of faulty signals could be more conspicuous in the frequency spectrum than in the time domain. For non-stationary signals, decomposition methods such as empirical mode decomposition (EMD) and variational mode decomposition decompose a complex signal into a series of modes, which contains different frequency components, ranging from high-frequency to low-frequency ones respectively. The noise is typically included in high-frequency components and can thus be filtered out.

Third, diagnostic information is extracted from the filtered data. Monitoring or simulation data may have large dimensions. For instance, vibration data are typically sampled at thousands of Hertz. That means a segment of the several-second vibration signal contains over ten thousand data points. The diagnostic information may only be several numeric values such as the maximum value and the number of the peaks. In this sense, the most essential step for diagnosis is to extract the low-dimensional representatives from the high-dimensional data, namely feature extraction. Features refer to the representatives of the original data. Features can be manually defined according to domain knowledge. Features can also be automatically learned by deep learning algorithms, which are state-of-the-art (SOTA) approaches for MFD. The features may need further selection and dimension reduction to reduce their redundancies and improve diagnostic performance.

Finally, diagnostic tasks are performed. The inputs for diagnostic models are the refined features. Depending on the diagnostic purposes, the outputs are different. For abnormality

detection, the output is whether the health condition is normal. From the mathematical perspective, this can be formulated as an outlier detection problem. For fault diagnosis, the output is the failure mode that the input features indicate. Fault diagnosis is a multiclass-classification problem. Each class refers to one failure mode. The diagnostic model is a classifier that classifies the inputs into the corresponding classes. For degradation prognosis, the output could be the degradation status or a numeric health indicator. In the former case, degradation prognosis is formulated as a multiclass-classification problem, in which multiple health statuses in the assets' life circle are defined as the classes. The model is to classify the inputs into the corresponding health classes. In the latter case, degradation prognosis is formulated as a regression problem, in which the model is to predict the defined numeric health indicator from the input features.

2.1.3. Deep learning for feature extraction and diagnostic tasks

Deep learning has become the SOTA approach for feature extraction and diagnostic tasks in various industrial sectors. It enables learning features adaptively from large amounts of heterogeneous input data. Deep learning has several variants such as deep neural network, deep brief network, deep reinforcement learning, etc. In the following, deep convolutional neural network (CNN) is introduced briefly as the theoretical basis of our research.

CNN was firstly proposed in the 1960s from the neurobiological experiments [6]. The modern multilayer CNN architecture was proposed by LeCun for a computer vision application [7]. In his work, the supervised training based on backpropagation was also firstly applied on CNN. It becomes the foundation of modern CNNs. Typically, a basic CNN architecture consists of multiple layers of convolution, pooling and activation in a row for feature extraction as well as one or more subsequent fully connected layers and a softmax function for classification. The convolutional layer contains a series of convolutional filters (also known as "kernel"). The convolution is executed by sliding the filters over the input. In each sliding step, the filters convolve the receptive fields of the input features, which have the identical size as the filters, resulting in the filtered features. The convolution operation is the same as digital filtering. The only difference is that the weights of convolutional filters are automatically learned during the training process, while the weights of digital filters are manually designed. The pooling layer aims to downsample the input features by taking the average or maximum value within a neighborhood region. This not only reduces the size of the outputs but also increases the robustness against local shifts and small distortion. Convolution and pooling are linear mathematical operations with multiplication and addition. To learn a complex pattern, non-linearity is introduced to the network by the activation function, which is a nonlinear transform function such as tanh, ReLu and sigmoid function. Stacking convolutional layers, pooling layers and activation functions in certain order can hierarchically extract the low-dimensional features from the input data. To perform the classification, fully connected layers and a softmax function are combined and placed at the end of the entire network. Apart from these basic elements, batch normalization has also become a standard operation within the modern CNN architecture. It was introduced in [8] to solve the problem of internal covariance shift. Batch normalization is to standardize the feature vectors within hidden layers by subtracting their mean value and being divided by their standard deviation.

Prior to the use of a CNN for feature extraction and fault diagnosis, a CNN model should be trained. Training a neural network is an optimization task. An optimization objective should be defined first, which is commonly termed loss function. For instance, a loss function can be mean squared errors (MSE) between the values predicted by the neural network and the true values. The optimization task is to minimize the loss function by adjusting the learnable parameters within the neural network. The most common optimization algorithm is stochastic gradient descent. In each training step, the gradient of the loss function with respect to the learnable parameters is computed in a single forward and backward pass through the network thanks to the so-called backpropagation algorithm. The values of the learnable parameters are updated towards the minimum loss. The training process usually takes a long time. Once the deep learning model is trained, it can be used for inference, where a single forward pass through the network is executed for each input. Therefore, inference costs much less computational time.

In the next subsections, the SOTA condition monitoring systems and the diagnostic algorithms for wheel flat detection and track geometry monitoring are reviewed.

2.2. Wheel flat detection

Wheel flat, as one of the most common failure modes on vehicle bogies, is an oval spot on the wheel running surface. It forms when the wheel is locked by braking and slides along the track. This is caused by improper braking or a defective brake system. On freight wagons, a wheel flat can alse arise due to the brake shoe braking. During the traditional shunting operation, wagons are braked by placing brake shoes on the rail. This may induce a wheel flat on one wheel. Wheel flat can induce large impact forces and impulsive noises, damaging the axle bearing and the track as well as causing noise pollution. In the maintenance processes, wheel flat is detected by visual inspection. However, miss detection may occur, depending on the flat size and the position. Therefore, automatic wheel flat detection during normal operation is highly desired.

2.2.1. Status quo in practice

Traditionally, condition monitoring of rolling stock is implemented by detecting the most critical failures of the related components. It aims to ensure operation safety and does not support rolling stock maintenance. Only catastrophic failures are concerned, such as hot box, out-of-gauge and excessive/imbalanced loads. Wayside wheel impact load detectors (WILD) that are used to detect excessive/imbalanced loads during vehicle pass-by can detect wheel flat as well. An exemplary commercial WILD shown in Figure 3 uses fiber optic sensors to measure the vertical loads, which can be alternatively measured by strain gauges. The conventional WILDs are installed trackside and managed by infrastructure managers for the sake of safe operation. The monitoring data are not provided for rolling stock maintenance.

Driven by the needs of CBM and PM of rolling stock, train monitoring systems are oriented more towards rolling stock maintenance. The new wayside train monitoring systems (WTMS) owned by railway undertakings are deployed where their vehicles frequently and routinely pass by. Consequently, more condition-related data can be collected. In WTMSs, optical sensing is dominated, which can visually inspect vehicle components for fault detection and measure the wear status of wheels as well as brake pads. The automatic inspection through

advanced computer vision systems can support and relieve the manual inspection processes described in Section 1.1. Frequent wear measurement can be used for wear prediction and thus optimization of maintenance schedules. Commercial computer-vision-based WTMSs are available on the market. For instance, the WTMS system provided by DTEC [10] acquires the images in vehicle-top, -bottom and -side views. The side-view cameras examine bogie frames, suspensions and brake pads to detect failures such as broken or missing springs, broken adapters, missing bearing end caps and loosened friction wedges. The bottom-view cameras inspect the underframe components such as wheel surfaces, brake riggings, couplers, lifting plates and wheel carriers. The missing, misplaced, damaged and deformed components as well as foreign objects can be detected. The left photo in Figure 4 presents a detected wheel flat by an underframe computer vision system. A representative application is the deployment of optical WTMSs in the shunting yards for freight wagons. The cameras are installed on a bridge-like frame to acquire the images of the pass-by freight wagons. The laser sensors and the underframe cameras are placed between the sleepers in the track. The right photo in Figure 4 shows a computer-vision-based WTMS used by DB cargo in a shunting yard. Placing WTMS in bottlenecks of the rail freight network, the coverage rate will reach approx. 95% throughout Germany [11].

Figure 3 Ansaldo STS WILD [9]

Figure 4 Left: wheel flat detected by an underframe computer vision system. Right: computer-vision-based WTMS used by DB cargo in shunting yards [11].

Although WTMS can involve advanced and comprehensive inspection technologies, it cannot assess vehicle behaviors during running. Onboard condition monitoring systems (OCMS) can bridge this gap. An OCMS typically consists of a centralized communication hub and

distributed sensor nodes. On locomotives and passenger vehicles, the communication hub refers to the train control and management system, which collects data for control and diagnosis from the sensors distributed on the train via vehicle/train bus [12]. On freight wagons, there were traditionally no electric systems. Recently, a so-called telematics device was introduced in railway industry for tracking and monitoring of freight wagons [13]. In the course of digitization of rail freight transport, many wagon owners in Europe have upgraded their fleets with telematics devices, such as DB Cargo, VTG, SAVVY, Wascosa, etc. On the one hand, a telematics device functions as a communication hub. It is installed on the wagon body and connected wirelessly or wired to the distributed sensors. The collected sensor data are pre-processed and transmitted further to the cloud server. The desired monitoring tasks can be realized by adding the corresponding sensor nodes. On the other hand, a telematics device internally contains an acceleration for vibration monitoring, specially designed to detect abnormal shock events during shunting operations [13].

As wheel flats are mainly caused by the abnormal braking process, monitoring of the brake system can effectively prevent the occurrence of wheel flats. Brake monitoring is integrated in the brake control system on modern locomotives and passenger vehicles. The measured signals for brake control such as air pressures within the brake cylinder and the main brake pipe are also used for fault diagnosis [14]. For freight wagons, brake monitoring mostly serves for automatic brake testing during train preparation. A pressure sensor installed in the brake cylinder examines whether the pressure associated with the piston travel in the brake cylinder is set correctly when the locomotive sends the “apply” or “release” command for brake tests. In addition, the position sensors can be added to monitor the position of the loaded/empty lever and the handbrake lever. All sensor data from the distributed sensor nodes are fused in the telematics device on each wagon and forwarded to the device in the locomotive. In this way, the driver can receive an alert if any abnormalities are detected [15]. However, brake monitoring on freight wagons has not been applied for real-time diagnosis during the running operation.

Once a wheel flat forms, vibration monitoring is the most effective way for WFD. Vibration directly represents vehicle dynamics and thus enables detection of any mechanic failures that induce abnormal dynamic behaviors. Also, the required accelerometers for vibration monitoring are very robust in harsh environments and affordable for massive deployment. However, Vibration monitoring has not been widely applied in practice. The current EU technical standard EN 15437-2 merely requires the onboard temperature monitoring of bearings [16]. Vibration monitoring is much more complicated than temperature monitoring in terms of data processing. The same vibration data can be used for different diagnostic tasks by means of corresponding data processing algorithms. It has attracted extensive research efforts from industry and academia. Commercial OCMSs are available on the market. For instance, the OCMS provided by SKF involves multiple sensors to measure bearing temperature, vehicle speeds as well as axlebox vibrations. It monitors vibration levels at axleboxes and detects wheel tread damages as well as early-stage bearing failures [17]. This monitoring system is designed for vehicle manufacturers for integration into the vehicle system, as certain sensors should be installed inside vehicle components and powered by the vehicle power supply. Its capabilities of fault diagnosis rely on the vehicle speed signal, which can be used, along with additional knowledge of some geometry data, to calculate fault characteristic frequencies (FCF) of bearings and wheels in the frequency spectrum. Another

type of OCMS is designed for plug and play, as shown in Figure 5. The system provided by Perpetuum allows easy installation at wheel level via a mechanical adapter [18]. It is self-powered by vibration energy harvesting and monitors the temperature and vibration levels. The vibration levels are processed into a so-called wheel/bearing health indicator, presenting the health status. A wheel flat generally results in a larger value of the health indicator. However, the threshold for diagnosis has to be defined based on the user experience.

These axlebox vibration monitoring devices have not been widely employed. One reason is the high investment costs, since each vehicle has to be equipped with eight devices at axleboxes. Another reason lies in maintenance logistics. During rolling stock maintenance, the bogie of a vehicle can be replaced with a new bogie. Unless the carbody-side communication hub can be automatically paired with the bogie-side sensor nodes and all the bogies are instrumented, the installation of alxebox sensors causes an additional management burden. Especially on freight wagons, operators desire to exploit the existing telematics devices on the carbody for WFD, avoiding additional investment on sensors [19].

Figure 5 Commercial onboard vibration monitoring systems provided by Perpetuum [18]

2.2.2. State of the art of the research

Commercial monitoring systems provide mature technologies for data acquisition. The main focus of the research lies in the data processing for fault diagnosis, especially concerning onboard vibration monitoring. Dynamics can be studied through multibody dynamic simulation (MBS), which is the standard method to research railway vehicle dynamics. A failure is modeled by the mathematical description. The failure model is inserted into the vehicle MBS model for dynamic simulations under various operating conditions. The wheel flat is commonly described by the variation of wheel radius versus angle position. The mathematical formula for the radius variation may be different [20-23]. Ren [20] used a 3D wheel flat model and a vehicle-track coupling model to investigate the maximum dynamic impact force induced by a wheel flat with different geometric shapes at different running speeds. Bernal et al. [21] introduced a 2D rounded wheel flat into a Y25 freight wagon model and conducted a feasibility study of WFD using accelerations at axlebox, bogie frame, and carbody, respectively. Three statistic features in the envelope power spectrum and the time domain were defined for diagnosis. The wheel flat vibration signatures were significantly weakened at the bogie and carbody level, making WFD difficult with the defined features. Bosso et al. [22] built a freight wagon model with a wheel flat for simulation studies and constructed a feature based on the maximum and root mean square (RMS) value of

accelerations during a single wheel rotation. A validation test showed that a small wheel flat can hardly be detected with the proposed features. Similarly, Ye et al. [23] have built a freight wagon model with a rounded wheel flat. Based on the simulation results, a surrogate model was developed to correlate the averaged peak value of the wheel-flat signal to the flat length and the vehicle speed. Given the measured acceleration and the vehicle speed, this surrogate model can detect the presence of a wheel flat and determine its length. One prerequisite is that the input measurement data should be obtained while the vehicle is running at an almost constant speed for a while.

Unlike simulation studies, the data-driven approach, as introduced in Section 2.1.2, can be directly applied to monitoring data. The majority of previous studies for WFD focus on signal processing to identify FCF and attempt to define a single feature based on the processed data for diagnosis. Baasch et al. [24] employed cepstral analysis for wheel condition monitoring. It was assumed that a local wheel surface failure like wheel flats induces a Dirac-impulse-like spike during a single wheel rotation. Given the continuous monitoring data along the track, cepstral analysis robustly extracted the distance between two spikes in the distance domain which should be equal to the wheel circumference. This "feature" can be used for diagnosis. Chen et al. [25] proposed a two-level adaptive chirp mode decomposition method as a time-frequency analysis method to process axlebox accelerations at variable speeds, where the rotation frequencies vary over time. The recognized FCF can be used for diagnosis. Shim et al. [26] conducted the cepstrum analysis and cross-correlation analysis of axlebox accelerations for WFD. By using amplitude ratios as the feature, it was concluded that the cross-correlation analysis overperforms the cepstrum analysis. Jiang et al. [27] used empirical mode decomposition (EMD) to decompose the raw signals into several intrinsic mode functions that separate the wheel flat signatures from interferences. Zhao et al. [28] analyzed the vibration signal in the high-order spectrum that can suppress Gaussian noise. Li et al. [29] proposed adaptive multiscale morphological filtering for denoising. In these works, the proposed signal processing methods were demonstrated merely on the selected data samples, while a statistic evaluation on a large number of samples with certain diversities was missing. The used validation data were mostly obtained on the laboratory test rigs or from the simulations, which can hardly represent the real-world complexity. Diagnosis based on a single feature such as FCF can hardly be robust to real-world monitoring data, where unknown perturbations can distort the frequency pattern of the wheel-flat signal. Furthermore, these works did not define a method for diagnostic decision-making. In comparison, Gericke [30] performed a complete WFD procedure. First, dozens of features were manually defined in the time domain, the frequency spectrum, the envelope spectrum, and the cepstrum. Second, feature selection was conducted through the wrapper and filter methods. It was found that a combination of several selected features overperformed any single feature. Finally, several classifiers like Naive Bayes classifier, k-nearest neighbor classifier, decision tree, and neural network were tested based on the selected features to find the best combination of the features and the classifier. Similarly, Kim et al. [31] built a neural network for WFD based on the extracted features using wavelet packet decomposition and Hilbert transform.

The periodic impulsive pattern of wheel flat is the typical failure pattern of rotating machines such as bearings and gearboxes, for which diagnostic algorithms have been more extensively studied. The diagnostic techniques used for rolling stock axle bearings were reviewed by Entezami et al. [32] and Xu et al. [33]. Liu et al. [34] and Wang et al. [35] reviewed condition

monitoring and fault diagnosis techniques for wind turbine bearings and gearboxes. The standard procedure of vibration data processing involves signal processing techniques for denoising, in order to spotlight FCFs. The calculated FCFs can be directly used for diagnosis by comparison with the theoretical FCFs, which are derived from rotating speeds and geometric parameters of bearings/gearboxes. A threshold of the residuals is defined for the diagnostic decision. In most cases, the calculated FCFs are merely defined as a part of the features. Other features could be the statistic indicators, such as RMS and kurtosis. After feature extraction, feature reduction and selection may be applied. To perform the diagnostic task, machine-learning-based methods are employed to train a diagnostic model based on the features and their ground-truth labels. However, this classical procedure needs prior knowledge about the monitored machines to define the appropriate features, which are decisive for a successful fault diagnosis. Moreover, today's condition monitoring scenarios are shifting into the era of big data. Conventional machine learning models cannot meet the requirements to process big data [36]. To overcome these problems, deep learning techniques have attracted extensive research in recent years, which aim to adaptively learn the features from the inputs and perform an end-to-end diagnosis. Lei et al. [37], Liu et al. [38] and Zhang et al. [39] reviewed the deep learning applications for fault diagnosis of rotating machines. Variants of deep neural networks, such as autoencoder (AE), CNN and recurrent neural network, have been widely applied for abnormality detection, fault diagnosis and degradation prognosis. AE can be trained by reconstruction of the inputs without their ground-truth labels. Therefore, it is commonly used for unsupervised feature learning. The learned features are provided for further diagnostic tasks. Alternatively, the reconstruction residuals can be directly used for abnormality detection. When AE is trained by healthy data, feeding abnormal data in the trained AE model results in large residuals. Long short-term memory as a variant of recurrent neural network can model the long-term dependency in a series of data and thus is the promising way for degradation prognosis. CNN is built upon multiple layers of filters, which are convolved with the inputs to iteratively extract the features. This mechanism is similar to that of digital filters. The difference is that the weights of convolutional filters are adaptively learned from the inputs, rather than manually designed. CNN is typically used in supervised classification and has achieved state-of-the-art performance for fault diagnosis. In particular, 1D CNN can achieve competitive diagnostic performance as other deep learning methods while having much less computational complexity. Therefore, it is suitable for real-time condition monitoring and fault diagnosis [40,41].

As pointed out in the relevant review papers [34-41], despite the progress of deep learning methods in machine fault diagnosis, several major challenges remain in applying the deep learning algorithms in the real-world industrial environment. First, deep learning methods lack interpretability. The deep learning models are commonly trained and experimentally validated on the limited collected data. This is unacceptable for safety-related applications. Risk assessment procedures should be investigated for a fair assessment of deep learning models. Second, deep learning models are prone to overfitting training data and may fail on test data whose distribution is largely different from that of training data. The distribution shift, also called the covariate shift, can be caused by variations of machines' operating conditions. It also occurs when the models are deployed on similar but different machines. Therefore, the robustness and generalization ability of deep learning diagnostic models should be investigated and improved. Third, the real-time capability of deep learning models is mostly demonstrated on powerful workstations. However, many monitoring tasks are

conducted on embedded systems, which have limited computational power. The models should be lightweight and executed in real-time on the target platform. Fourth, most diagnostic models can only recognize the known failure modes that are defined as the classes and included during the training procedure. However, there could be tens of unseen failure modes in the real-world diagnostic task, which are not included in the training data. It is necessary to tackle this issue, avoiding false alarms or miss detections.

2.3. Track geometry inspection and monitoring

Regular assessment of track geometry is a standard task of track inspection and maintenance. The quality of track geometry is represented by five track geometry parameters defined in the standard EN 13848-1 [42], namely track gauge, cross-level, longitudinal level, lateral alignment, and twist. Depending on the line speeds, the corresponding track sections should be inspected every several months.

2.3.1. Status quo in practice

The standard way for track geometry inspection is based on TRVs, which are equipped with track geometry measurement systems (TGMS). The chord-based and inertial-sensor-based methods are widespread and adopted in most commercial TGMS [43].

The chord-based method relies on a frame, i.e. chord, with transducers moving on the track. For a simplified illustration, Figure 6 shows the chord moves over a sinusoidal irregularity with amplitude Y and wavelength λ in the train-forward direction x. One transducer per rail measures the displacement z between the chord and the rail surface. The measurement point is at a distance αL from the left-hand end. Two chord ends slide along the track. The measured relative displacement $z(x + \alpha L)$ can be derived by:

$$z(x + \alpha L) = y(x) - \alpha[y(x) - y(x + L)] - \mathrm{y(x} + \alpha L) \tag{1}$$

where L is the chord length and αL is the distance between the measurement point and the left end.

The vertical coordinate of a point $y(x)$ on the sinusoidal irregularity can be expressed by:

$$y(x) = Ysin(2\,\pi x/\lambda) = Ysinkx \tag{2}$$

where $k = 2\,\pi/\lambda$ is the wavenumber.

Substituting eq. (2) into eq. (1), we can reach eq. (3).

$$z(x + \alpha L) = Ysinkx(1 - \alpha + \alpha coskL - cosk\alpha L) + Ycoskx(\alpha sinkL - sink\alpha L) \tag{3}$$

It can be expressed as eq. (4) and (5) in the form where the amplitude Z and phase φ are functions of the wavenumber k and the chord length L [44]. Eq. (4) and (5) are termed the transfer functions of the chord-based TGMS.

$$Z = Y[(1 - \alpha + \alpha coskL - cosk\alpha L)^2 + (\alpha sinkL - sink\alpha L)^2]^{1/2} \tag{4}$$

$$\varphi = tan^{-1}[(\alpha sinkL - sink\alpha L)/(1 - \alpha + \alpha coskL - cosk\alpha L)] \quad (5)$$

For a given λ, the amplitude Y can be calculated by the measured Z and the known L. In some cases, however, the amplitude Y cannot be obtained, as its amplitude gain is always zero. For instance, the chord length L is an uneven multiple of the irregularity wavelength λ, and the measurement point is at the middle of the chord, i.e. $\alpha = 0.5$. This intrinsic problem is the main disadvantage of chord-based TGMSs.

Most modern chord-based TGMSs, such as the ones provided by Mermec [45], employ multiple non-contact laser sensors, which have no limitations on vehicle speeds. In harsh weather conditions, such as snow, mechanical transducers are preferred, which cannot be operated at very high speeds. For instance, Infranord measurement vehicles (IMV) used by the Swedish operator Trafikverket have two variations. IMV100 is equipped with mechanical transducers and operated at a maximum speed of 100km/h for snowy days. IMV200 is equipped with optical transducers and operated at a maximum speed of 200km/h for normal days [46].

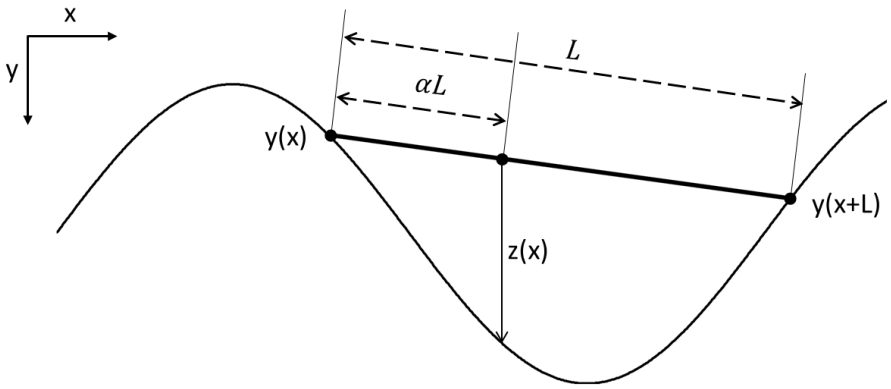


Figure 6 Principle of the chord-based method (modified based on [44])

The inertial-sensor-based TGM method determines the track geometry parameters from the vertical and lateral position of the rail relative to an inertial reference. The inertial reference refers to an inertial measurement unit (IMU), composed of accelerometers and gyroscopes for the measurement of accelerations and angular velocities. The IMU is commonly installed on a vibration-isolated platform attached to the carbody underframe. The optical sensors measure the relative displacement of the IMU reference relative to the rails in the vertical and lateral directions and scan the rail profiles simultaneously. As exemplarily shown in Figure 7, most commercial TGMSs have followed this measurement principle due to simple installation. The main supplies on the market are listed in a technical report [47], including AVANTE, DMA, Mermerc, etc. The TGMS used by DB Netz is more complicated. It involves two additional measurement units mounted on the axleboxes of the front and rear bogie, which measures the axlebox accelerations and the relative displacement between the wheels and the rails. The measurement chain between the IMU reference and the two axleboxes units has been established by laser sensors. In this way, the TGMS can cover longer wavelengths [48]. The main disadvantage of the inertial-sensor-based TGM method is that the system must be operated above a minimum speed, e.g. above 10-30 km/h, avoiding noise and offsets of IMU outputs predominated at low speeds.

In track maintenance practice, infrastructure managers (IM) usually employ several dedicated TGVs to routinely inspect the whole rail network. However, it is difficult to timetable the dedicated TGVs due to their non-regular schedule and track occupation. This problem is exacerbated by the increase in the rail traffic on the track with limited capacity. To cope with this problem, track geometry monitoring on in-service vehicles has been proposed, aiming to assess track geometry quality without interrupting the normal traffic. This will significantly increase the availability of track geometry data and thus increase the reliability of degradation prognosis for PM.

Figure 7 A commercial TGMS based on IMU and laser sensors [45]

2.3.2. State of the art of the research

Track geometry monitoring systems should be robust and affordable for mass deployment on in-service vehicles. There are broadly two approaches. The first approach is to deploy unattended geometry measurement systems (UGMS) on in-service vehicles [49]. Most UGMSs have a similar measurement principle to the IMU-based TGMS, with a more compact and lightweight structure design. Escalona et al. [50] proposed a UGMS, comprising two video cameras, two laser line-projectors, an IMUS and a rotary encoder. The track geometry was derived based on the kinematic chain between the inertial reference and the rail position, using the position and orientation of the projected laser line on the railheads as well as the acceleration and angular velocity acquired by the IMU. Peng et al. [51] developed a similar UGMS with two video cameras per rail. This UGMS can obtain lateral alignment, longitudinal level as well as track gauge and detect rail surface defects. The main advantage of UGMSs is that they are compliant with EN 13848 standard series. The measured track geometry parameters can be used to determine track quality indicators (TQI) defined in the standard EN 13848-6 [52]. These TQIs form the basis for the evaluation of track geometry quality classes and the planning of maintenance actions in practice.

However, the complex UGMSs require high investment and operating costs, preventing widespread deployment on in-service vehicles. Many studies attempted to reconstruct track geometry merely with vehicle dynamic responses measured by accelerometers. Axlebox accelerations were often used to identify the vertical rail profile by double integration of accelerations [49]. They usually contain much high-frequency vibration noise and offset drifts, resulting in accumulated integration errors. Signal processing techniques are required to tackle this issue. For instance, Xu et al. [53] used a high-pass filter with a very low cutting-

off frequency to remove zero-shift of accelerations and resampled accelerations in the equally spaced spatial domain prior to double integration. A successful practical application is the use of an Intercity Express train in Germany to monitor longitudinal irregularities [54]. However, track alignment cannot be derived by this approach, since the wheel does not follow the lateral alignment exactly as in the vertical direction. Agh [55] experimentally investigated the correlation between the axlebox accelerations and the track geometry parameters measured by a TGV on a straight line. The vertical axlebox acceleration was significantly correlated with the second-order derivative of the decolored longitudinal level. In contrast, no significant statistical relationship between acceleration and alignment in the lateral direction was observed. To address this problem, researchers attempted to apply model-based approaches. A vehicle dynamic model was often used to convert the measured vehicle dynamic response into the estimated track geometry. Ripke et al. [56] combined acceleration measurements with an MBS model of the vehicle, on which the accelerometers were installed. The alignment was estimated by accelerations and then corrected by the MBS model using a dedicated correction mechanism. The estimated alignment was compared with the one measured by a TGV. However, this approach was vitiated by the comparison results. Rosa et al. [57] and Munoz et al. [58] involved a Kalman filter along with the vehicle model for alignment estimation. The proposed methods were validated on a vehicle running on a straight track with irregularities. The main obstacle of model-based approaches in practical use is the prior knowledge of the relevant parameters for modeling. In addition, the up-to-date wear status of the vehicle components, especially wheels, can hardly be considered during modeling, which has a significant impact on vehicle dynamic responses. In recent work [59], a deep learning approach based on Wasserstein generative adversarial network (GAN) was employed to reconstruct longitudinal and lateral irregularities from axlebox accelerations. However, this method was merely tested on simulation data. The abovementioned studies rely on axlebox accelerations to estimate the track geometry. It is difficult to maintain electrical systems on axleboxes which are subject to very high vibration levels. A more robust solution should have monitoring systems on bogies or even on carbodies [49]. However, the isolation effect of suspension systems prevents the intuitive double integration of accelerations. Obrien et al. [60] reconstructed the track longitudinal level from the vertical bogie acceleration and angular velocity by using cross-entropy optimization. Xiao et al. [61] designed a Kalman filter for estimation based on carbody accelerations. Li et al. [62] applied deep learning techniques on vertical carbody accelerations to estimate track longitudinal irregularities. An AE model was trained on irregularities to obtain latent representation, to which an additional estimator was trained to project carbody accelerations. In a second stage, a Bayesian deep learning model was trained to reconstruct irregularities from latent representation outputted by the estimator.

The second general approach for track geometry monitoring avoids the reconstruction of geometry parameters. Instead, it aims either to detect discrete rail/track failures which induce large vehicle dynamic responses or to represent the track quality by vibration levels. Bolouchi et al. [63] proposed to use multi-resolution analysis based on continuous wavelet transform (CWT) to decompose carbody accelerations into several frequency bands. Different track or rail failures can be detected and distinguished according to magnitudes of frequency spectra in the corresponding ranges. In addition, the standard deviation of acceleration was used to represent track quality. Similarly, Tsunashima et al. [64] tested CWT and EMD to detect track faults based on carbody accelerations. Furthermore, Tsunashima [65] conducted a machine learning approach for the diagnosis of track geometry defects. The vertical and lateral

accelerations and the roll rates were acquired on the carbody. The RMS value over a short period of the raw data was extracted as the feature. Afterward, a support vector machine (SVM) was employed as the classifier for fault diagnosis. Kaewunruen [66] used RMS amplitudes of axlebox accelerations in the specific frequency range to inform the growth of rail corrugation. Chudzikiewicz et al. [67] defined an indicator, which is similar to the ride comfort index, based on the vertical axlebox accelerations to approximate TQI for track quality classification. Rosa et al. [68] proposed to train a classifier to detect large track lateral irregularities based on the RMS features.

In a summary, the main research interest is to develop a robust and affordable monitoring system on in-service vehicles for track quality assessment. Following the existing standards and rules of track maintenance, the first approach aims to reconstruct track geometry parameters with inexpensive and robust sensors. Using Accelerometers along with advanced data processing techniques has become the most promising solution. The main obstacle lies in the geometry reconstruction in the lateral direction due to the complex lateral dynamics of railway vehicles. In the second approach, the research focuses on identifying a feature from the acceleration that can substitute TQIs for track quality assessment. In this way, the reconstruction of track geometry parameters can be avoided. However, this approach is not compliant with the existing standards.

3. Research Questions and Specific Objectives

From the perspective of maintenance needs and boundary conditions in harsh railway environments, we have identified two challenges for condition monitoring and fault diagnosis, namely real-time data processing and robustness. We attempt to tackle them in the application scenarios of WFD and track geometry monitoring. In addition to the two general challenges, there are some unsolved issues in these two applications. This motivates us, on the one hand, to tackle the challenges related to real-time processing and model robustness. On the other hand, we attempt to solve application-specific issues. For WFD, our method only needs the carbody acceleration as the input and can be executed in real-time on embedded systems. For track geometry monitoring, we propose a computer-vision-based solution to support the reconstruction of track alignment, which only needs a cheap off-the-shelf camera per rail.

Our research has been included in three publications, which are the main constituents of this dissertation. The publications will answer the following research questions.

Pub 1.	D. Shi, Y. Ye, M. Gillwald, M. Hecht (2021), "Designing a lightweight 1D convolutional neural network with Bayesian optimization for wheel flat detection using carbody accelerations", International Journal of Rail Transportation, 9:4, 311-341, DOI: 10.1080/23248378.2020.1795942
RQ 1.	**Can wheel flat be detected in real time on embedded systems using carbody accelerations?** The previous studies mostly used axlebox accelerations for WFD. However, it would be beneficial to exploit the existing telematics devices installed on the carbody for WFD, despite its difficulty due to the isolation effect of suspension. Telematics devices as embedded systems have limited computation power, which brings additional challenges for real-time computing. In this sense, we propose a lightweight 1D CNN for the end-to-end WF, which requires much less computational complexity and can be executed on embedded devices.
Pub 2.	D. Shi, Y. Ye, M. Gillwald, M. Hecht (2022), "Robustness enhancement of machine fault diagnostic models for railway applications through data augmentation", Mechanical Systems and Signal Processing, Volume 164, 2022, 108217, ISSN 0888-3270, https://doi.org/10.1016/j.ymssp.2021.108217.
RQ 2.	**Are the algorithms for wheel flat detection robust to variable railway operating conditions?** The previous studies were mostly conducted on simulation and laboratory data. The robustness of the proposed algorithms in the real-world environment has not been investigated. To fill this gap, we conduct an empirical study of model robustness based on a large amount of field data collected on different freight wagons running under different operating conditions.
RQ 3.	**How can the robustness of the diagnosis algorithms be improved?** If the diagnosis models are not robust enough to withstand the variations of operating conditions, it is necessary to improve their robustness. We propose to exploit MBS to

	generate simulation data under various operating conditions. This data can be further augmented and fed into the training process of diagnosis models. This data augmentation approach can enhance the models for wheel flat detection.
Pub 3.	D. Shi, et al. (2022), "Deep learning based virtual point tracking for real-time target-less dynamic displacement measurement in railway applications", Mechanical Systems and Signal Processing, Volume 166, 2022, 108482, ISSN 0888-3270, https://doi.org/10.1016/j.ymssp.2021.108482.
RQ 4.	**How can track alignment be monitored in real time by using inexpensive sensors?** The previous studies have shown that the lateral acceleration is not directly correlated to the track alignment. We hypothesize that the information on the lateral displacement of the wheel relative to the rail may complement the lateral acceleration to derive the track alignment. Therefore, we propose a novel method based on computer vision and image processing techniques to track the relative wheel movement in the lateral direction in real time. The used video cameras are off-the-shelf available and affordable for mass deployment.
RQ 5.	**Is the proposed method based on optical sensing robust to variable railway operating conditions?** The main concern of optical sensing is its robustness against severe weather conditions. Despite the lack of real data, we propose synthetic image corruption to simulate different weather conditions and use the synthetic images to augment the training dataset for robustness enhancement.

The accepted manuscripts of the three publications are presented in Section 4, 5 and 6. The research questions are discussed in Section 7 based on the experiment results.

4. Wheel Flat Detection Using Carbody Accelerations

This publication presents the proposed lightweight 1D CNN for real-time WFD using carbody accelerations. It addresses the specific challenge of WFD at the carbody level, where wheel flat signatures are much weaker than those on axleboxes or bogies. Furthermore, the publication shows the computational complexity of the designed lightweight CNN is even lower than that of common signal processing algorithms in terms of floating-point operations (FLOP). It can be executed in real time on an embedded device. The proposed method has been validated on the measurement data collected on a freight wagon running under normal operating conditions. Its superiority was verified by comparison with several SOTA lightweight CNN for machine fault diagnosis. In addition, the mechanism of CNN has been investigated by visualization of the learned interim features within the hidden layers. As hypothesized, CNNs work similarly to signal processing methods. It can hierarchically suppress noise and spotlight the periodic impulsive pattern of a wheel-flat signal.

The accepted manuscript below is an article published by Taylor & Francis in International Journal of Rail Transportation on 24th July 2020, available online: https://doi.org/10.1080/23248378.2020.1795942

Designing a lightweight 1D convolutional neural network with Bayesian optimization for wheel flat detection using carbody accelerations

Dachuan Shi *, Yunguang Ye, Marco Gillwald, Markus Hecht

Institute of Land and Sea Transport Systems, Technical University of Berlin, Berlin 10587, Germany

* Corresponding E-mail: dachuan.shi@tu-berlin.de, Tel.: +49 030 314 79806 and Fax: +49 030 314 22529

Abstract

A large number of freight wagons in Europe have been recently equipped with embedded systems (ES) for vehicle tracking. This provides opportunities to implement real-time fault diagnosis algorithms on ESs without additional investment. In this paper, we design a 1D lightweight Convolutional Neural Network (CNN) architecture, i.e. LightWFNet, guided by Bayesian Optimization for wheel flat (WF) detection, which is a common failure on wheel surfaces. We tackle two main challenges. 1) Carbody accelerations have to be used for WF detection, where the signal-to-noise ratio is much lower than that on axleboxes and thus WF detection is much more difficult. 2) ESs have very limited computation power and energy supply. To verify the proposed LightWFNet, the field data measured on a tank wagon under operational conditions are used. In comparison to state-of-the-art lightweight CNNs, LightWFNet is validated for WF detection by using carbody accelerations with much lower computational costs.

Keywords: Wheel Flat, Fault diagnosis, Machine Learning, Convolutional Neural Network, Bayesian Optimization

1. Introduction

1.1 Background

In-service wheelsets are visually inspected by authorized persons during operational processes of freight transport. Operators do not notice that their wagons are suffering from wheel flats (WFs) until visual inspections are carried out. The inspection time for a wagon is very limited, depending on the wagon type and the axle numbers. For a 4-axle wagon, it usually only takes about five minutes to inspect the whole wagon, incl. loading conditions, completeness of the wagon, conditions of the wagon components, etc. [1]. The detection probability significantly depends on environmental conditions such as darkness and weather, physical and mental conditions of inspectors and visibility of wagon components due to constructive design or contamination. In this sense, it is not easy to identify WFs. Especially, a small WF can be hardly detected by human inspections. On the other hand, a small WF could be neglected even when it is found by inspectors since it does not cause any safety issue. According to the General Contract for Usage of wagons [2], the decisive criterion of wheelset maintenance is the length of wheel flats. The wheelset has to be taken when the flat length exceeds 60 mm (for wheel diameter greater than 840 mm). A small flat e.g. with a length of 20 mm could exist on the wagon for a long time, although they can result in non-negligible negative effects such as periodic large impacts on wagons as well as track and pulse-shaped noises, as illustrated in Figure 1. Therefore, it is desired to use condition

monitoring solutions for early detection of WFs, allowing operators to optimize their maintenance planning and take maintenance measures as early as possible.

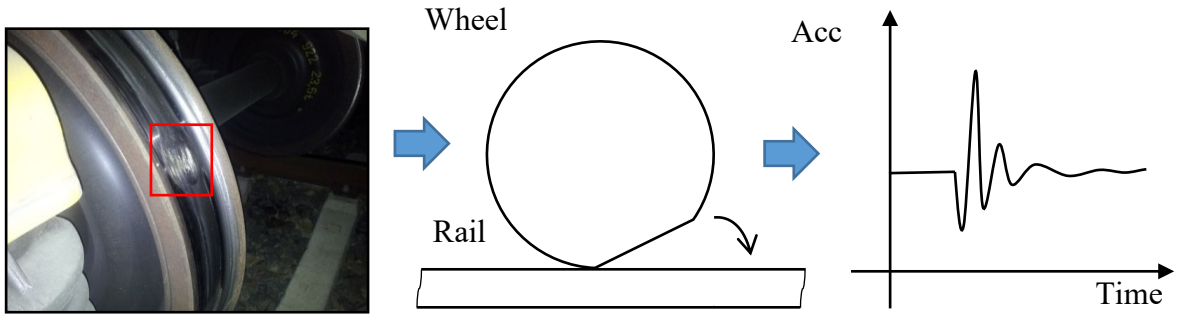


Figure 1. Wheel flat on the running surface of a wheel and illustration of the impact induced by wheel flats

However, freight wagons are not electrified. A comprehensive condition monitoring system that requires grid power is currently not applicable to freight wagons in practice. This limitation of condition monitoring applications on freight wagons is also pointed out in [3]. A potential solution is to use energy harvesting systems (EHS) such as axle generators from Schaeffler [4] and vibration energy harvesters from Perpetuum [6] to power monitoring systems installed on wheelsets. A four-axle vehicle is equipped with eight units of the EHS enabled monitoring systems. However, these commercial systems, which have been already applied on passenger trains, require high investment costs. Due to low margins of freight transport, their price is not acceptable for freight wagons.

a b

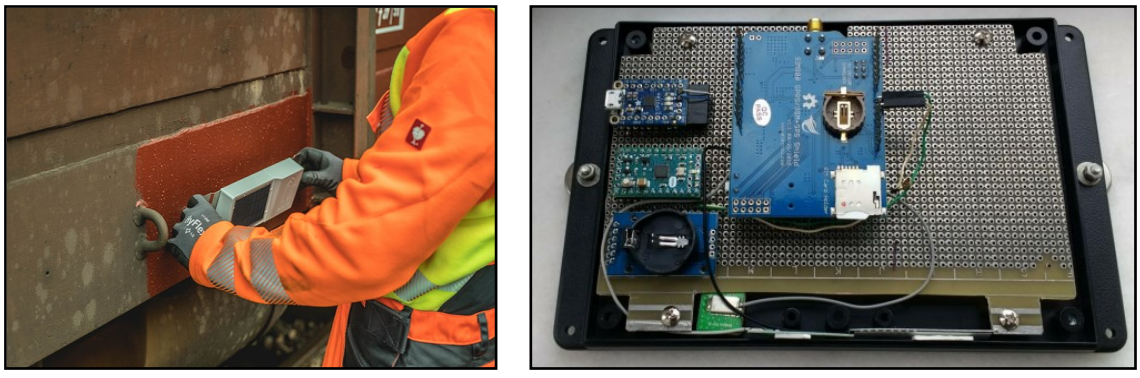


Figure 2. Example of the telematics device: (a) commercial device powered by solar energy [9]; (b) PCB prototype under development [10]

In current practice, the solution is to use a compact embedded system powered by batteries or solar energy for the basic function of tracking and monitoring. Such a system is called “telematics device”, shown in Figure 2. With the trend of digitalization, more and more freight wagons in Germany, Switzerland and Austria are equipped with telematics devices. For instance, DB Cargo will equip 19,000 freight wagons with telematics devices by the end of 2018 and will equip the entire wagon fleet of ca. 70,000 wagons by 2020 [8]. A telematics device is an onboard monitoring device and usually contains a three-axle accelerometer for shock detection and a GPS module for wagon tracking. The whole device is controlled by a

microcontroller like Arduino Uno shown in the right photo of Figure 2 or a more powerful mini-computer like Raspberry Pi and NVIDIA Jetson Nano. A microcontroller can merely ensure data acquisition and communication, whereas a mini-computer enables a more advanced onboard data processing. This provides an opportunity to deploy a WF detection model for real-time WF detection with the support of existing software frameworks such as TensorFlow Lite and JetPack. In this paper, we focus on the development of a lightweight WF detection model. The hardware design and the deployment of the model are out of scope.

1.2 State of the art

WF detection has been investigated over the decades. In the light maintenance process, the condition of wheels is mostly visually inspected. During the overhaul, the entire wheelset is detached from the vehicle and examined by nondestructive inspection techniques in the workshop such as ultrasonic testing and magnetic particle inspection. In terms of the automatic inspection during the train operation, WFs can be detected either by wayside or onboard condition monitoring systems. The most common condition monitoring approaches for the detection of railway wheel defects are viewed by Alemi et al. [11], incl. strain gauge based systems, fiber bragg grating based systems, ultrasound-based systems, vibration monitoring, acoustic emission monitoring, laser techniques and computer vision systems. In practice, wayside systems are owned and installed by infrastructure managers for ensuring the safe operation of railway traffic. The obtained information of rolling stock failures is rarely shared with wagon owners for maintenance optimization of rolling stock. The information density in terms of one vehicle relies on the number of monitoring systems across the network and the operation plan of the individual wagons. Continuous monitoring of the individual vehicle can be hardly achieved.

In comparison to wayside approaches, onboard approaches are more suitable for condition-based maintenance of rolling stock. Onboard systems can continuously collect the condition data, enabling not only fault diagnostics but also predicting the degradation of vehicle components. The common onboard solution for WF detection is to install accelerometers on vehicles for vibration condition monitoring, due to the low price and robustness of the accelerometers. The accelerometers can be installed on axle bearing housing, bogie frame or carbody underframe. A majority of the previous studies intended to use axlebox accelerations for WF detection. The focuses of the research lie in three areas. (1) advanced signal processing methods are used to eliminate the signal interference in order to spotlight the faulty signal patterns of WFs. Jiang et al. [12] proposed to use empirical mode decomposition (EMD) to decompose the raw signal into several intrinsic mode functions that separate the failure signal mode from interferences. Zhao and Shi [13] analyzed the signal in the high-order spectrum that can suppress Gaussian noise. Li et al. [14] proposed adaptive multiscale morphological filtering for denoising. Liang et al. [15] analyzed WF vibration signals by three commonly used time-frequency analysis methods, i.e. short-time Fourier transform (STFT), Wigner-Ville transform and wavelet transform (WT). It was concluded all three methods could present time-frequency information of a wheel flat. (2) The low-dimension features (also called as “indicators” or “health index” in some articles) are defined to represent the faulty signal patterns of WF. Gericke [16] constructed 27 features of axlebox accelerations in the time domain, FFT spectrum, envelope spectrum and Cepstrum for wheel flat detection on freight wagons. Apart from the generic statistic features such as RMS value, mean value,

skewness factor and kurtosis, a series of specific features were defined to represent the specific characteristics of the WF patterns at axlebox level. Bosso et al. [17] defined a WF severity index in the time domain, combining RMS and peak values. This index was validated in a field trial for axlebox accelerations. Bernal et al. [18] used the multibody dynamics simulation to investigate the detectability of WF for Y25 railway freight wagon using axlebox, bogie and carbody respectively. The proposed indicators based on time-domain RMS values, time-domain crest values and dominant frequency in the envelope spectrum can correctly reflect the abnormality caused by WF in axlebox and bogie accelerations. (3) An automatic diagnostic method is proposed for WF detection. The expert-system-based solution is to define a threshold of the defined features based on the statistic observation. In the machine learning (ML) approach, the diagnostic method usually refers to a classification algorithm. Gericke [16] tested several classification methods. Naïve Bayes classifier, k-nearest neighbor classifier, decision tree and neural network (NN) were trained based on the selected features, in order to find the best combination of the features and the classifier. Regarding WF detection by using carbody accelerations, the methods proposed in [17] and [18] were proved to be less effective on carbody accelerations. In our previous work [19], gradient tree boosting and random forest were applied on the several specific features defined in the envelope spectrum and statistic features defined in the time domain, with the help of additional information on vehicle speeds. The trained models were tested in different speed ranges and have achieved an average accuracy of 82.74%.

The majority of the previous WF detection methods work well on axlebox and bogie accelerations, but not on carbody accelerations. The conventional ML approach for carbody accelerations in [19] relies on envelope analysis and several handcrafted features, which has not achieved high accuracy but has a high computational cost. To tackle this issue, we propose to use deep learning (DL) approach, i.e. lightweight convolutional neural network (LCNN), to increase the diagnostic accuracy with lower computational complexity. DL has been rapidly developed in the ML community and widely deployed for machine fault diagnosis (MFD) in the last decade. For the DL applications on mobile and embedded devices, the resource-limited application of DL models is becoming an important research topic. In the ML community, the researchers have been working towards three areas, i.e. hardware specializing such as TPU [20], model compression [21] and lightweight network design. Especially for CNN, an optimized design of the LCNN architecture could significantly reduce the computational costs and achieve a good trade-off between performance and complexity.

SqueezeNet [22], Xception [23], ShuffleNet [24] and MobileNet [25] are the most famous pioneers of LCNN architectures for computer vision applications. They have been further improved as ShuffleNet v2 [26], Mobilenet v2 [27] and Mobilenet v3 [28]. The strategies for lightweight NN can be summarized as follow. (1) Decreasing the parameter number in the network while attempting to preserve network performance. For instance, SqueezeNet replaced the majority of 3×3 filters with 1×1 filters and decreased the number of input channels to 3×3 filters by using the proposed fire module. MobileNet proposed two global hyper-parameters allowing scaling of the model size according to the computation resource. ShuffleNet used group convolution where the input channels are divided into several groups and convolution is performed independently for each grouped channel. (2) Using depthwise separable convolution (DSC) instead of the regular convolution. Xception replaced the

inception module with the DSC module based on the existing Inception V3 architecture. MobileNet built a streamlined architecture based on DSC. (3) Maximizing the network performance with the reduced network size. ShuffleNet proposed channel shuffle, changing the order of the channels, which can compensate for the reduced interaction between different channels due to group convolution. MobileNet V2 proposed to use linear bottlenecks to avoid manifold collapse caused by nonlinear ReLU, and inverted residual connection to improve memory efficiency. MobileNet V3 added the squeeze and excite (SE) in the residual layer to improve the channel interaction and use network search to automatically optimize the network architecture. (4) Optimizing design by taking into account other evaluation metrics. Most LCNN architecture design is guided by computational complexity. ShuffleNet V2 proposed to reduce memory access costs by keeping equal channel width of the input and output channel as well as carefully using group convolution, to reduce the degree of fragmentation and to reduce elementwise operation. Other interesting lightweight designs can be found in [29-35], which followed more or less the aforementioned strategies.

In the area of machine fault diagnosis, the researchers have been attempting to adapt the LCNN proposed for computer vision applications, where the input data is usually 2D metrics with a small width. For instance, the image size of the ImageNet dataset is 256×256. The CIFAR-10 dataset consists of 32×32 images. In comparison, the sensor data for MFD is mostly a 1D array of acceleration with a high sample frequency (over 1 kHz). To reduce the dimensionality of the input data, the first way is to use signal processing methods to transfer 1D sensor data into a 2D time-frequency matrix through e.g. WT and STFT. Liu et al. [36] used STFT for preparing 2D input data and adapted ShuffleNet V2 with batch normalization and L2 regularization for feature extraction and classification. The second way is to use 1D CNN instead of 2D CNN [37-39]. Regarding the reduction of input array size, Ma et al. [39] used wavelet packet transformation (WPT) to obtain multiscale 1D wavelet coefficients as the input data. The different scales of wavelet coefficients were fed into LCNN as the different channels. The final input data is reduced to a tensor of 64×1×16 (16 denotes the number of channels). The proposed LCNN is a streamlined CNN, being composed of small-size filters (3×1 and 1×1). Similar work was presented in [38], where the activation function of CReLu was introduced into the DSC blocks to further reduce the number of parameters. Although the CNN architectures in the mentioned studies were optimized in terms of computational complexity, the costs of the used signal processing techniques such as WT and STFT have not been taken into account. The overall complexity of the diagnosis model was not discussed. In contrast, Wang et al. [37] directly fed the raw data with the dimension of 12800×1×2 to the network for remaining useful life (RUL) prediction of machinery. The proposed LCNN is similar to MobileNet, being composed of DSC blocks with the added SE units. Although it avoided signal processing, it did not take measures to reduce the size of the input raw data (i.e. 12800 sample points) and the large filter size. The large width of the input array and filter size can result in high computational costs during the convolution operations.

1.3 Current gaps and our contributions

Based on the literature review on WFD methods and LCNN as well as our experience with field data in real operating conditions, the following two gaps have been identified.

(1) Lack of understanding on application complexity.

The majority of the proposed MFD methods are validated by the laboratory datasets and have achieved more than 99% hit rates e.g. in the CWRU bearing fault datasets, which can hardly represent the complexity of the real application environment. In the case of WF detection by using telematics devices, we are facing a two-fold complexity. The first one is related to pattern recognition. The telematics devices are usually placed on the wagon body so that carbody accelerations should be used as the input data. The vibration on the wheel is transferred via axle bearing to axlebox, further via primary suspension to bogie frame and finally via secondary suspension (e.g. centre bowl and side bearers in case of freight wagons) to carbody. The longer the vibration transfer path is, the lower the vibration level can be measured. Especially on freight wagons, the primary suspension between axlebox and bogie frame as well as the center bowl/side bearers between bogie frame and carbody are strongly nonlinear due to friction damping [40]. Due to suspension isolation, impulsive signals caused by WFs are damped significantly in carbody accelerations and thus could be buried in signal noise and disturbances. Figure 3 shows the typical vertical accelerations of WF signals in the time domain and envelope spectrum at axlebox, bogie and carbody at the vehicle speed of 45 km/h respectively. The measurement was performed on a tank wagon with Y25 bogies running on a mainline. The peaks caused by WF in axlebox and bogie accelerations are clearly visible in the time domain, whereas the WF signature is almost hidden in the noise in carbody accelerations. In the envelope spectrum, the rotation frequency and its harmonies are clearly recognizable at the axlebox and bogie levels. At the carbody level, only the first and second-order of the rotation frequency has a much higher magnitude. This pattern is, however, unstable due to disturbances in the real operational conditions. For instance, abnormal track irregularities could result in strong vibration and thus bury the WF peaks, especially at the carbody level. The discrete impacts caused by e.g. rail squats, bad rail welds and turnouts may disturb the periodic pattern of WF within a limited time window. This finally results in miss detection. On the other hand, these discrete impacts form several consecutive peaks in the acceleration signal, being similar to WF patterns. This may cause a false alarm.

The second complexity concerns the hardware restriction of the telematics device. The processing unit of a telematics device is a mini-computer like Raspberry Pi. For instance, the commonly used Raspberry Pi 3 Model B has a quad-core 1.2GHz 64bit ARM CPU and 1GB RAM. It allows real-time CNN inference with state-of-the-art CNN models for computer vision tasks. However, the power consumption increases significantly from 1.3W in the idle mode to around 3.5W when executing a CNN model, depending on the individual models and software frameworks [41]. Other components of the telematics device such as the GPS module, GPRS module for remote data transmission, and local wireless transmission module consume hundreds of milliwatts [42]. A low-profile solar panel of the size 16×10 cm for a telematics device can provide around 15Wh/day [42], powering the device only for about three hours in continuous operation. This means that the continuous measurement and processing of accelerations cannot be performed. A typical telematics device is designed to wake up, for instance, every half hour to perform a one-second measurement and the subsequent data processing. The raw acceleration data will be discarded. Only the essential parameters such as diagnosis results, GPS values and timestamps are wirelessly transmitted to the cloud via the mobile network. In practice, the power supply system based on the (solar) battery is usually designed to last one year at least, taking into account the inspection interval of freight wagons (which is quite different from passenger vehicles). Therefore, it is necessary

to design a lightweight diagnostic model, requiring less computation power and thus less power consumption.

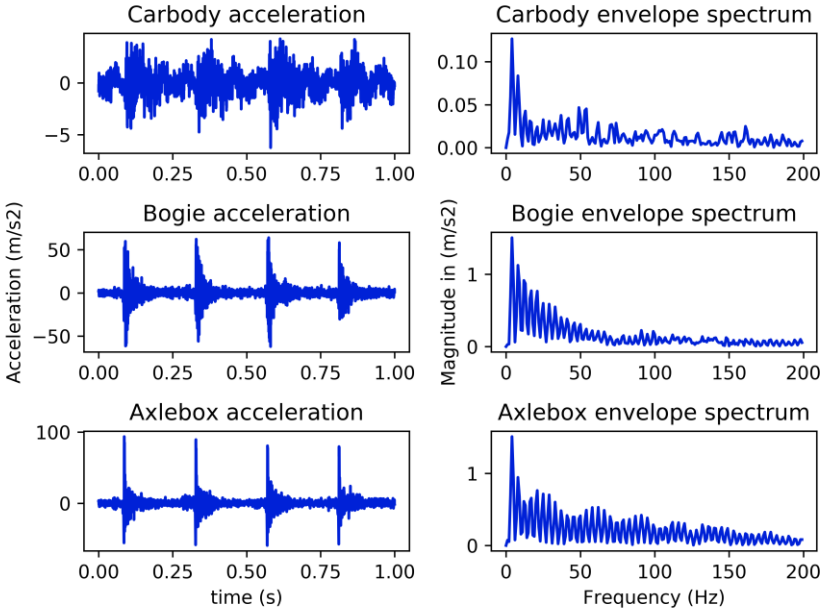


Figure 3. Comparison of carbody, bogie, and axlebox acceleration signals and their envelope spectrums

(2) Lack of a systematic approach for designing a lightweight diagnosis method.

A model for fault diagnosis typically contains signal processing and classification. It should take into account both processes to design a lightweight diagnosis method. The computational time complexity of several common methods for signaling processing and classification is listed in Table 1. The big O notation expresses the asymptotic behavior of time complexity, where N is the size of the inputs. It should be noticed that the computational complexity for classification refers to inference complexity, rather than training complexity. It has been commonly thought that ML models have very high computational costs. It is true when it refers to training complexity. Once the models are trained and deployed, the trained models have much less computational complexity for inference. Its complexity is comparable to that of signal processing techniques, as shown in Table 1. The frequency analysis techniques like FFT, Hilbert transformation and EMD have even much higher complexity, especially when the input size is large. It is quite challenging to perform such frequency analysis on a low-consumption embedded system. The complexity of LCNN is not listed since it significantly depends on its architecture. A well-designed LCNN could have less complexity than a signal processing method. (This will be shown in Section 4.3.) Therefore, the computational costs of both signal processing and classification should be considered. If the signaling processing could be avoided, it will save many computational resources.

Table 1. Computational complexity of several common methods for signaling processing and classification

Signal processing methods	Computational complexity
Fast Fourier Transformation [43]	$O(N\ log(N))$

Hilbert Transformation	$O(N\ log(N))$
Continuous Wavelet Transformation [44]	$O(N)$
Discrete Wavelet Transformation [45]	$O(N)$
Empirical Mode Decomposition [47]	$O(N\ log(N))$

Classification methods	Computational complexity
Random Forest [48]	$O(N)$
Support Vector Machine (Kernel) [49]	$O(N)$
Naïve Bayes [50]	$O(N)$

To cover the gaps, we propose a process to design a LCNN architecture for MFD using vibration data. The designed LCNN is demonstrated for WFD by using carbody accelerations in this work. To be more specific, our contributions are summaries as follows.

1. Proposing a systematic design process of LCNN. It starts with turning a simple CNN for the specific input data. Lightweighting is guided by the strategies that are commonly accepted in the ML community [51]. Bayesian optimization with Gaussian process (BOGP) is introduced for identifying the proper parameters and network structure. As the model performance may not linearly change with the linear variation of the individual parameters and the number of parameter combinations is too large, the conventional grid-search-based parameter variation could not help find the best combination of parameters. After preliminary tuning, some parameters could be manually adjusted to achieve a better tradeoff between performance and computational complexity.
2. Designing a mini-size modular LCNN architecture for WF detection, called LightWFNet. The state-of-the-art (SOTA) CNN blocks such as DSC, SE and bottleneck as well as the activation function hard swish are introduced in the architecture. Its performance and computational complexity are compared with the SOTA LCNN for MFD and the classic time-frequency analysis.
3. Understanding the mechanism of feature extraction within CNN through visualization of interim layer outputs. CNN works as a combination of diverse adaptive filters to decompose the input data. This mechanism is visualized and compared to the standard frequency and time-frequency analysis.

In the following, the theoretical basis is introduced in Section 2. Section 3 explains the design process and the proposed LightWFNet. The experiments based on field data and the analysis of the experiment results are presented in Section 4. Section 5 draws the conclusions.

2. Convolutional neural network and Bayesian optimization

The modern multilayer CNN architecture was proposed by LeCun for a computer vision application [52]. This basic CNN architecture, called as LeNet, consists of multiple layers of convolution, pooling and activation in a row for feature extraction as well as one or more

subsequent fully connected layers and a softmax function for classification. In order to learn a complex pattern, non-linearity is introduced by the activation function, which is a nonlinear transform function such as tanh, ReLu and sigmoid function. Apart from the basic elements, batch normalization has also become a standard operation within the modern CNN architecture. Batch normalization is to standardize the feature maps within hidden layers by subtracting the mean and then dividing the standard deviation. This was introduced in [53] to solve the problem of internal covariance shift within the feature maps. The arrangement of these standard elements is essential to design a well-performed CNN architecture. This section will introduce the specific core elements used in LightWFNet as well as BOGP for hyperparameter tuning.

2.1 Depthwise separable convolution

Replacing the regular convolution with depthwise separable convolution (DSC) [25] is an effective lightweighting strategy. DSC converts the regular convolution into a depthwise convolution and a pointwise convolution. The regular convolution is executed by simultaneously sliding a filter over all input channels and combining the results in a single step. In contrast, the depthwise convolution is executed by sliding a filter over each channel. Afterward, the pointwise convolution combines the outputs of the depthwise convolution. In this way, the number of convolution operations and parameters is significantly reduced. In the following, the operation of a regular convolution and a DSC are explained in detail.

The regular convolution is executed by sliding the filters over the input array simultaneously in all channels. In each sliding step, the filter convolves the receptive fields of the input feature map in all channels, which have the identical size as the filter. Given a 1D multi-channel input feature map of size $N \times 1 \times J$ and M convolutional filters of size $K \times 1 \times J$, where N is the array length of the input feature map, K is the size of the filter and J is the channel number, the regular convolution requires the following numbers of multiply-accumulate (MAC) operations.

$$MAC_{conv} = M \cdot J \cdot K \cdot (N - K + 1) \tag{1}$$

The DSC separates the convolution operation into a depthwise convolution and a pointwise convolution. Given the input array of size $N \times 1 \times J$, there should be J filters of size $K \times 1$ for the depthwise convolution. The convolutional operations are executed by sliding each filter over the corresponding channel of the input array. In each sliding step, one filter convolves the receptive field in the corresponding channel, which has the identical size as the filter. The output arrays have thus the dimension of $(N - K + 1) \times 1 \times J$. In the next step, the pointwise convolution is executed by using a filter of size 1×1 that iterates through every single point and linearly combines the depthwise channel. The final results meet the results of the regular convolutional operation. However, the DSC uses much fewer MAC operations than the regular convolution so that computation complexity is reduced. The MAC of DSC with M pointwise filters is obtained by eq. (2). We get thus the reduction using eq. (3).

$$MAC_{DSC} = J \cdot K \cdot (N - K + 1) + (N - K + 1) \cdot J \cdot M = (N - K + 1) \cdot (M + K) \cdot J \tag{2}$$

$$\text{Ratio} = (M + K)/(M \cdot K) \tag{3}$$

As the filter number M is usually much greater than K, the reduction ratio will be around $1/K$. For instance, using the filters of size $K=3$ for the depthwise convolution, 1D DSC has 2-3 times fewer computation costs than regular 1D convolution.

2.2 Squeeze and excitation

Squeeze and Excitation (SE) is an architectural unit that improves channel interdependencies by weighting each feature map [55]. The SE unit can be added to any CNN building block, causing negligible additional computational costs. Given the input feature maps X of size $N \times 1 \times C$, the feature map in each channel of the input block is squeezed to a scalar value using global average pooling, which describes the statistical feature of each channel. Therefore, the complete output U has the size $1 \times 1 \times C$. This is the "squeeze" process. In the "excitation" block, two FC layers with the non-linear activation functions form a bottleneck structure (the first FC layer has a reduced channel of C/r), which compresses the interchannel information into the fewer channels and rescales it to the original channel size. As the weights of two FC layers are learned automatically during training, this is called adaptive recalibration based on the gating mechanism. The result of the "excitation" process is the scalar per channel. For the notation, we denote the weights of the first FC layer of size $1 \times 1 \times C/r$ as W_1, the weights of the second FC layer of size $1 \times 1 \times C$ as W_2, the ReLu actication function as δ and the sigmoid activation function as σ. The gating scales S can be calculated by eq. (4). The final outputs $\tilde{x}$ are calculated by multiplying the input feature map with the gating scale in each channel, see eq. (5).

$$\mathrm{S} = \sigma(W_2\delta(W_1U)) \tag{4}$$

$$\widetilde{x_c} = s_c x_c \tag{5}$$

2.3 Linear bottleneck block

The skip connection in the residual network [56] has been proved to be very effective for a deep network. An empirical study [57] reveals that the skip connection preserves gradient flow by shortcutting the long paths of a deep network, instead of solving the vanishing gradient problem. The real effective forward path of a deep network is much shorter than the designed one. Shortcutting results in different possible paths during the training process, so that the entire network can be regarded as an ensemble of many paths, rather than a single deep path. This ensemble effect allows that the skip connection also has a positive effect on a normal network for the MFD task, which is not deep as the ones proposed for computer vision applications. To achieve the skip connection, the input layer must have the same dimensionality as the last convolution layer before the addition. The left graph in Figure 4 illustrates the standard residual block. Given the input layer of size $N \times 1 \times C$, there should be C filters within the subsequent convolution layers (Although two convolution layers are shown in the graph for illustration, there could be more convolution layers). The output of the last convolution layer should have the same size of $N \times 1 \times C$ (meaning that convolution should be executed with stride one and padding), so that it can be added with the input and then transformed by the activation function.

In order to reduce computational complexity, regular convolution layers can be replaced with a stack of pointwise convolution and depthwise separable convolution. The first pointwise convolution is used to shrink the channel numbers with the factor s. The subsequent DSP replaces the regular convolution. In this way, the inner blocks are narrower than the outside blocks, like a "bottleneck" structure. It is worth noting that the skip connection will be disabled if any convolution layer is executed with a stride length larger than 1. Sandler et al. [28] proposed using a linear bottleneck to prevent non-linearity from destroying damaging feature maps. The activation function is removed at the end of the bottleneck block.

2.4 Hard version of activation function

The activation function introduces nonlinearity in the neural network. The most accepted activation function is ReLu, which is a very simple piecewise function and can be defined by $f(x) = \max(0, x)$. It either returns the value of the inputs or the value of 0 if the input is a non-positive value. It costs less computational complexity than the classic sigmoid and tanh activation function and avoids the saturation problem. On the other hand, it results in the problem of e.g. dead neurons [58] and bias shift [59]. In recent years, new functions like Swish [60] and Mish [61] were proposed to replace the ReLu function. Swish has been widely used and significantly improves network performance, especially in deeper layers within a network. Swish can be defined by $f(x) = \mathrm{x} \cdot \sigma(\beta \mathrm{x})$, where $\sigma(\mathrm{x}) = (1 + \exp(-z))^{-1}$ is the sigmoid function and β is a constant or a trainable parameter. However, the sigmoid function has much more computational complexity than the ReLu function. Therefore, the hard version of the activation function is proposed to approximate the original function by the combination of piecewise linear functions. The hard sigmoid function can be defined by eq. (8), while the hard swish function is defined by eq. (7) [28].

$$\mathrm{Hsigmoid} = ReLu6(x + 3)/6 \tag{6}$$

$$\mathrm{Hswish} = x \cdot ReLu6(x + 3)/6 \tag{7}$$

where $ReLu6$ is the ReLu capped by the units at 6 and thus is defined by $f(x) = \min(\max(0, x), 6)$.

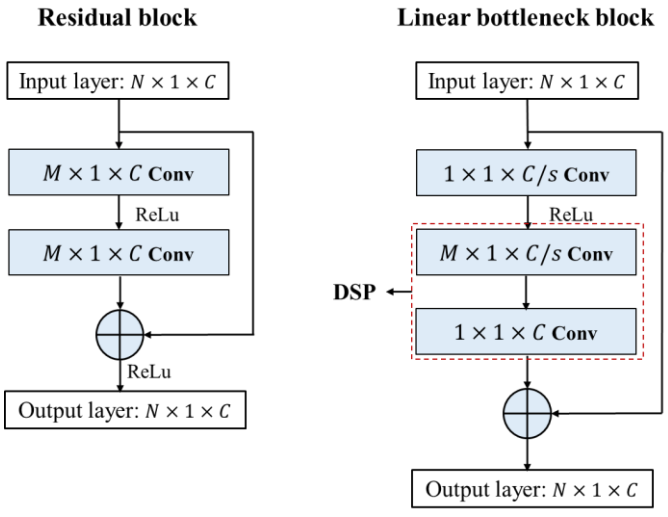


Figure 4. Comparison between the residual block and linear bottleneck block

2.5 Bayesian optimization with Gaussian process

Parameter tuning for neural networks is an optimization problem. Optimization is to find an input $x^* \in X$, where $X \subset \mathbb{R}^d$ and $d \geq 1$, within the limited steps. This input minimizes (or maximizes) the value of an object function $f: X \to Y$. In the case of hyperparameter tuning, an input x refers to the combination of d parameters. The object function is the trained neural network with the parameter combination x. The output value is usually the result of the validation loss (or accuracy). For a shadow network, grid search for each tunable hyperparameter can be done, where a discrete range of each parameter is defined, and the discrete values within the range are exhausted one by one for model training. In such a way, the parameter sensitivity can be investigated and the best combination of the parameters can be identified. In terms of a large network, grid search can hardly be done, since there are too many combinations of hyperparameters. In this case, BOGP can be applied.

BOGP is composed of a surrogate model and an acquisition function. The Gaussian process is the most popular method to build the surrogate model. Given observations $(x_i, y_i = f(x_i))$ for the steps $i = 1{:}\,t$, where t denotes step t, a probabilistic surrogate model is built for the objective function $f(x)$. The prediction of this model at a new query point x_q with a kernel function k_θ (θ is a hyperparameter) is a normal distribution of $y_q \sim \mathcal{N}(\mu, \sigma^2 | x_q)$ with [62]:

$$\mu(x_q) = k(x_q, X)K^{-1}y \tag{8}$$

$$\sigma^2(x_q) = k(x_q, x_q) - k(x_q, X)K^{-1}k(X, x_q) \tag{9}$$

where $k(x_q, X)$ is the cross-correlation vector of the query point x_q to the inputs X and K is the kernel matrix:

$$\mathrm{K} = \begin{pmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{pmatrix} + \sigma_n^2 I \tag{10}$$

where σ_n^2 is a noise term representing stochastic functions.

Once the surrogate model is established, the acquisition function is to specify the next sample x_{t+1} by maximizing the expected improvement (EI), $x_{t+1} = arg \max_x EI(x)$. EI is defined by:

$$\mathrm{EI}(x) = (\rho - \mu)\Phi(z) + \sigma\phi(z) \tag{11}$$

where Φ is the corresponding cumulative density function and ϕ is the Gaussion probability density function with $z = (\rho - \mu)/\sigma$. (μ, σ^2) are the predicted value through eq. (9).

3. Network design

The modern CNN architectures are modular and built by stacking the same or similar blocks to extend the network depth. In the recent LCNNs for MFD [36-38], the contributions lie in proposing a new LCNN block by introducing one or two lightweighting measures. Instead,

we propose a systematic approach for designing LCNN under limited computational resources. The flowchart of the design process is shown in Figure 5.

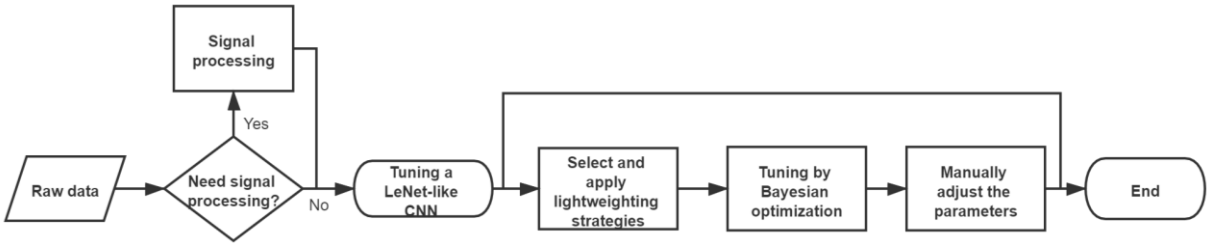


Figure 5. Design process of a LCNN architecture

3.1 BOGP for parameter tuning

Firstly, it should be considered what data is used as the input data for CNN. The input data pre-processed by signal processing techniques have a lower dimensionality. However, the computational complexity of signal processing cannot be negligible. Taking the raw acceleration signal as the input data avoids pre-processing and saves many computational resources. However, the raw data have a higher dimensionality, which should be addressed during the network design. We start with designing a basic 1D LeNet-like CNN with the raw acceleration signal as the input data. This CNN merely contains the basic elements like the regular convolution layer, max pooling and the ReLu activation function. In terms of training, we use mini-batch stochastic gradient descent with Adam optimization [63]. We apply BOGP for parameter tuning with 100 iterations. In this demonstration, the raw carbody acceleration is sampled with a frequency of 5000 Hz. The input data for CNN is a series of one-second data samples. Given the input of size 5000×1, the tunable hyperparameters and the corresponding ranges are presented in Table 2.

Table 2. Tunable hyperparameters and the corresponding ranges of the LeNet architecture

Hyperparameter	Range	Optimal value	Hyperparameter	Range	Optimal value
Learning rate	$e^{-5} - e^{-2}$	e^{-5}	Conv2 filter size	3 - 11	11
Batch size	16 - 64	17	Conv3 filter number	24 - 64	24
Epoch	20 - 60	54	Conv3 filter size	3 – 11	10
Layer number	1 - 5	4	Conv4 filter number	48 - 128	66
Conv1 filter number	6 - 16	10	Conv4 filter size	3 - 11	3
Conv1 filter size	3 - 11	4	Conv5 filter number	96 – 256	-
Conv2 filter number	12 – 32	27	Conv5 filter size	3 - 11	-

Figure 6 shows the results of hyperparameter tuning with BOGP. The results give us the insight that a 1D LeNet can detect WF by feeding the raw carbody acceleration. It also

provides a starting point for the optimization of the network architecture and parameters toward lightweight. The convergence plot indicates that the validation accuracy is improved from 84.75% to 93.53% through parameter tuning in 150 iterations. The partial dependence plots indicate the partial influence of each parameter on validation accuracy and the surrogate model for the objective function. The black points on the surrogate model refer to the sample points, while the red star refers to the sample point achieving the best validation result. This point may not be the global optimum. The partial dependencies reveal the general trend of parameter influences for further manual adjustment. In this demonstration, a smaller learning rate could achieve better accuracy. Batch size and epoch have no influence. The optimal depth of LeNet is four layers. A smaller number of the filters in the third layer and a larger size of the filters in the second layer could result in better accuracy. Based on this, the optimized architecture of LeNet has been determined, as shown in the left graph of Figure 7.

3.2 Proposed LightWFNet

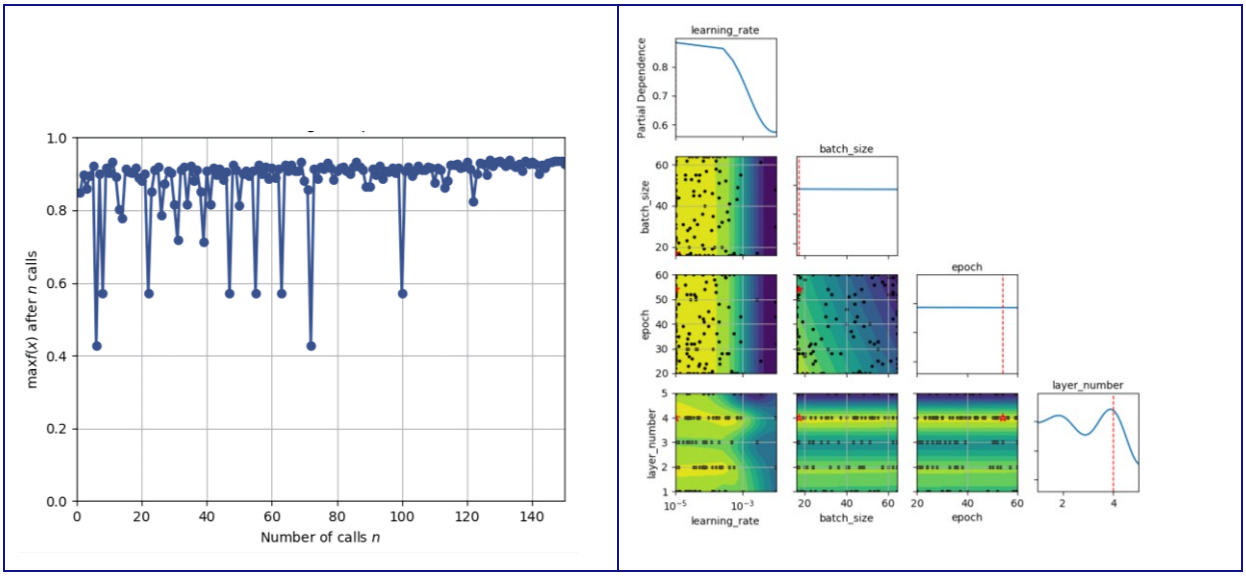
In order to achieve the best tradeoff between accuracy and model complexity, the following lightweight strategies are considered to optimize the architecture. As the raw acceleration data, depending on sample frequency and length of the time window, usually has a large dimension, the first layers should attempt to reduce the input dimensionality. Second, the small size of filters could significantly reduce the number of parameters. We should fix the filter size of 3×1 and turn the structure and other parameters. Third, the regular convolution can be replaced with DSC with little influence on network performance. Fourth, the linear bottleneck bock with the skip connection can further reduce the dimensionality and improve the performance. Based on these considerations, the CNN structure and the parameters are automatically tuned by BOGP at first and then manually adjusted to achieve a better tradeoff between accuracy and complexity. As BOGP can only be used for parameter tuning, the control parameters are defined for tuning the structure, which works as the "switches" to enable or disable the specific structures. For instance, the parameter "SE" denotes Squeeze and Excitation. In case $SE = 1$, a SE unit is inserted into the block.

The right graph of Figure 7 shows the architecture of LightWFNet evolved from the tuned LeNet. It consists of three parts for downsampling, feature extraction and classification respectively. The fat font indicates the main layer within the block. The parameters of the filter number, the filter size and the stride size for the main layer are given in the block. For instance, "S=2" for stride step of 2, while "3×1@8" denotes 8 filters of size 3×1.

Part 1 attempts to reduce the dimensionality without distorting the signal patterns. We use two subsequent DSC blocks to smooth the noise and enhance the WF signature. The DSC block is composed of a DSC layer, a batch normalization layer and an activation layer. The DSC has almost the same effect as the regular convolution block but fewer MAC operations. The batch normalization layer normalizes the inputs for the subsequent layer, avoiding covariance shift across the entire network. As the activation function, Hard-Swish is applied to balance the accuracy improvement due to nonlinearities and implementation difficulty of swish-activation in embedded devices. Subsequently, a max-pooling layer is applied for 10-times downsampling, given that the input raw data has a dimension of 5000×1. This downsampling factor could be adjusted according to the input dimension.

a. Convergence plot

b. Partial dependency of learning rate, batch size and epoch



c. Partial dependency of filter number in each layer

d. Partial dependency of kernel size in layer

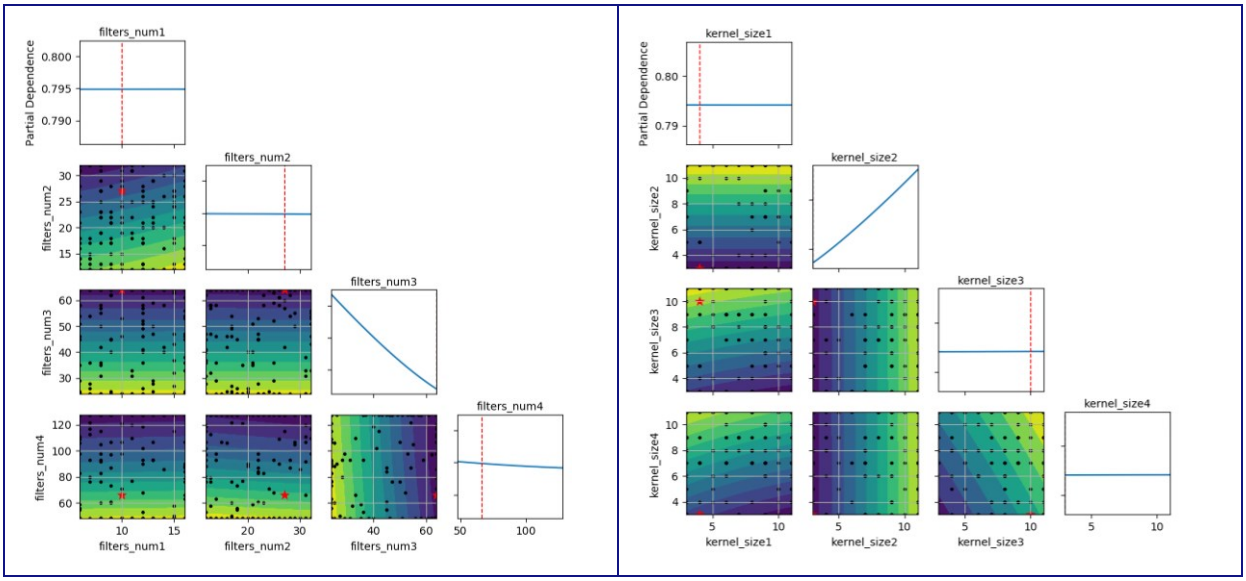


Figure 6. Results of hyperparameter tuning with BOGP: (a) convergence plot, where y-axis stands for validation accuracy and x-axis for iteration step; (b) partial dependency of learning rate, batch size, epoch and layer number; (c) partial dependency of filter numbers in each layer; (d) partial dependency of filter size in each layer. The color map presents the surrogate model of the corresponding two parameters, where brighter color means higher accuracy and darker color means lower accuracy. The curve plot indicates the partial influence of each parameter on accuracy.

The second part is to extract hierarchy features from the downsampled data by using two subsequent linear bottleneck blocks, which can be repeated to increase the depth of the network. In the case of WFD, a four-layer CNN (i.e. two bottleneck blocks) is sufficient for WFD according to the results of parameter tuning. Consider the multi-channel 1D input feature map having the dimensions $N \times 1 \times C$, where N denotes the array length and C

denotes the channels, a bottleneck block uses C/s pointwise filters to shrink the size of the intermediate output channels and subsequently use N-channels DSC layers to increase the output channels, where s is the shrinking factor. This forces the neural network to compress the information into the thin bottleneck layer in order to reduce the dimensionality, reduce the redundant information and strength the abstract signature of multiple peaks caused by WF. Especially when it incorporates the SE unit to weight the features, more informative features will have higher weights and thus are highlighted, whereas the less activated features are suppressed. In the case of WF signals, that means, the amplitude of the features that represent peaks will be further increased. Similar to introducing hard swish, the hard version of the sigmoid function is used to replace the original sigmoid function within the SE unit. In the experiments, we have observed that the information shrinkage could result in the high sensitivity of the CNN model to the peaks so that the trained model is more likely to classify a healthy data sample as a WF data sample, i.e. a false alarm. The reason is that the model treats the peaks due to noise or other impulsive interference as the WF signal. This effect could be compensated by the skip connection (i.e. residual structure). It is worth noting that the first bottleneck block has the stride size of 2 for further dimension reduction and thus does not have the skip connection. The second bottleneck block has a stride size of 1 and thus incorporates the residual connection. The two variants of the linear bottleneck blocks with the SE units are shown in Figure 8. Finally, the array length of the outputs is further reduced by 4×1 max pooling.

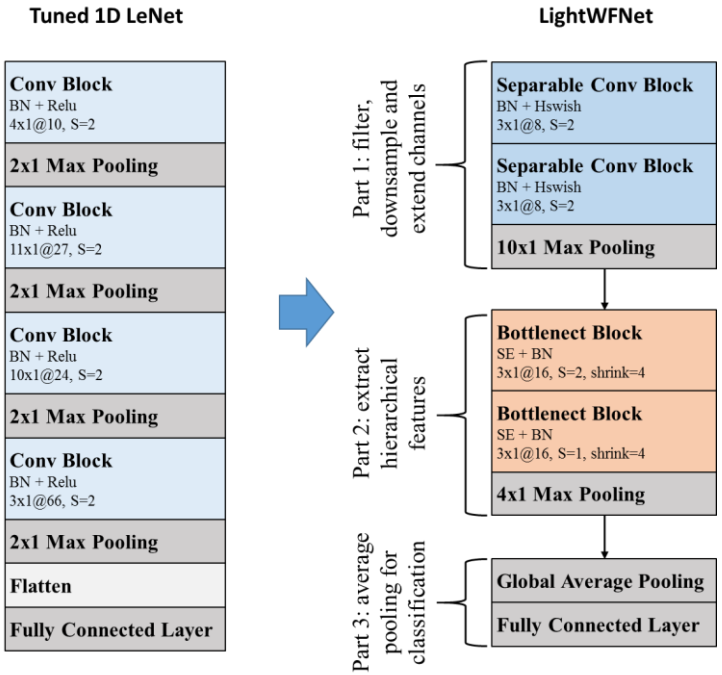


Figure 7. Architecture of the tuned LeNet and the proposed LightWFNet, where BN denotes batch normalization layer, "S=2" for stride step of 2, "3×1@8" denotes 8 filters of size 3×1, "Hswish" for hard swish activation function, "SE" for squeeze and excite, "shrink=4" denotes the shrink factor within bottleneck block. "Conv/Separable Conv block" is a sequential stack of the convolution/DSC layer, batch normalization layer and the activation function. Bottleneck block is introduced in Section 2.

So far the abstract signature of the input raw data has been extracted. In 1D CNN, each feature map is a 1D array in the corresponding channel. The next stage is to obtain one feature

value per channel as the input for the one-layer fully connected neural network (FCNN). We use the global average pooling (GAP) operation instead of the flatten operation. GAP calculates the average value of the feature arrays per channel, rather than flattening all points in the arrays as the features. The latter could result in overfitting and have more trainable parameters. This approach is firstly proposed in [64] and is commonly accepted now. Finally, the FCNN with softmax activation is used as the classifier.

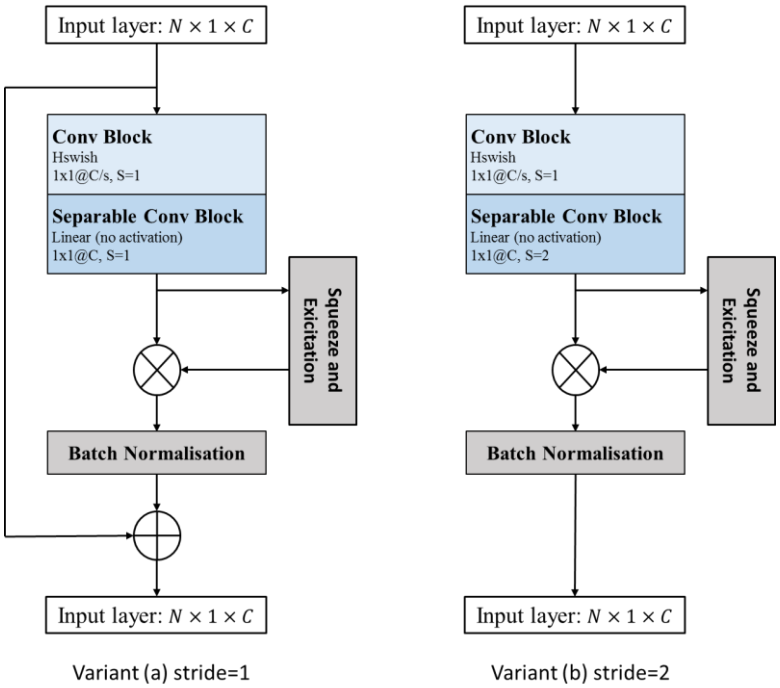


Figure 8. Architecture variants of the repeatable linear bottleneck block with the SE unit

4. Validation experiments

To validate the proposed LightWFNet, the WFD models are trained and tested on carbody accelerations obtained from a field test in real operating conditions. The CNN architectures are implemented in the TensorFlow framework. The trained model can be loaded on the embedded system for real-time WFD with the support of TensorFlow Lite. The test results are compared to the ones delivered by the tuned LeNet CNN, a classical ML-based method [65] and the SOTA LCNN for MFD [36-38] in terms of network performance and computational complexity.

4.1 Data preparation

In the field test, a tank wagon was instrumented, see Figure 9. The empty wagon was firstly operated on the mainline for about 140 km with a synthetic WF of 20 mm. After replacing the defective wheel with an intact wheel, the wagon was further operated for over 600 km in the empty and fully loaded conditions respectively. The journey covered tens of curves with a radius ranging from 200 m to over 5000 km. The vibration data was measured by Brüel & Kjaer type 4520 triaxial piezoelectric accelerometers with a sensitivity of $1.02\,mv/ms^{-2}$. Table 3 gives an overview of the usable data after data cleaning, where the implausible data due to failures of the measurement system and the data at a stillstand or very low speeds were

discarded. The WF detection model is designed for a telematics device, which takes one-second samples of discrete acceleration data. Therefore, the continuous field data is split into thousands of one-second data samples as the input data for CNN. Given the time window of $1s$, the lowest vehicle speed v that enables WFD can be derived from eq. (12). Therefore, the lowest vehicle speed for WF detection is $20.8\ km/h$. Based on this, the acceleration signals at vehicle speeds over $25\ km/h$ are used. We use 60% of the entire data for training, 20% for validation and 20% for testing, regardless of running conditions.

$$\begin{gathered} 2 * \pi d/v \leq t \\ v \geq 2 * \pi d/t = 2 * \pi * 0.92 = 20.8\, km/h, \end{gathered} \tag{12}$$

where d is the diameter of wheels; t is the length of the time window. The diameter of a new wheel for railway freight wagons is normally $0.92m$.

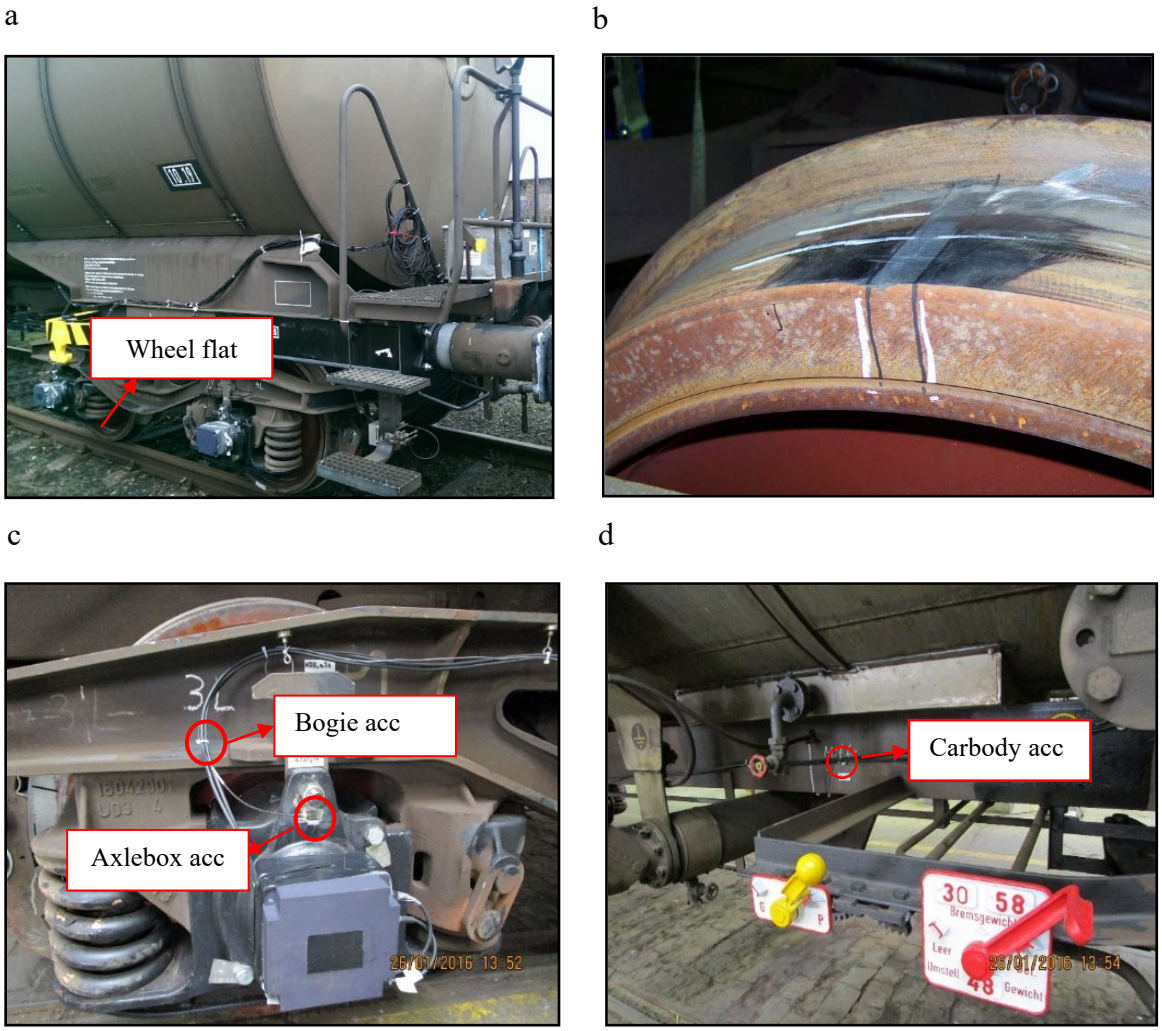


Figure 9. Field test: (a) instrumented tank wagon and position of wheel flat; (b) 20 mm synthetic wheel flat; (c) position of the accelerometers at axlebox and bogie; (d) position of the accelerometer at carbody

The running speed of a freight wagon ranges from 0 to 120 km/h in practice. The WFD model should be robust against speed variation. The variation of vehicle speeds affects the amplitude of vibration signals [17], the faulty frequency [12] and thus the number of the impulses

caused by WF in a fixed time window. In order to compensate for the effect that vibration amplitude increases with the vehicle speed, the raw acceleration data is standardized at each sample step using eq. (13).

$$x_{st}^{t} = (x^{t} - \bar{x})/\sigma \tag{13}$$

where x^t is data point at time t, $\bar{x}$ is the mean value over the sample time and σ is the standard deviation over the sample time.

Furthermore, the dataset is divided into different speed ranges to investigate the influence of vehicle speeds on detection accuracy. Theoretically, both very low and very high speeds have a negative effect on the WF pattern. At the speed of 20.8 km/h, only two WF peaks appear within the one-second data sample, forming a periodic pattern. This periodic pattern could be distorted by any interference peaks caused by e.g. rail squats, bad rail welds and turnouts. At very high speeds, the amplitude of WF peaks could decrease. This effect was reported in [66]. The WF amplitude first increases with the increase of vehicle speeds reaches the maximum at a so-called critical impact speed and then drops down with the further increase of speeds. The critical impact speed relies on the condition of the WF (i.e. width/length ratio and length). In the experiment, the acceleration data at speeds above 25 km/h is used for training and cross-validation. The testing dataset is divided into different speed ranges. Table 4 presents the distribution of datasets. One data sample refers to 5000 data points, i.e. one-second acceleration with the sampling frequency of 5000 Hz.

Table 3. Overview of the usable field data after data cleaning

Wheel flat conditions	Speed range	Faulty data volume	Normal data volume	Sample frequency
synthetic, 20 mm	25 – 105 km/h	10210 s	13532 s	5000 Hz

Table 4. Dataset dimension for training, validation and testing

Dataset	Train	Valid.	Test in total	Test: 25-45 km/h	Test: 45-65 km/h	Test 65-85 km/h	Test: 85-105 km/h
Num. of samples	14245	4749	4748	1142	571	1509	1526
Ratio faulty to normal	6077: 8168	2034: 2715	2099: 2649	534: 608	240: 331	1034: 475	291: 1235

4.2 Evaluation metrics

The number of positive cases and negative cases within the dataset could be imbalanced. Especially in the case of fault detection, the amount of faulty data is usually much smaller than that of normal data. Therefore, we use balanced accuracy (bACC) to indicate the performance of a WF detection model, which compensates for the negative effect of imbalanced datasets. Recall and precision are also presented to indicate the capability of true

detection and to avoid false alarms. In the case of WFD, we define "wheel flat" as positive and "intact wheel" as negative. The number of real positive cases in the dataset is denoted by P, while the number of real negative cases in the dataset is denoted by N. The number of positive cases classified correctly is denoted as TP (true positive). The number of negative cases classified correctly is denoted as TN (true negative). The number of false classification of positive cases is FP (false positive, i.e. false alarm). The number of false classification of negative cases is FN (false negative, i.e. miss detection). Then, precision is calculated by TP/P, while recall is calculated by $TP/(TP + FN)$. A higher precision indicates fewer false alarms, while a higher recall means a higher capability for WFD. In practice, high precision is more important than high recall, since false alarms could cause additional issues such as wasting manpower for inspection. Furthermore, bACC is calculated by normalizing TP and TN over P and N respectively:

$$\text{bACC} = (TP/P + TN/N)/2 \tag{14}$$

For comparison of computational complexity among different CNN architectures, the total parameters within the network and Floating Point Operations (FLOPs) are commonly used as evaluation metrics. A FLOP is viewed as a basic unit of computation, denoting an addition, subtraction, multiplication or division of two floating-point numbers.

4.3 Experiment results

Detection accuracy The test results of LightWFNet are compared to those delivered by the tuned LeNet, a classical ML-based method and the SOTA LCNN for MFD in terms of accuracy and computational complexity. The SOTA LCNNs are implemented and adapted for our datasets, incl. deep separable convolutional network (DSCN) proposed in [37], lightweight deep residual convolutional neural network (LDR-CNN) proposed in [38] and adapted ShuffleNet V2 used in [36]. DSCN takes the raw data as the input data for CNN. LDR-CNN uses WPT to transfer the raw data into multi-channel 1D wavelet coefficients as the input data. Adapted ShuffleNet V2 uses STFT to transfer the raw data into 2D matrix. The same procedures for signal processing are implemented to deliver the correct input data for LDR-CNN and ShuffleNet V2. As the input data in the original work has different dimensions, the hyperparameters are tuned to identify the best parameter combination for our datasets. Apart from the DL-based method, the comparison also includes a classic diagnostic method using time-frequency-domain features and gradient boosting decision tree (GBDT) [65]. The WF signal within a short time window can be viewed as a quasi-stationary signal. Despite the vehicle speed continuously changing over time, the acceleration or deceleration of a freight wagon normally is around $1\,m/s^2$. Its influence on the rotation frequency can be thus neglected. We use Hilbert-transform (HT) and CWT to highlight the rotation and instantaneous frequencies of the WF signal in the frequency and time-frequency domain respectively, as shown in Figure 3. Several statistic values like skewness and kurtosis in the envelope spectrum and scale-averaged wavelet power of wavelet coefficients [67] are calculated as the features.

Table 5 and Figure 10 shows the testing results in terms of bACC, recall and precision. Each method was trained and tested ten times. The mean values of results and the standard deviations are presented. In general, our LightWFNet overperforms the SOTA LCNN

architectures and GBDT in terms of bACC. The performance of LightWFNet is stable within the ten-times training procedure (having very low standard deviations) in comparison to other LCNN architectures. LightWFNet has a much higher precision but a relatively lower recall. Therefore, it is not sensitive to impulsive interference and thus generates fewer false alarms. The traditional diagnosis method GBDT with the handcrafted features in the frequency domain and the time-frequency domain has achieved a poor result. This approach significantly relies on the quality of the defined features. Although the periodic impulsive patterns of WF are visible through the frequency or time-frequency analysis, it is very challenging to manually define the low-dimension features to represent the high-dimension spectral coefficients, particularly taking into account the diverse interference under the harsh application environment. The tuned LeNet without any specific design has achieved a good bACC of 91.30%, even better than ShuffleNetV2 and DSCN. It proves that a simple multilayer CNN can effectively extract hierarchical features from the 1D acceleration signal for diagnosis. It is not necessary to transform 1D sensor data to 2D matrix, as ShuffleNetV2 does, which must result in a large increase of the computation complexity, but may not improve the diagnosis performance. In terms of DSCN, it is very unstable during the training process. Six trained models out of ten have a good bACC of over 92% with relatively high precision and low recall. However, the left four models have very high recall of over 92% and low precision of around 65%. It means that the models are very sensitive to impulsive patterns and treat most discrete impacts as WFs, although they could be caused by e.g. rail squats, bad rail welds and turnouts. LDR-CNN has achieved comparable performance with higher recall and a lower precision in comparison to LightWFNet. A lower precision means a higher possibility of false alarms, which is more critical than miss detection in the context of WF detection and should be avoided.

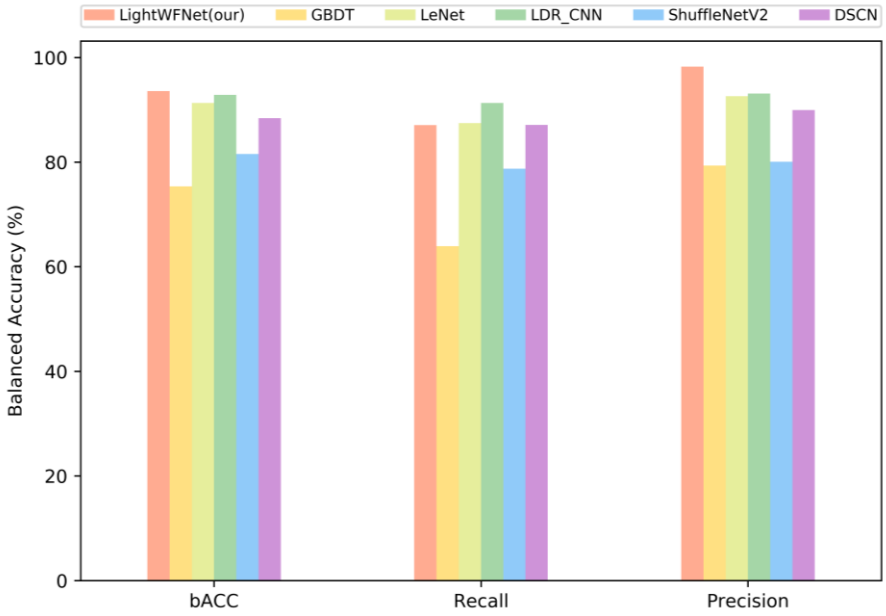


Figure 10. Comparison of bACC, recall and precision between different LCNN architectures

Computational complexity The computation complexity of a deep neural network is commonly measured by the number of its total learnable parameters and FLOPs. Table 6 compares these two metrics of different LCNNs. Our LightWFNet has much less computational complexity than other LCNNs. The lightweight measures reduce 21 times

parameter number and 378 times FLOPs from LeNet to LightWFNet. DSCN has a comparable number of parameters, but much higher FLOPs in comparison to LightWFNet. It is mainly caused by the large width of the input sensor data. In the LightWFNet, Part 1 is specifically designed to reduce the input dimension in terms of array length and remain the pattern information by increasing the channel numbers, so that the subsequent blocks for feature extraction have the low-dimensional inputs. This has not been considered in other LCNNs for MFD.

Table 5. Test results of different methods for wheel flat detection by using carbody accelerations

Result	LightWFNet	GBDT	LeNet	LDR-CNN	ShuffleNetV2	DSCN
bACC	93.58±0.32%	75.36%	91.30±0.95%	92.86±0.41%	81.54±0.67%	88.39±6.23%
Recall	87.06±0.56%	63.94%	87.44±1.37%	91.29±3.28%	78.74±3.43%	87.11±4.43%
Prec.	98.24±0.82%	79.33%	92.60±0.73%	93.10±3.70%	80.08±2.27%	89.95±13.98%

The computation complexity of a diagnosis model should also consider the costs of signal processing. The signal processing method could have comparable costs as an ML-based classification method. To have a better sense of FLOPs, FLOPs of FFT for a 5000-points input array are calculated using eq. (15) [43]. Surprisingly, LightWFNet merely has one-third of FLOPs of FFT. ShuffleNetV2 requires STFT as pre-processing, whose FLOPs can be calculated by eq. (16). LDR-CNN requires WPT as preprocessing, whose FLOPs can be calculated by eq. (17). This fact indicates that taking the raw data as the input can save a lot of computational power for the whole computing process. To further demonstrate the efficiency of LightWFNet, Figure 11 illustrates the efficiency (i.e. bACC versus FLOPs) of each diagnosis model, including signal processing.

$$FLOPs_{FFT} = 2.5N \log_2 N = 46{,}237 \tag{15}$$

$$FLOPs_{STFT} = M \cdot 2.5N \log_2 N = 1{,}895{,}722 \tag{16}$$

where N is the number of data points and M denotes the time resolution of STFT, depending on the segment length and overlap for STFT. In our case $N = 5000$ and $M = 41$ with the segment length of 256 and overlap of 50%.

$$FLOPs_{WPT} = 2^L \cdot \mathrm{N} = 80{,}000 \tag{17}$$

where L denotes the level of wavelet decomposition. In our case $L = 4$

Table 6. Computational complexity of different LCNN architectures

Result	LightWFNet	LeNet	LDR-CNN	ShuffleNetV2	DSCN
Parameters	825	15,143	6,662	808,794	1,032
FLOPs	13,053	4,934,270	1,016,480	120,215,430	209,681

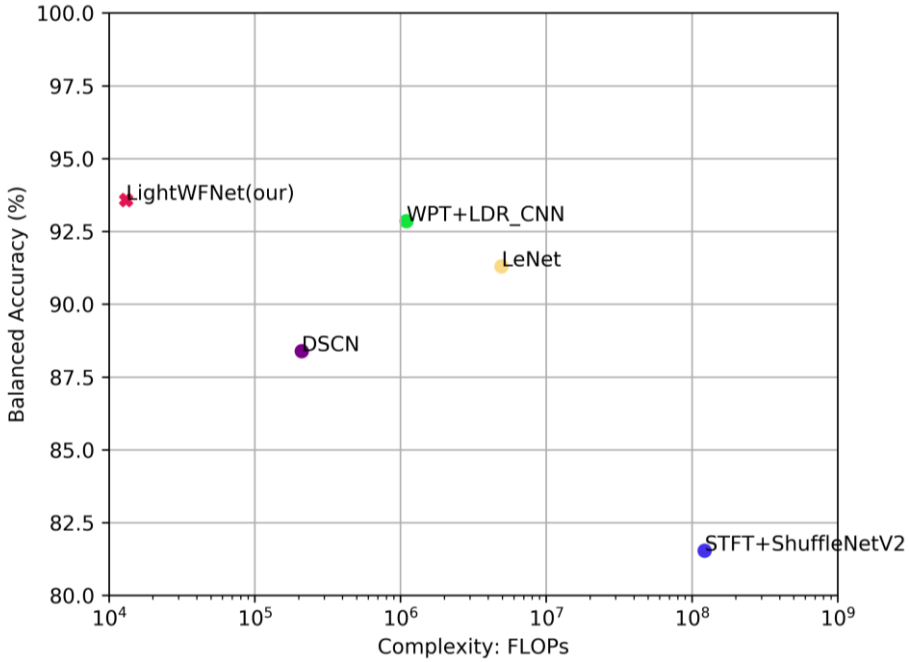


Figure 11. Comparison of the efficiency of between different WF detection models

CNN mechanism The basic concept of fault diagnosis is to highlight faulty patterns by filtering out interference or decomposition, where the signal is convolved with the filter function. In Fourier transformation, the filter function is the cosines function. In wavelet transformation, the filter function could be different wavelet functions. In CNN, the filter function is the stack of convolutional filters, whose parameters are automatically learned during the training process. Therefore, CNN works as a combination of adaptive filters. Jia et al. [68] and Zhang et al. [69] attempted to visualize the filter parameters to understand the effect of adaptive filtering. We visualize the interim layer outputs during the forward pass within the trained LightWFNet to promote an intuitive understanding. Figure 12 shows the transformation results of the raw data through different filtering/decomposition methods. The raw data contains four periodic WF peaks which are visible, despite strong interference. The left side of Figure 12 shows the results of STFT, CWT, HT and WPT respectively. Although the WF patterns could be more or less recognized, the pattern information is embedded in the high dimensional feature map. It is still difficult to define low-dimensional features to further represent the pattern information, which can be used for classification. The right side of Figure 12 shows the evolution of the feature maps within LightWFNet. The noise is eliminated and the visibility of the WF peaks is strengthened layer by layer. The final feature map in each channel has merely 16 data points but can clearly show the abstract patterns of WF. From each feature map, the average value is calculated as a feature for classification. In this way, the representative features of the input raw data are extracted layer by layer through diverse filters, hierarchical structure and pooling operations within CNN. The depth of the layers ensures that the noise information can be filtered out and the abstract patterns of the input signals can be extracted. Therefore, the CNN model could be more invariant to signal variations, which could be caused by variations in measurement conditions such as vehicle speeds, track conditions, etc.

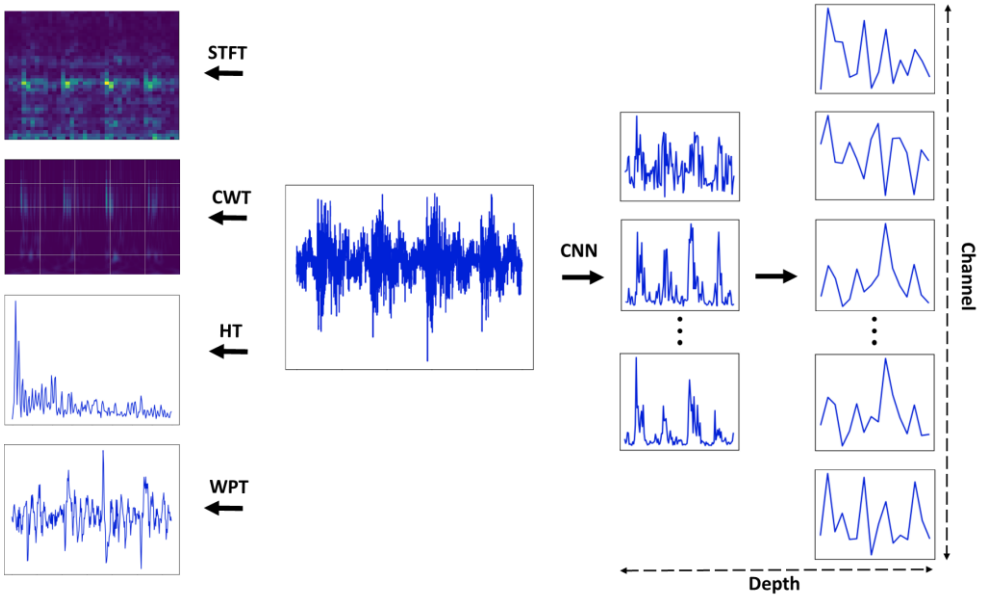


Figure 12. Transformation results of the raw data (in the middle) through different filtering/decomposition methods. The left side presents the results of STFT, CWT, HT and WPT. The right side presents the evolution of the feature maps within LightWFNet.

Table 7. Test results of different methods for wheel flat detection in different speed ranges

Speed ranges	LightWFNet	GBDT	LeNet	LDR-CNN	ShuffleNetV2	DSCN
25-45 km/h	96.02%	81.85%	89.54%	95.94%	85.05%	95.87%
45-65 km/h	95.17%	73.74%	90.61%	95.17%	81.52%	94.89%
65-85 km/h	99.08%	77.61%	84.05%	97.50%	85.93%	97.99%
85-105 km/h	67.86%	58.55%	71.24%	80.51%	63.48%	68.40%

Influence of vehicle speed variation The one-second carbody acceleration as the input data for LightWFNet is significantly influenced by the vehicle speed. Figure 13 compares the WF signals at the carbody level and their CNN decompositions at different speeds. At low and medium speeds, the increase of vehicle speeds aggravates the WF pattern at the carbody level mainly through the increase of rotation frequency and thus the number of WF peaks within the fixed time window. CNN could extract the information of WF peaks. At high speeds, the WF pattern changes from the subsequent peaks to a zig-zag pattern which is hardly recognizable. A high speed could result in the fact that the time of one wheel rotation is less than the time of the oscillation of one WF impact. That means two adjacent WF peaks may overlap, which changes the WF pattern. CNN fails to follow this new pattern. To investigate the influence of the vehicle speed on the accuracy of the trained WF detection models, we

divided the test dataset into different speed ranges. Table 7 and Figure 14 shows bACC delivered by different models for different speed ranges. All the models show a common fact that detection accuracy in the highest speed range (85-105 km/h) drops dramatically. This confirms our observation in Figure 13 and may suggest that wheel flat detection on carbody should not be performed at high speeds.

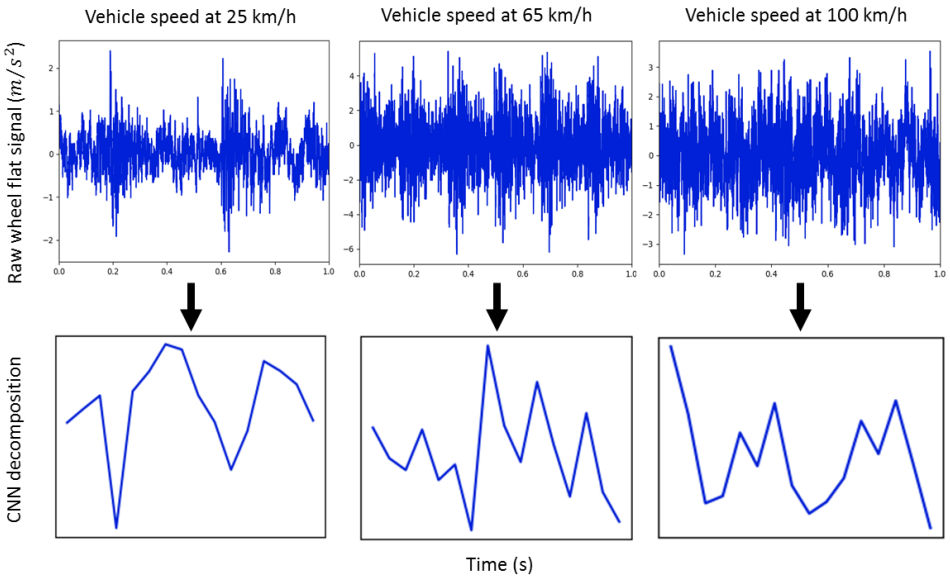


Figure 13. Comparison of WF signals and their CNN decomposition at different speeds

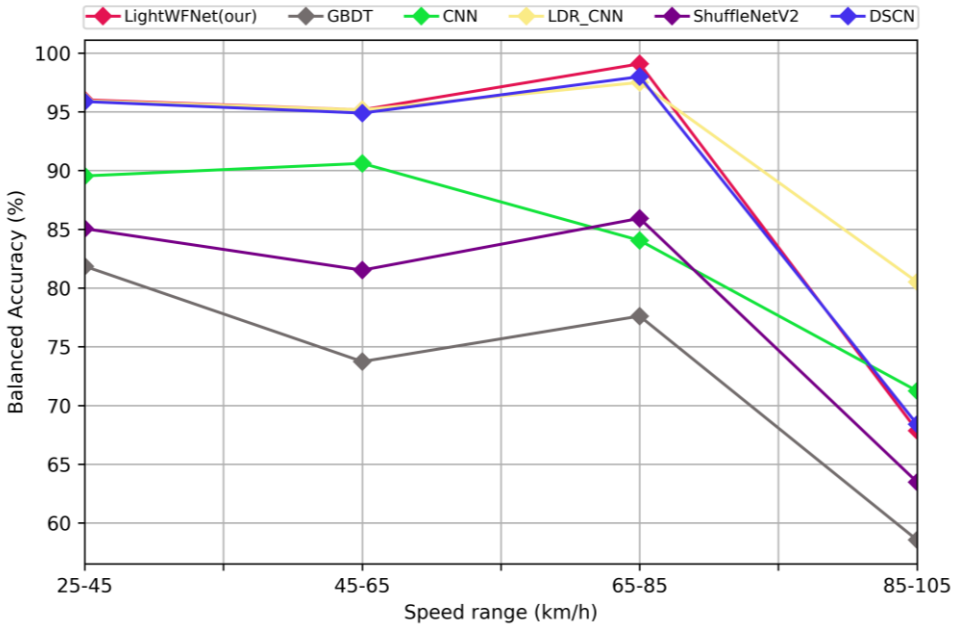


Figure 14. Balanced accuracy of different models in different speed ranges

5. Conclusions and future work

In this paper, a systematic design process of LCNN is proposed and demonstrated for WFD. The generated 1D LCNN architecture is named as LightWFNet. The CNN design for a

specific task could start with a basic CNN structure. Thanks to Bayesian optimization, the CNN parameters can be automatically tuned. Afterward, lightweight is guided by the existing strategies and executed by BOGP to tune the parameter and the structure. Afterward, the tuned structures and parameters can be manually adjusted to achieve a better tradeoff between accuracy and complexity. In this way, LightWFNet evolves from a simple LeNet-like CNN. LightWFNet covers the entire chain of automatic diagnosis, being fed by raw accelerations, downsampling, extracting features and performing classification. On the one hand, LightWFNet incorporates the strategies of depthwise separable convolution, small filter size, downsampling at an early stage, hard version of activation functions and bottleneck structure for lightweight. On the other hand, LightWFNet introduces squeeze and excitation and residual structure to enhance network performance. In the experiments, LightWFNet overperforms the tuned LeNet, the state-of-the-art LCNN for MFD and a classic ML-based method in terms of accuracy and complexity. In particular, the results show the computational cost of LightWFNet is much lower than the common methods of signal processing. This allows LightWFNet to be deployed on the existing telematics devices for real-time diagnosis. Furthermore, the decomposition and filtering effect of CNN is visualized by displaying the interim layer outputs. It reveals the mechanism of CNN for feature extraction. The experiment of speed variations indicates that WF detection by using carbody accelerations is very challenging at high speeds (above 85 km/h). At high speeds, the WF pattern changes from the periodic peaks to a zig-zag pattern. All methods fail to deliver a satisfying result. In practice, this can be easily solved by an engineering approach. For instance, only the detection results at speeds between 25-85 km/h are used for maintenance decision-making. Alternatively, we can use the classification probability delivered by LightWFNet, rather than the label. The probabilities for one-day monitoring are aggregated and displayed on the dashboard for maintenance decision-making. The outlier points due to false diagnoses can be easily recognized and eliminated.

In practice, other factors such as wagon conditions and track conditions could also have a great impact on the acceleration patterns. Therefore, the robustness of a WF detection model under different conditions should be investigated. We will continue to refine the present work and improve the robustness and generalization of the DL-based diagnostic model.

Acknowledgment

The experiment data used in this paper is supported by the previous projects of Chair of Rail Vehicles TUB. The research is funded by the EU Shift2Rail project Assets4Rail (Grand number: 826250) under Horizon 2020 Framework Programme.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

[1] Leiste M, Hecht M, et al. Roadmap zur Digitalisierung der Wagentechnischen Untersuchung. Chair of Rail Vehicles; Technische Universität Berlin. 2018; Intern report 16/2018.

[2] GCU, General contract of use for wagons – GCU. Edition dated 1 January 2018.

[3] Bernal E, Spiryagin M, Cole C. Onboard condition monitoring sensors, systems and techniques for freight railway vehicles: a review.IEEE Sensors Journal.2019;19:4-24.

[4] Schaeffler. Axlebox generator for railway applications. 28.01.2020. Available:

[5] https://www.schaeffler.com/remotemedien/media/_shared_media/08_media_library/01_publications/schaeffler_2/brochure/downloads_1/org_de_en.pdf

[6] Wheelwright HE, Vincent D. Track defect and wheel damage: detection and location. 2020. Perpetuum's white paper. 2020. Available:

[7] https://perpetuum.com/download/track-defect-and-wheel-damage-detection-and-location/?wpdmdl=1480&refresh=5e302ed474a071580216020

[8] DB Cargo. Mit intelligenten und leisen Güterwagen für Wachstum im Schienengüterverkehr. Deutsche Bahn. Feb. 2018. Available: https://www.deutschebahn.com/de/presse/pressestart_zentrales_uebersicht/DB-Cargo--Mit-intelligenten-und-leisen-G%C3%BCterwagen-f%C3%BCr-Wachstum-im-Schieneng%C3%BCterverkehr-1440210.

[9] Harms C. Im minutentakt: Wie VTG die Güter aus dem Schwarzen Loch holt. Allianz pro Schiene. 03.2017. Available: https://www.allianz-pro-schiene.de/themen/aktuell/interview-hanno-schell-digitalisierung-gueterwagen-vtg/

[10] Shi D, Marin-Perianu R, et al. Wireless data communication concept for cargo condition monitoring system. 2018. Technical report. Deliverable D2.3. EU Shift2Rail project; Grand Nr:730863-S2-OC-IP5-03-2015

[11] Alemi A, Corman F, Lodewijks G. Condition monitoring approaches for the detection of railway wheel defects. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit. 2016;231:961–981

[12] Jiang H, Lin J. Fault diagnosis of wheel flat using empirical mode decomposition-Hilbert envelope spectrum. Mathematical Problems in Engineering. 2018;2018:1–16.

[13] Zhao R, Shi H. Research on wheel-flat recognition algorithm for high-speed train based on high-order spectrum feature extraction. Journal of Mechanical Engineering 2017;53:102

[14] Li Y, Zuo M. J, Lin J. Fault detection method for railway wheel flat using an adaptive multiscale morphological filter. Mechanical Systems and Signal Processing. 2017;84:642–658.

[15] Liang B, Iwnicki S. D, Zhao Y, Crosbee D. Railway wheel-flat and rail surface defect modelling and analysis by time–frequency techniques. Vehicle System Dynamics. 2013;51:1403–1421.

[16] Gericke C. Methoden zur on-board-Diagnose von Radlaufflächenschäden: ein Beitrag zur zustandsorientierten Instandhaltung von Schienenfahrzeugen. Dissertation; Technische Universität Berlin. 2013

[17] Bosso N, Gugliotta A, Zampieri N. Wheel flat detection algorithm for onboard diagnostic. Measurement. 2018;123:193–202.

[18] Bernal E, Spiryagin M, Cole C. Wheel flat detectability for Y25 railway freight wagon using vehicle component acceleration signals. Vehicle System Dynamics. 2019;:1-21

[19] Shi D, Bruni S, et al. Models for reliability, statistical information, real time health status of the rolling stock, prognostic analysis, and economic data. 2018. Technical report. Deliverable D4.2. EU Shift2Rail project. Grand no: 730863-S2R-OC-IP5-03-2015.

[20] Jangamreddy N. A survey on specialised hardware for machine learning. 2019. DOI: 10.13140/RG.2.2.20697.26725

[21] Cheng Y, Wang D, Zhou P, Peng T. A survey of model compression and acceleration for deep neural networks. IEEE signal processing magazine. 2019. Special issue on deep learning for image understanding (arxiv extended version).

[22] Forrest N, Song H, et al. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. 2016. arXiv:1602.07360v4

[23] Chollet F. Xception: deep learning with depthwise separable convolutions. 2016. arXiv:1610.02357v3

[24] Zhang X, Zhou X, Lin M, Sun J. Shufflenet: an extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2018.

[25] G. Howard, et al. Mobilenets: efficient convolutional neural networks for mobile vision applications. 2017. arXiv:1704.04861.

[26] Ma NN, Zhang XY, Zheng HT, Sun J. Shufflenet v2: practical guidelines for efficient CNN architecture design. 2018. arXiv:1807.11164v1

[27] Sandler M, Howard A, et al. Mobilenetv2: inverted residuals and linear bottlenecks. 2018. arXiv:1801.04381

[28] Howard A, Sandler M, et al. Searching for MobileNetV3. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). 2019.

[29] Passalis N, Tefas A. Training lightweight deep convolutional neural networks using bag-of-features pooling. IEEE Transactions on Neural Networks and Learning Systems. 2019;30:1705-1715.

[30] Lu Z, Qin S, et al. One-shot learning hand gesture recognition based on lightweight 3D convolutional neural networks for portable applications on mobile systems. IEEE Access. 2019;7:131732-131748.

[31] Yu R, Xu X, Shen Y. Rhnet: lightweight dilated convolutional networks for dense objects counting. 2019 Chinese Control Conference (CCC). 2019;:8455-8459.

[32] Cui Y, Shi Y, Sun X, Yin W. S-net: a lightweight convolutional neural network for n-dimensional signals. 2018 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). 2018;:1-4.

[33] Li D, Shangyou Z, et al. Design of high performance convolutional neural network for lightweight platform. 2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC). 2019;:1-8.

[34] Xiang H, Chen L, Xu W. DrfNet: a lightweight and high accuracy network for resource-limited implementation. 2017 IEEE 12th International Conference on ASIC (ASICON). 2017;:1086-1089.

[35] Zhang X, Zou Y, Wang W. Ld-CNN: a lightweight dilated convolutional neural network for environmental sound classification. 2018 24th International Conference on Pattern Recognition (ICPR). 2018;:373-378.

[36] Liu H, Yao D, Yang J, Li X. Lightweight convolutional neural network and its application in rolling bearing fault diagnosis under variable working conditions. Sensors (Basel). 2019;6;19(22):4827.

[37] Wang B, Lei Y, Li N and Yan T. Deep separable convolutional network for remaining useful life prediction of machinery. Mechanical Systems and Signal Processing. 2019;134:106330.

[38] Ma S, Liu W, et al. Lightweight deep residual CNN for fault diagnosis of rotating machinery based on depthwise separable convolutions. IEEE Access. 2019;7:57023-57036.

[39] Ma S, Cai W, Liu W, Shang Z, Liu G. A lighted deep convolutional neural network based fault diagnosis of rotating machinery. Sensors (Basel). 2019(10):2381.

[40] Iwnicki S. D, Stichel S, Orlova A, Hecht M. Dynamics of railway freight vehicles. Vehicle System Dynamics. 2015;53:995–1033.

[41] Velasco-Montero D, Fernández-Berni J, et al. Performance analysis of real-time DNN inference on Raspberry Pi. Real-Time Image and Video Processing 2018. 2018.

[42] Shi D, Ulianov C, et al. Energy concept for cargo condition monitoring. 2019. Technical report; Deliverable D2.2; EU Shift2Rail project; Grand no: 730863 - S2R-OC-IP5-03-2015

[43] Frigo M, Johnson S. G. FFT benchmark methodology. 2020. Available: http://www.fftw.org/speed/method.html

[44] Munoz A, Ertle R, M Unser. Continuous wavelet transform with arbitrary scales and O(n) complexity. Signal Processing. 2002;82:749 – 757

[45] Sundararajan D. Fundamentals of the discrete Haar wavelet transform. 2011. Available:

[46] https://www.dsprelated.com/Documents/d_sundararajan_lpaper.pdf

[47] Wang Y.H, Yeh C.H, Young H.W.V, et al. On the computational complexity of the empirical mode decomposition algorithm. Physica A: Statistical Mechanics and its Applications. 2014;400:159-167.

[48] Louppe G. Understanding random forests: from theory to practice. PhD dissertation. 2015. arXiv:1407.7502v3

[49] Claesen M, Smet F.D, Suykens J.A.K, D Moor B. Fast prediction with SVM models containing RBF kernels. 2014. arXiv:1403.0736

[50] Zhang F, Webb G.I. A comparative study of semi-naive Bayes methods in classification learning. Proceedings of the Fourth Australasian Data Mining Workshop. 2005;:141-156.

[51] Rasmussen C, Williams C. Gaussian processes for machine learning. MIT Press. 2006. ISBN 026218253X.

[52] LeCun Y, et al. Handwritten digit recognition with a back-propagation network. Proc. Adv. Neural Inf. Process. Syst. 1990;:396–404.

[53] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. 2015. arXiv:1502.03167

[54] Sifre L, Mallat S. Rigid-motion scattering for texture classification. Comput. Sci. 2014. arxiv:1403.1687v1.

[55] Hu J, Shen L, Albanie S, et al. Squeeze-and-excitation networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018;:7132-7141.

[56] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 2016;:770–778.

[57] Veit A, Wilber M, Belongie S. Residual networks behave like ensembles of relatively shallow networks. 2016. arXiv:1605.06431

[58] Maas, AL, Hannun AY, Andrew Y. Rectifier nonlinearities improve neural network acoustic models. Proc. icml. 2013;30.

[59] Djork-Arné C, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (ELUS). 2016. arXiv:1511.07289.

[60] Ramach P, Zoph B, Le Q. Searching for activation functions. 2017. arXiv:1710.05941.

[61] Misra D. Mish: a self-regularized non-monotonic neural activation function. 2019. arXiv:1908.08681

[62] Martinez-Cantin R. Funneled Bayesian optimization for design, tuning and control of autonomous systems. IEEE Transactions on Cybernetics. 2019;49:1489-1500

[63] Kingma D, Ba J. Adam: a method for stochastic optimisation. International Conference on Learning Representations. 2015;:1–13

[64] Lin M, Chen Q and Yan S. Network in network. ICLR 2014. 2014. arXiv:1312.4400.

[65] Kou L, Qin Y, Zhao X, Fu Y. Integrating synthetic minority oversampling and gradient boosting decision tree for bogie fault diagnosis in rail vehicles. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit. 2018;233(3):312–325.

[66] Ren Z. An investigation on wheel/rail impact dynamics with a three-dimensional flat model. Vehicle System Dynamics. 2018;57(3):369-388.

[67] Wang X, Makis V, Yang M. A wavelet approach to fault diagnosis of a gearbox under varying load conditions. Journal of Sound and Vibration. 2010;329:1570–1585.

[68] Jia F, Lei Y, Lu N, Xing S. Deep normalised convolutional neural network for imbalanced fault classification of machinery and its understanding via visualisation. Mechanical Systems and Signal Processing. 2018;110:349–367.

[69] Zhang W, Li C, Peng G, Chen Y, Zhang Z. A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working loads. Mechanical Systems and Signal Processing. 2018;100:439-453.

5. Robustness Enhancement of Diagnostic Models

This publication presents the empirical study of model robustness against variable operating conditions encountered in normal operation. The study showed that the performance of the diagnosis models for WFD is significantly impaired, when vehicles run at different speed ranges or the diagnosis models are executed on different vehicles. To enhance model robustness, a novel data augmentation framework was proposed. It incorporates MBS to simulate arbitrary operating conditions and fast weighted feature-space averaging (FWFSA) to augment the simulated faulty data. The proposed MBS-FWFSA framework has been validated on four datasets collected on different freight wagons running under different conditions. Its superiority was verified by comparison with several SOTA data augmentation methods for machine fault diagnosis.

The accepted manuscript below is the article published by Elsevier in Mechanical Systems and Signal Processing on 23rd July 2021, available online:

https://doi.org/10.1016/j.ymssp.2021.108217

Robustness Enhancement of Machine Fault Diagnostic Models for Railway Applications through Data Augmentation

Dachuan Shi *, Yunguang Ye, Marco Gillwald, Markus Hecht

Institute of Land and Sea Transport Systems, Technical University of Berlin, Berlin 10587, Germany

* Corresponding E-mail: dachuan.shi@tu-berlin.de, Tel.: +49 030 314 79806 and Fax: +49 030 314 22529

Abstract

The performance of machine learning based machine fault diagnosis (MFD) models could be impaired due to operating condition variations encountered in the real-world industrial environment, such as variations of operating speeds and loads. One major reason for this robustness problem is a lack of adequate training data, especially faulty data, measured in various operating conditions. To cover this gap, we propose a novel data augmentation framework for robustness enhancement in railway MFD applications. First, multibody dynamic simulation (MBS) for physical modeling is applied to simulate arbitrary faulty and operating conditions. Second, fast weighted feature-space averaging (FWFSA) as a new data augmentation technique is developed to augment the simulated faulty data, producing infinite reality-augmented simulation data. The proposed MBS-FWFSA can fit in arbitrary MFD algorithms and transfer-learning settings with minimal effort. Moreover, an in-depth empirical study has been carried out to investigate the causality between condition variations and robustness. A new metric has been defined to evaluate robustness. The experiments also revealed the effect of the proposed MBS-FWFSA and its outperformance against several state-of-the-art augmentation methods. The code and data used in this paper have been shared in our GitHub repository: https://github.com/quickhdsdc/Robustness-Enhancement-of-Machine-Fault-Diagnostic-Models.

Keywords: Data augmentation, Machine fault diagnosis, Machine learning, Robustness, Covariate shift

1. Introduction

1.1 Background and motivation

Machine learning (ML) and deep learning (DL) have been widely applied for machine fault diagnosis (MFD) in railway and industrial applications [1,2]. The conventional ML setting assumes that the test dataset follows the same data distribution as the training dataset. However, real-world industrial applications may not follow this assumption. Variations of operating conditions may result in covariate shift, which refers to the distribution shift of the independent variables (as the inputs for ML models) [3]. When an ML-based MFD model has been trained in the source domain under specific operating conditions, and the trained model is deployed for diagnosis in the target domain under condition variations, it may suffer from performance impairment due to covariate shift. This problem is regarded as a robustness problem. Robustness is the property that the model performance is close when tested within and outside the source-domain data distribution. In the area of MFD, several previous studies have paid attention to the robustness against Gaussian noise [4-6], variations of loads and speeds [7,8], and loss of sensor data [9].

Various measures have been proposed to enhance the robustness of ML and DL models. They broadly fall into three categories, namely network design, domain adaptation/generalization, and data augmentation. The specific network design aims to enhance the representation and generalization ability of deep neural network (DNN) and improve its robustness. Li et al. [10] proposed adaptive batch normalization by modulating the statistics from the source to the target domain. This can substitute the standard normalization layers in an arbitrary neural network (NN). Multiscale architectures can enhance robustness by operating across scale space at each convolutional NN layer, such as Multigrid Networks [11] and Multiscale Dense Networks [12].

Domain adaptation/generalization is a sub-discipline of transfer learning. It aims to reduce the disparity between the source and target domain. Domain adaptation assumes that the unlabeled data in the target domain is available during the training process to align the covariate shift. The alignment process either occurs in the pre-processing or is directly included during the NN training. As a pre-processing measure, importance weighting is to reweight the source domain data points according to the estimated probabilities that they fall in the target domain data distribution. If the probability is high, a high weight is assigned to the corresponding point. The common solutions are kernel mean matching and Kullback-Leblier (KL) importance estimation [3]. Alternatively, data points in the source and target domain can be transformed into a new representation space to minimize their distribution disparity. Various algorithms have been proposed to learn this transformation, such as transfer component adaptation and low-rank reconstruction [13]. The above methods can be involved as a loss function during the NN training. Qian et al. [14] introduced KL divergence in the loss function for unsupervised discriminative feature learning in bearing fault diagnosis. The disparity between the source and target domain is caused by variations of rotating speeds and loads. Chai et al. [15] proposed an adversarial network-based domain adaptation method to tackle covariate shift caused by variations of the fault size and the operating speed. In the domain generalization setting, multiple labeled source domains are required for training. The trained model can be directly deployed for the unseen target domain. Li et al. [16] and Liao et al. [17] proposed adversarial network based methods to extract domain-general features for bearing fault diagnosis under variable speeds. Zheng et al. [18] transferred the source and target domain data into a Grassmann manifold as the domain-general features for bearing diagnosis. The proposed method was validated under variations of operating speeds and sensor locations.

A prerequisite of the above approaches is the availability of multiple domains. Unfortunately, for real-world MFD problems, collecting faulty data in different operating conditions is highly laborious and costly. This is particularly true in the railway sector due to additional operation and safety issues. Therefore, model-based approaches using multibody dynamic simulation (MBS) are the standard methods to investigate railway vehicles' abnormal behaviors under faulty conditions. However, the simulation data can hardly represent real-world complexity and is insufficient to train a robust ML diagnostic model. We have been attempting to take advantage of data-driven and model-based approaches for MFD in railway applications [19]. In this paper, we propose a new data augmentation technique and a novel data augmentation framework to tackle the robustness problem from its root, i.e. lack of faulty data in various conditions. Next, previous work related to data augmentation is reviewed.

1.2 Related work

Synthetic data can be generated by manipulation of the existing data based on domain knowledge. For image data, the common manipulation is geometric transformation, cropping, rotation and flipping [20]. Images can also be corrupted with noise, blur and compression for data augmentation [21]. For acoustic data, the variations of vocal track length, stochastic feature mapping and speech rate-distortion produce synthetic data [22]. For time series, similar approaches such as windows cropping, warping, flipping and noise injection can be applied in the time, frequency or time-frequency domain [23]. Time series averaging derives the average time series from the existing data and use it as the generated synthetic data. Forestier et al. [24] proposed weighted dynamic time warping Barycenter averaging (WDBA) to increase the synthetic data's diversity by weighting the individual data points in the average time series. For MFD, segments of faulty vibration data were stretched or cropped along the time axis [16]. Afterward, interpolation/extrapolation was applied on the stretched/cropped segments to recover the time length. This augmentation method can simulate the variation of operating speeds. The same concept was implemented by resampling faulty data in the time and frequency domain [25]. Apart from time stretching and resampling, Li et al. [26] empirically studied the effect of noise injection, signal translation and amplitude shifting for bearing fault diagnosis. These techniques introduce stochastics into the original signals, simulating stochastic background noise. Meng et al. [27] proposed to divide a data sample into multiple segments. Each sample contains data points over one rotation. The segments can be recombined randomly to create the synthetic data samples, having certain divergences from the original ones. Based on this, Yu et al. [28] applied the temporal flip of the local segments prior to signal recombination.

Synthetic data can be generated by learning the existing data representation. Generative adversarial networks (GANs) [29] use two neural networks contesting with each other to generate synthetic data. GANs have been widely used to generate synthetic images, acoustic and time series [20-23]. For MFD, various GANs were used to address the data imbalance problem, such as Wasserstein GAN with gradient penalty [30], dual discriminator conditional GAN [31], conditional-deep convolutional GAN (C-DCGAN) [32] and conditional variational autoencoder GAN [33]. Apart from GANs, Han et al. [34] applied a stacked autoencoder (SAE) for data augmentation, where the reconstructed signals outputted by the SAE’s decoder were regarded as synthetic data. Regardless of different GAN or AE variations, the learning-based method requires a few labeled data in one domain for training to generate more data in the same domain. They alone are not able to generate the unseen-domain data.

Synthetic data can be generated by modeling and simulation. Model-based augmentation methods usually incorporate statistical modeling of time series data, such as Gaussian trees and autoregressive models [23]. The MFD tasks concern physical systems, which allow physical modeling. This concept is well known as the model-based approach for MFD. Traditionally, the model outputs are compared with the measurement outputs for diagnosis [35]. The compared value may be a threshold of residuals or similarities between the simulated and measured outputs. One outstanding advantage of the model-based approaches is that arbitrary operating and faulty conditions can be simulated as long as an accurate model has been built and validated. Intuitively, physical models can be employed for data

augmentation. Sobie et al. [36] built a bearing dynamics model based on second-order ordinary differential equations. They investigated the cross-dataset robustness, where a model was trained on one dataset and tested on a different one. It was found that ML MFD methods achieved better performance when the models were trained on the simulation dataset and tested on the experimental dataset than trained on one experimental dataset and tested on others. Gao et al. [37] combined finite element method (FEM) simulations and GAN for data augmentation. FEM generated simulation data for the unknown fault types that cannot be collected from the real world. GAN is then applied to the mixture of the simulated and measured faulty data for data augmentation.

1.3 Contributions

In light of our previous work and the above review, we propose a novel data augmentation framework for robustness enhancement in railway MFD applications. It mainly consists of MBS and fast weighted feature-space averaging (FWFSA), which is a new time series averaging technique developed in this work. MBS simulates arbitrary faulty and operating conditions, while FWFSA mimics the realistic background from a few measurement data to produce infinite reality-augmented simulation data (RASF).

The intuition behind our proposal is a step forward from the existing thoughts in [36,37]. Sobie et al. [36] suggested training the diagnostic models on the pure simulation data. Gao et al. [37] suggested a mixture of the GAN-augmented simulation data and real-world data. The latter is closer to our proposal. The first difference is that we avoid the involvement of GAN. Training a GAN model requires a large amount of training data with adequate diversity and a high effort for hyperparameter tuning to converge the training loss. This seems paradoxical to simulation. Running simulations to generate sizeable training datasets is very time-consuming. Furthermore, data generated under one simulation condition lacks diversity. The simulation tools typically involve stochastics by probability density function to enrich data diversity, such as track irregularities within the railway MBS task. This can hardly enrich the patterns of faulty signals. The second difference is that we add the learned realistic background information into the simulated faulty data instead of directly using the simulated faulty data for training. This process is like noise injection, where FWFSA learns the noise from real-world data.

Previous work validated the proposed data augmentation methods by comparing test accuracies of diagnostic models with and without data augmentation. The investigated operating conditions mainly refer to operating speeds and loads. In this work, we systematically investigate the robustness problems caused by condition variations and conduct extensive validation experiments. Also, we propose a metric for robustness evaluation, taking into account several types and severities of condition variations. The experiment data was measured on railway freight wagons by our research team in past years. The measurements were conducted on different positions on different wagons running on different track lines in different speed ranges with different measurement systems. The obtained data can undoubtedly represent real-world complexity. The faulty signals are induced by wheel flat (WF), a flat spot on railway vehicles' wheel tread. MFD is referred to as WF detection in this study, i.e. a binary classification problem. Due to data availability, we did not conduct the study for multi-class classification. Nevertheless, the proposed data augmentation framework MBS-FWFSA and evaluation procedures are not limited to binary classification. Furthermore,

our method serves as a complement to other measures for robustness enhancement. For instance, domain adaptation and generalization can be applied to the synthetic domains produced by the proposed MBS-FWFSA to improve the model robustness further. Our contributions in this paper can be summarised as follows:

1) We propose a novel data augmentation framework, combining MBS and FWFSA to produce infinite RASF data for robustness enhancement of MFD models. The proposed MBS-FWFSA outperforms three state-of-the-art methods, i.e. C-DCGAN [32], SIM [36], and SIM-GAN [37].
2) We propose a new time series averaging method FWFSA, inspired by WDBA.
3) We systematically investigate the robustness problem caused by condition variations and the effect of data augmentation through extensive experiments.
4) We define a proper metric evaluating model robustness under several condition variations.
5) We make our code and data available at https://github.com/quickhdsdc/Robustness-Enhancement-of-Machine-Fault-Diagnostic-Models.

The rest paper is structured as follows. Section 2 formulates the problem and briefly introduces the theoretical background. Section 3 details the proposed data augmentation framework MBS-FWFSA. Section 4 describes the procedures and preliminaries of the experiments. Section 5 presents the experiment results. Section 6 discusses the experimental findings. Section 7 draws the conclusions.

2. Preliminaries

2.1 Problem formulation

Let $D_s^r = \{(x_{s,i}^r, y_{s,i}^r)\}_{i=1}^I$ be the source domain consisting of I real-world data samples measured under one operating condition, where $x_{s,i}^r \in X_s^N$ and $y_{s,i}^r \in Y_s^C$. N denotes the dimension of the data samples. C is the number of health conditions. In our case, it concerns WF detection, i.e. $C = 2$. The joint probability distribution $p(x, y)$ refers to the probability of finding the pair (x, y) in the labeled feature space $X \times Y$. It can be decomposed as $p(x, y) = p(x)p(x|y)$, where $p(x)$ is the marginal distribution and $p(x|y)$ is the conditional distribution. A diagnostic model f is to predict y given a data sample x. When the diagnostic model $f(x, y)$ is trained and tested in the same domain D_s^r, the test accuracy is referred to as clean accuracy ACC_C^f. The variations of operating conditions lead to covariate shift, defined as the marginal distribution shift within the source domain $p_s^r(x)$. It results in $S \times T$ target domains, i.e. $D_{t1}^{r1}, D_{t1}^{r2}, \cdots, D_{tV}^{rS}$, where $p_s^r(x) \neq p_{tv}^{rs}(x)$. S denotes the number of variation severities, while V is the number of variation types. The disparity between the source domain and a target domain can be measured by the distribution distance $DD(p_s^r, p_{tv}^{rs})$. The diagnostic model $f(x, y)$ trained in the source domain D_s^r surfers from performance impairment, when $f(x, y)$ is directly deployed in a target domain D_{tv}^{rs} under the v-th condition variation with the s-th severity. The impaired accuracy is denoted as $ACC_{v,s}^f$. The robustness of $f(x, y)$ under this condition is $RB_{v,s}^f$, given by:

$$RB_{v,s}^f = 1 - (ACC_c^f - ACC_{v,s}^f) \tag{1}$$

$RB_{v,s}^{f}$ is a value in the range (0,1), and a higher value of $RB_{v,s}^{f}$ indicates the higher robustness. The actual robustness should be represented by its mean robustness under different condition variations with different severities. The mean robustness mRB^{f} for the diagnostic model $f(x,y)$ is obtained by:

$$mRB^{f} = \sum_{v=1}^{V}\sum_{s=1}^{S} RB_{v,s}^{f} \Big/ (V \cdot S) \tag{2}$$

The objective of data augmentation is to generate $\tilde{S} \times \tilde{T}$ synthetic source domains, i.e. $D_{s1}^{a1}, D_{s1}^{a2}, \cdots, D_{s\tilde{V}}^{a\tilde{S}}$, in order to improve the mean robustness mRB^{f}. As the actual condition variations in practice cannot be completely simulated, it results in $\tilde{S} \times \tilde{T} \subset S \times T$. In addition, the effect of data augmentation can be reflected by the distribution distance $DD(p_{tv}^{rs}, p_{s\tilde{v}}^{a\tilde{s}})$. An effective data augmentation method should generate the synthetic source domains with a small value of $DD(p_{tv}^{rs}, p_{s\tilde{v}}^{a\tilde{s}})$.

2.2 Maximum mean discrepancy

The distribution distance between two domains can be measured by maximum mean discrepancy (MMD), which is a common nonparametric kernel embedding-based distance measure. MMD is defined as the distance between the means of two domains' data samples mapped into a reproducing kernel Hilbert space (RKHS), given by [38]:

$$MMD(p_s, p_t) = \|\mu[p_s] - \mu[p_t]\|_{\mathcal{H}} \tag{3}$$

where p_s is the probability distribution of the source domain, and p_t is the probability distribution of the target domain. $\mathcal{H}$ denotes the RKHS.

In this work, we use the empirical estimate of MMD using Gaussian kernel for embedding, given by:

$$MMD(X_s, X_t) = \left[\frac{1}{N^2}\sum_{i,j=1}^{N}\varphi(x_{s,i}, x_{s,j}) - \frac{2}{NM}\sum_{i,j=1}^{N,M}\varphi(x_{s,i}, x_{t,j}) + \frac{1}{M^2}\sum_{i,j=1}^{M}\varphi(x_{t,i}, x_{t,j})\right]^{1/2} \tag{4}$$

where $x_s \in X_s^N$, $x_t \in X_t^M$, and φ is the Gaussian kernel.

2.3 Weighted dynamic time warping Barycenter Averaging

Dynamic time warping Barycenter averaging (DBA) is a technique for averaging a set of time series. Let $x \in X^N$ be a set of time series in the vector space E (usually a Euclidean space) induced by dynamic time warping (DTW). DBA is to calculate the average time series iteratively $\bar{x}$ from X^N by minimizing (5).

$$argmin\ \bar{x} \in E \sum_{i=1}^{N} DTW^2(\bar{x}, x_i) \tag{5}$$

This average time series $\bar{x}$ is the produced synthetic data. Weighted DBA (WDBA) was proposed by Forestier et al. [24] for time series data augmentation. It averages and reweights a series of data samples to generate averaged time series as the synthetic data. Let $D =$

$\{(x_1, w_1), \ldots, (x_I, w_I)\}$ be the weighted time series domain, containing I time-series samples. WDBA is to minimize (6).

$$argmin\ \bar{x} \in E \sum_{i=1}^{N} w_i \cdot DTW^2(\bar{x}, x_i) \tag{6}$$

where x_i is the ith time series sample within D, and w_i is the ith weight.

The initial average time series $\bar{x}_0$ is the medoid of the dataset D, defined by (5). The weights w_i are defined as follows. $\bar{x}_0$ is assigned with a weight of 0.5. Then, its five nearest neighbors with respect to the closest distance in the space E are searched. Two out of five neighbors are randomly selected and assigned with a weight of 0.15, respectively. The rest data within D share the remaining weight of 0.2. WDBA iteratively updates the average time series $\bar{x}$ in $J \times I$ steps. At each step j, each data point $\bar{x}_{j,n}$within $\bar{x}_j$ is aligned by its DTW alignment (DTWA) between $\bar{x}_j$ and each time series sample x_i within X, calculated by eq. (7). The sum of the weights is recorded for each step j by (8). Each data point $\bar{x}_{j,n}$within $\bar{x}_j$is finally calculated by (9). This process is repeated for J times to obtain the target $\bar{x}$.

$$\bar{x}_{j,n} = \bar{x}_{j-1,n} + DTWA(\bar{x}_{j-1}, x_i)_n \cdot w_i \tag{7}$$

$$w_{j,sum} = \sum_{i=1}^{I} \sum_{n=1}^{N} w_{i,n} \tag{8}$$

$$\bar{x}_{j,n} = \bar{x}_{j,n} / w_{j,sum} \tag{9}$$

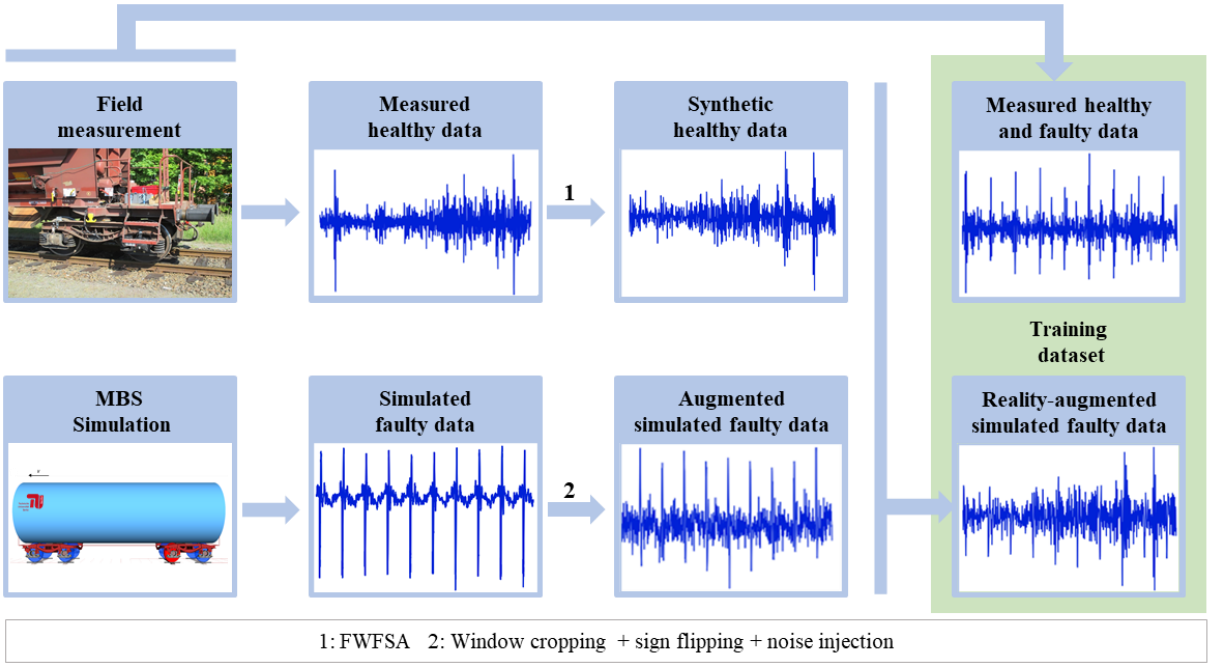


Figure 1. Procedure of the proposed MBS-FWFSA-based data augmentation framework to produce reality-augmented simulation data

3. The Proposed MBS-FWFSA based data augmentation framework

3.1 General procedure

Figure 1 presents the proposed data augmentation framework. First, we produce an arbitrary amount of synthetic healthy data using FWFSA. Given a set of real-world data as the source domain D_s^r, we select a specific subset D_s^{r0} of healthy data with $y_{s,i}^r \in Y_s^0$, which can reflect the specific real-world operating environment, such as tight curves, turnouts, severe track irregularities, and rail detects. Such unusual healthy data is prone to be the corner case in railway MFD applications, as it shows different patterns from the usual healthy data and is the minority of the healthy data. Applying the proposed FWFSA on the subset D_s^{r0} produces infinite synthetic measured healthy data $D_s^{a0} = \{(x_{s,i}^a, y_{s,i}^a)\}_{i=1}^m$ with $y_{s,i}^a \in Y_s^0$, which mimics the given dataset with sufficient diversity and enlarges the number of unusual corner cases.

Second, we build an MBS model of a typical freight wagon with a specific failure (i.e. WF in our case). This physic model generates continuous vibration data under various operating conditions, e.g. vehicle speed, track layout, and WF size. As the simulation is very time-consuming and the simulation data under one condition has high similarity, we simulate each operating condition merely for a short duration to obtain representative simulation data. To produce an arbitrary amount of the simulated faulty data $D_s^{a1} = \{(x_{s,i}^a, y_{s,i}^a)\}_{i=1}^m$ with $y_{s,i}^a \in Y_s^1$, we apply window cropping, sign flipping and noise injection [23] on the simulation data.

Finally, the RASF data D_s^a is a combination of the D_s^{a0} and D_s^{a1} with a random pick and a scaling factor s. We randomly pick the i-th data sample from the normalized D_s^{a0} and j-th data sample from the normalized D_s^{a1}. The k-th data sample within D_s^a can be obtained by eq. (10). A higher value of the scaling factor s indicates a lower signal-to-noise ratio. We use $s = 1$ in the case of WF detection using axlebox acceleration.

$$x_{s,k}^a = x_{s,j}^{a1} + s \cdot x_{s,i}^{a0} \tag{10}$$

Carrying out various simulations under different conditions, we obtain arbitrary synthetic source domains $D_{s1}^{a1}, D_{s1}^{a2}, \cdots, D_{s\tilde{V}}^{a\tilde{s}}$. The final training data is a mixture of synthetic and original source domains. It is worth noting that the original source domains typically contain much more healthy data than faulty data. The synthetic faulty data can compensate for the unbalance within the original source domains and supplement the unknown fault information under various operating conditions. Next, the MBS model and the FWFSA technique are described in detail.

3.2 Multibody dynamic simulation model

In our previous work [19], we built an MBS model of a tank wagon in the commercial MBS software SIMPACK, see Figure 2. The model comprises 15 rigid bodies: one car body, two bogie frames, four wheelsets and eight axleboxes. The rigid bodies are connected with parameterized force elements and constraints, representing the physic characteristics of spring and damping elements. More details of the vehicle model and the modeling parameters can be found in [19]. The wagon bogie is the European standard Y25 bogie for freight wagons. Its suspension parameters have been empirically derived from the laboratory test. The entire wagon model has been validated through natural-frequency measurements in the previous projects, as shown in Figure 3.

WF is one of the most common and critical failures on the wheel tread, typically induced by abnormal tread braking. A fresh WF can be simplified as a chord of the wheel circle, illustrated as AOB in Figure 4 (a). As a result of the wear process, the fresh WF becomes a worn WF in the short term, illustrated as $A'OB'$ in Figure 4 (a). We modeled the worn WF (also termed as haversine flat) by eq. (11) and (12).

$$\Delta d = -d_f\left[1 - cos\left(2\pi x_f / L_f\right)\right]/2 \tag{11}$$

$$d_f = L_f^2/(16R) \tag{12}$$

where Δd denotes the variation of the wheel radius, L_f the WF length, d_f denotes the WF depth, x_f the longitudinal distance along the WF, and R the wheel radius.

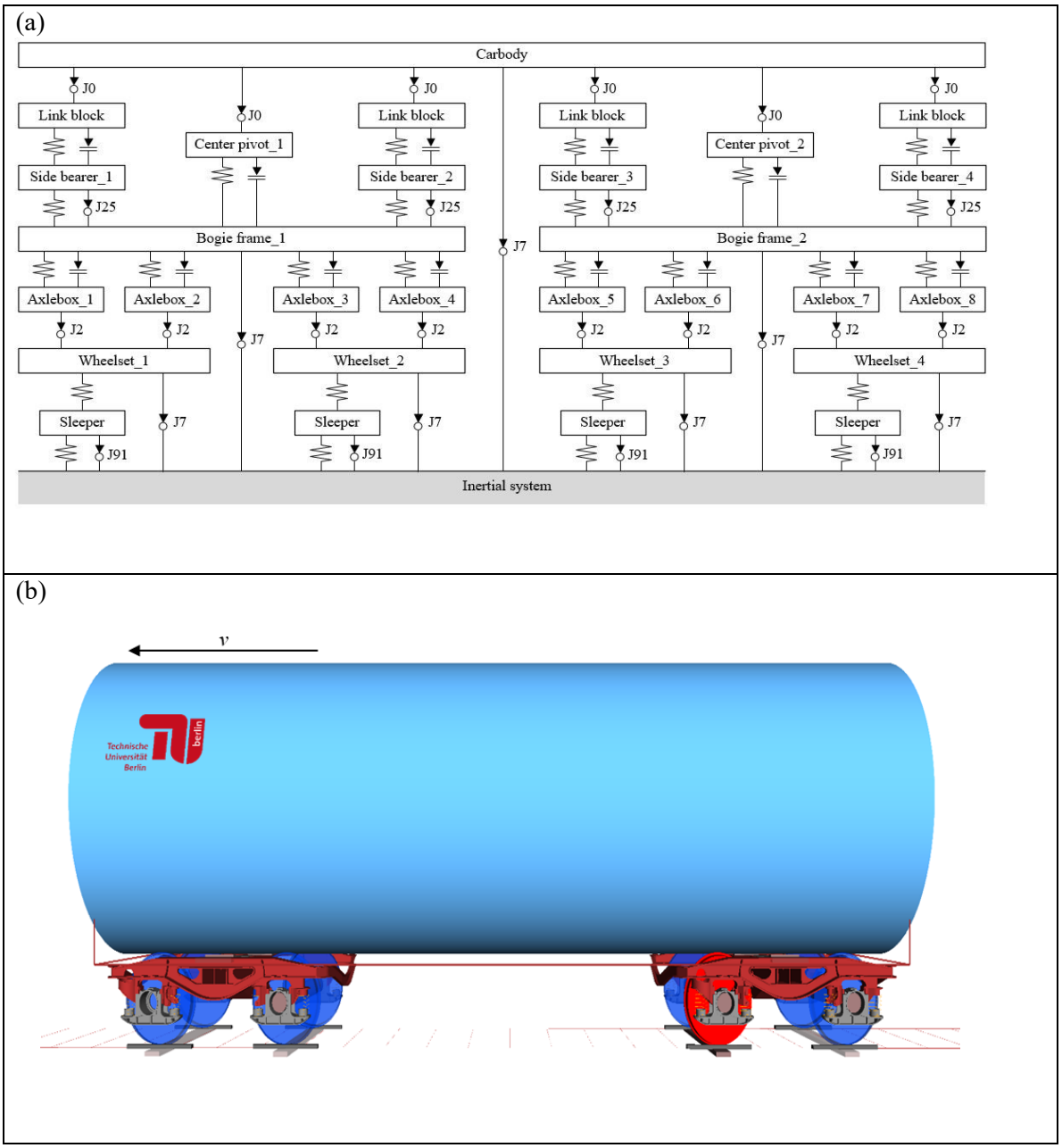


Figure 2. MBS model of a freight wagon with Y25 bogies (a) topology diagram (b) MBS model in SIMPACK

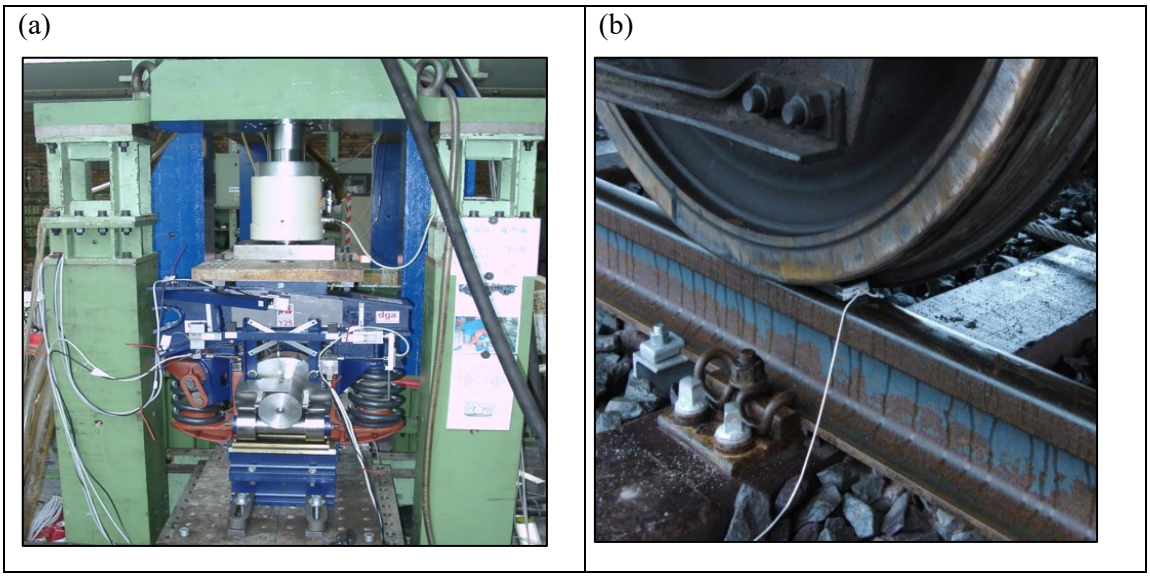


Figure 3. (a) Laboratory measurement of Y25 suspension parameters; (b) Measurement of vertical natural frequencies using a wedge for model validation

3.3 Fast weighted feature-space averaging

The original WDBA has two problems for MFD applications. Firstly, WDBA has to calculate the dataset medoid using DTW. The complexity of the medoid calculation is $O(I^2)$, where I is the total number of time series samples within the dataset. For each sample, one DTW with the complexity of $O(N^2)$ is repeated, where N is the dimension of a time series sample. The total complexity is $O(I^2N^2)$. For MFD problems, the input data is typically waveform time series sampled at high frequencies (commonly over 5 kHz), which means a large value of N and thus a vast computational complexity. Secondly, WDBA calculates the DTW distance between $\bar{x}_{j-1}$ and each data x_i within the set T at each iteration step j, as shown in (7). This is based on the hypothesis that each time series x_i within the domain D is similar. However, this hypothesis may be invalid in MFD applications. Due to strong interferences and background noise, the time series x_i within the healthy dataset may have very different shapes and thus large DTW to each other.

We propose FWFSA to solve the above issues for typical MFD input data, i.e. high-frequency waveform time series data. The essential intuition behind FWFSA is to use the low-dimensional feature space as the representation of the raw time series for distance calculation. This vastly reduces the data dimensionality and avoids invoking DTW. The distance between the time series is defined as the Euclidean distance between their features. The issues caused by strong interferences and background noise are also avoided by using the feature representation instead of the raw waveform data. The pseudocode of FWFSA is given in Table 1. The functions for feature extraction are given in Table 2. These functions are applied for the raw data in the time domain, the envelope spectrum and the scale-averaged wavelet spectrum, generating 21 features for each time series. We choose the above-defined statistical features and the time-frequency analysis techniques, as they are commonly used for MFD. Nevertheless, the proposed FWFSA is not confined to them.

Table 1. FWFSA Algorithm

Algorithm: FWFSA

Input: domain D to average

Input: defined functions $f_m(x)$ for feature extraction

Input: number of iterations J

Output: averaged $\bar{x}$ after J iteration

$F = \{f_1(x_1), \dots, f_M(x_1), f_1(x_2), \dots, f_M(x_2), \dots, f_M(x_I)\}$ be domain D's M-dimension feature space

$\bar{x}_0$ be the initial average time series randomly selected from D

for $j = 1 \rightarrow J$ **do**

  for $i = 1 \rightarrow I$ **do**

    $w_i = \sqrt{\sum_{m=1}^{m=M}(f_m(\bar{x}_{j-1}) - f_m(x_i))^2}$ be the weight for x_i

  end

  $w_{i'} = [w_i - min\ (w_i)]/[max(w_i) - min\ (w_i)]$ be the weights rescaled by min-max normalization

  $\bar{x}_j = \sum_{i=1}^{i=I}(w_{i'} \cdot x_i)$ be the updated average time series

end

4. Experiment design

Condition variations lead to dataset shift, incl. covariate shift, prior probability shift and concept shift. If the learning task remains unchanged, the problem is usually simplified as covariate shift [3]. There are different types of condition variations with different severities in practice. How do they affect covariate shift and diagnostic performance? Can data augmentation compensate for covariate shift and improve diagnostic performance? To answer these questions, we start with quantification of covariate shift, where MMD is employed as the measure. Next, the diagnostic performance under different condition variations is investigated. Several baselines incorporating the most common signal processing and ML techniques are tested as the representative MFD methods. Finally, the effect of data augmentation is validated by examining covariate distribution and diagnostic performance. On the one hand, the generated synthetic data is exemplarily visualized, and its effect on covariate shift is visualized and measured by MMD. On the other hand, the MFD baseline performance is tested with and without data augmentation. In this experiment, the proposed MBS-FWFSA is compared with three state-of-the-art augmentation methods, i.e. C-DCGAN [32], SIM [36], and SIM-GAN [37].

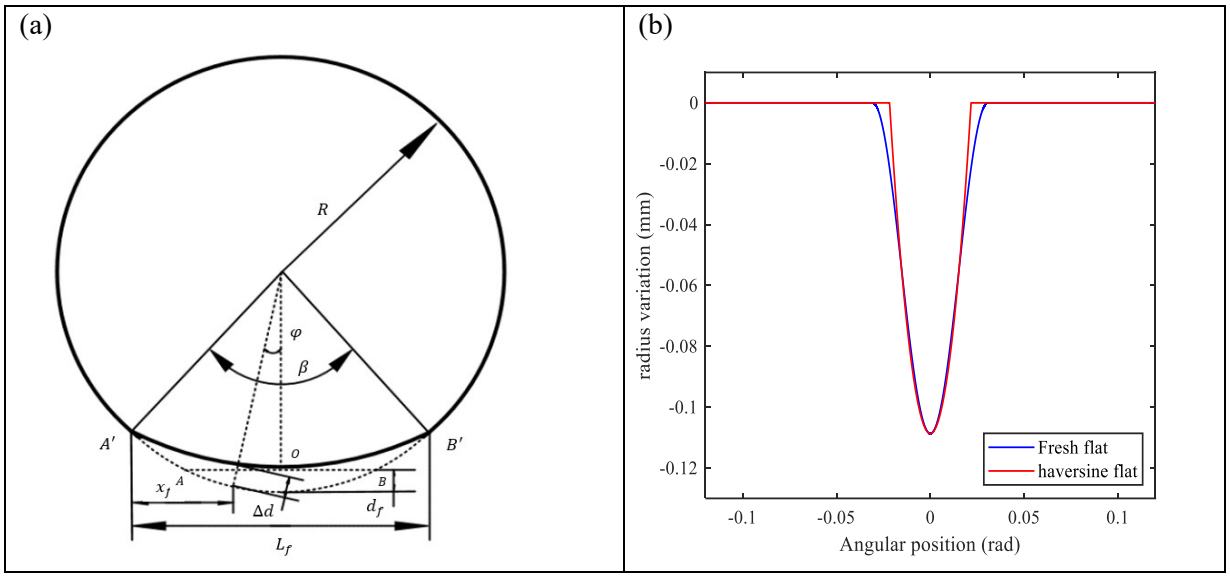


Figure 4. (a) Comparison between a fresh WF (AOB) and a worn WF ($A'OB'$) on a wheel; (b) Modeled fresh and worn (/haversine) WF ($L_{AOB} = 20\ mm$, $L_{A'OB'} = 28.3\ mm$) [19]

Table 2. Functions for feature extraction

Features	Definition	Formula
$\boldsymbol{f_1}$	Peak-to-Peak value	$max(x_i) - min\ (x_i)$
$\boldsymbol{f_2}$	Root Mean Square	$(\sum_{i=1}^{N} x_i^2/N)^{1/2}$
$\boldsymbol{f_3}$	Kurtosis value	$\sum_{i=1}^{N}[(x_i - \bar{x})/\sigma]^4/N$
$\boldsymbol{f_4}$	Impulse factor	$max(x_i)/(\sum_{i=1}^{N}\lvert x_i\rvert/N)$
$\boldsymbol{f_5}$	Margin factor	$max(x_i)/(\sum_{i=1}^{N}\sqrt{\lvert x_i\rvert}/N)^2$
$\boldsymbol{f_6}$	Shape factor	$max(\lvert x_i\rvert)/f_2$
$\boldsymbol{f_7}$	Crest factor	$max(\lvert x_i\rvert)/f_2$

The overall experiments are conducted on a railway MFD application, i.e. WF detection by axlebox vibration monitoring on freight wagons. The common condition variations during the regular railway operation are categorized into three groups. In the following subsections, the experimental data, the defined condition variations in railway operation, the baseline MFD methods, and the baseline data augmentation methods are introduced.

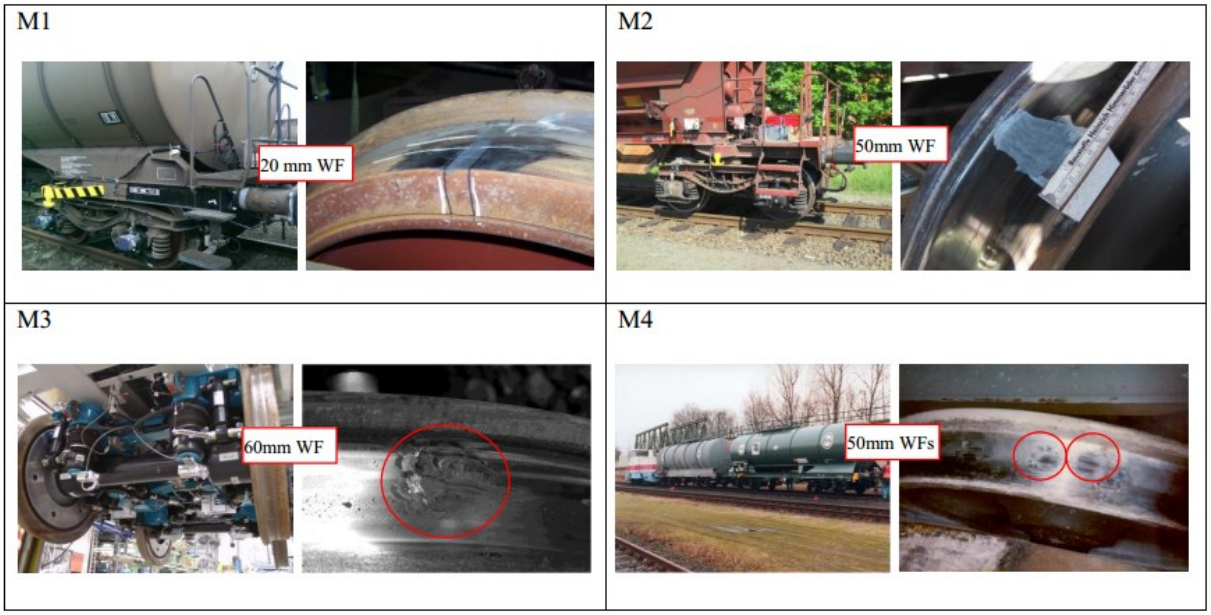


Figure 5. Field measurements. The first two measurements M1 and M2 were executed with synthetic WFs. In the last two measurements M3 and M4, WFs were accidentally generated in the braking process.

Table 3. Overview of the usable field data after data cleaning

Measurement	WF conditions	Speed range	Sample frequency	Sample number
M1	synthetic, 20 mm	15 – 105 km/h	5000 Hz	11817
M2	synthetic, 50 mm	15 – 45 km/h	5000 Hz	911
M3	natural, 60 mm	15 – 85 km/h	500 Hz	7533
M4	natural two WFs, 50 mm	15 – 55 km/h	500 Hz	231

4.1 Field measurements on freight wagons

The experimental data stems from four field measurements conducted on different wagons running on different lines in different speed ranges with different measurement systems, see Figure 5. The obtained data can undoubtedly represent real-world complexity. In M1 and M2, a single WF with different lengths was synthetically produced by grinding the wheel surface. M1 was performed on the mainline in regular railway freight operation for two days. M2 was performed in the urban area for several hours. M3 was carried out to test a newly designed freight wagon bogie on the mainline. A WF was occasionally generated. M4 was carried out for a derailment test on a shunting yard at low speeds. Two adjacent WFs were generated by the emergency brake. All datasets contain healthy and faulty data. Table 3 gives an overview of the measurement conditions. The continuous vibration data is segmented into 2.048-seconds time-series samples without overlap. The data samples of M3 and M4 are upsampled to 5000 Hz by interpolation. More details of data availability in each speed range can be found in Table 13 in Annex I.

4.2 Condition variations

The variation of vehicle speeds affects WF peaks' amplitude [45] and the faulty frequency. In MFD applications, the continuous raw signal is usually divided into small segments through a sampling time window. One segment is used as one data sample for the diagnostic models. Given a pre-defined length of the time window, the number of WF peaks increases with increased speeds. Also, the amplitude of vibrations and WF peaks increases with increased speeds. Figure 6 exemplarily shows two samples of the axlebox vibration signal in the time domain, the envelope spectrum, and the wavelet scalogram at 20 km/h and 100 km/h. The WF patterns, i.e. periodic peaks, are clearly visible at both low and high speeds, and significantly vary with the vehicle speed.

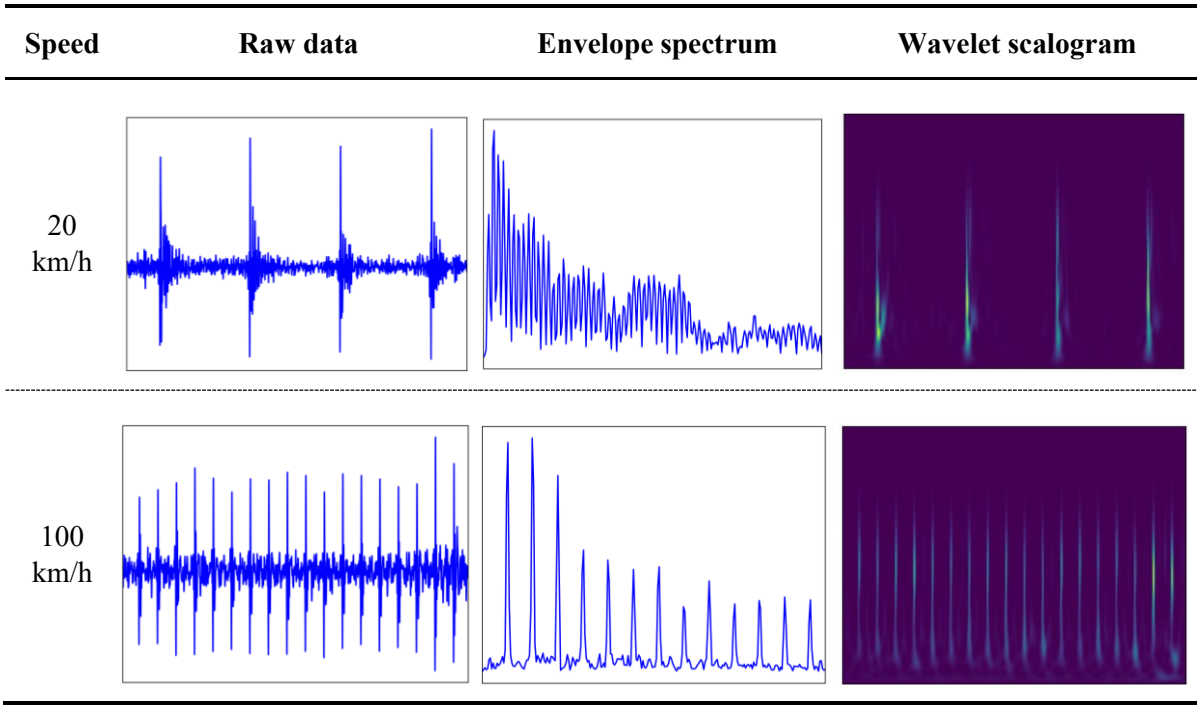


Figure 6. Exemplary comparison of raw axlebox acceleration, its envelope spectrum through Hilbert-transform, its wavelet scalogram at 20 km/h and 100 km/h, respectively

The variation of monitoring objects entails differences in the measured signals, relying on the object conditions. In the context of WF detection for railway vehicles, it mainly concerns the vehicle and fault conditions. The faulty condition includes WF geometry and distribution on the wheel surface. The vehicle condition is determined by loading conditions, wear conditions and vehicle characteristics such as bogie types. These conditions exert a substantial influence on vehicle dynamics and thus the vibration manifestation. Figure 7 exemplarily shows four samples in M1-M4, respectively. In M1 and M2, it concerns a single synthetic WF. In M3, a natural WF was generated by braking during the operation. In the case of a single WF, the amplitude of the WF peaks increases with the increased WF size. The two adjacent WFs in M4 lead to a different shape of WF peaks, where two oscillations overlap. Therefore, the amplitude of the individual peaks is much lower than that of a single WF of the same size. However, the disparities caused by object variations are visually marginal.

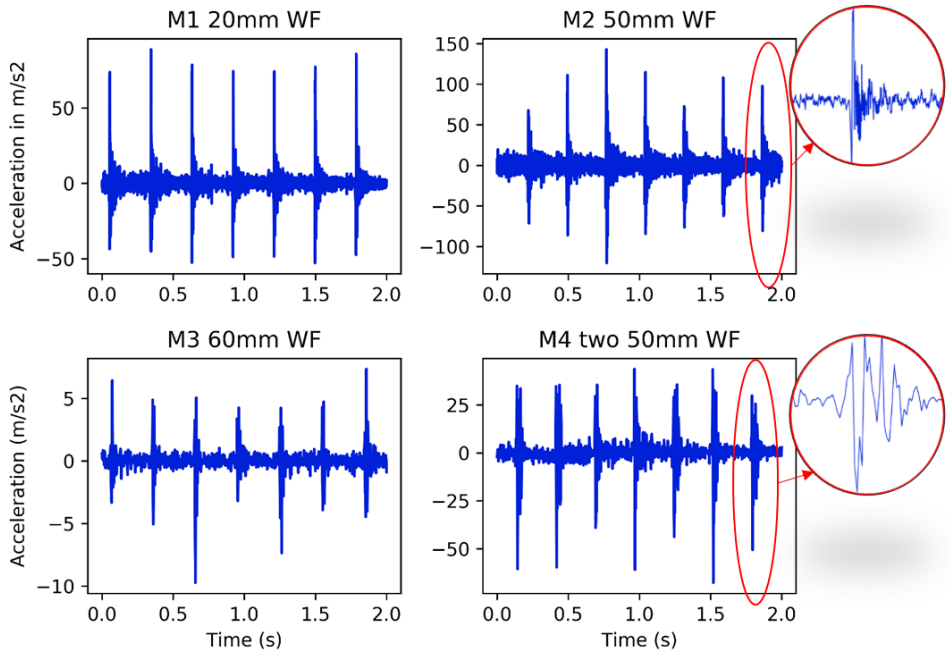


Figure 7. Exemplary comparison of raw axlebox accelerations at the same vehicle speeds in M1-M4 respectively, representing the variation of vehicle and fault conditions

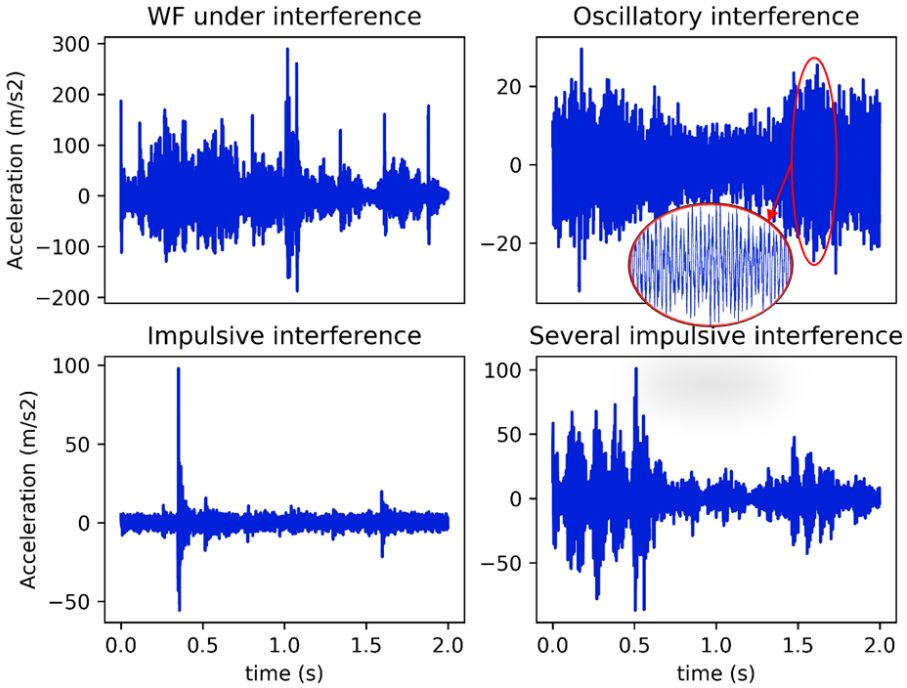


Figure 8. Axlebox accelerations under different signal interferences

The variation of signal interferences may disturb the regular patterns of healthy and faulty signals. The interferences are commonly induced by severe track irregularities and discrete rail detects. For instance, severe vertical track irregularities cause oscillatory interferences in the vertical axlebox acceleration, as shown in the upper right diagram in Figure 8. A discrete rail defect such as rail squats and broken rails or the turnout in a switch and crossing area

could result in an impulsive interference, as shown in the lower-left diagram. Several adjacent impulsive interferences form a periodic-like pattern, which may mislead the diagnostic models. These interferences may also disturb the WF pattern, as shown in the upper left diagram.

4.3 MFD baselines

Fault diagnosis is a classification problem in general. Deep convolutional neural network (CNN) has been the state-of-the-art approach for classification in various domains. In modern CNN architectures, the residual block in ResNet [39] has become a standard component. In our previous work [40], we have proven that residual blocks help WF detection. Therefore, we adapted a 1D ResNet [41] to an MFD baseline. The architecture of ResNet is shown in Figure 9 and Table 14 in Annex II. The first two convolutional blocks are composed of the convolution layer, batch normalization layer and activation function of ReLu, followed by the max-pooling layer for downsampling. The subsequent two blocks are residual blocks with the residual connection for feature extraction. The feature map in each channel is squeezed by average pooling into one scale value fed into a fully connected network for classification. ResNet takes the 1D raw vibration data in the time domain as the inputs. Each input sample has 10240 data points.

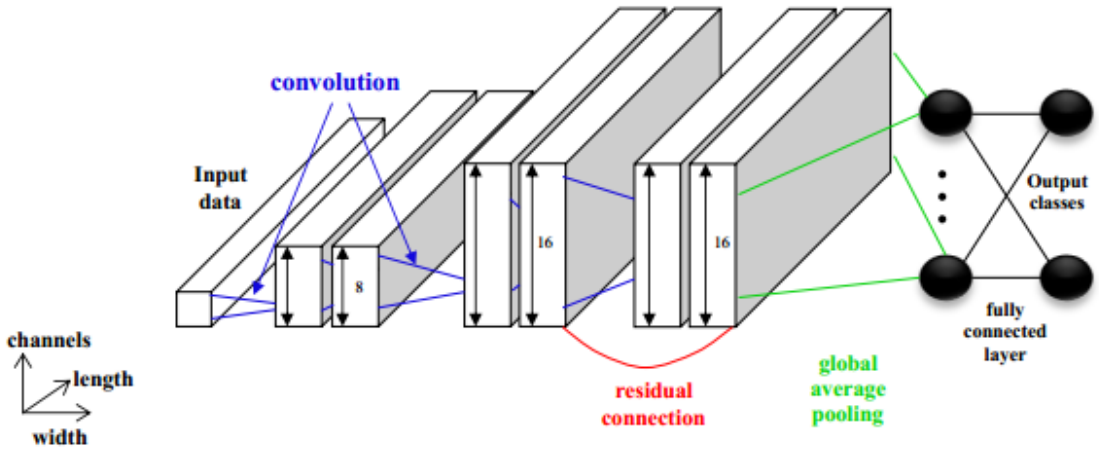


Figure 9. Architecture of the modified ResNet for time series classification (modified from [41])

We also consider whether signal processing techniques can affect the model robustness. In the second baseline CWT+ResNet [42], we use the continuous wavelet transform to obtain the scale-averaged wavelet spectrum as the input for a 1D ResNet. Each sample has 10240 data points as well. Therefore, ResNet and CWT+ResNet share identical architecture. In the third baseline HT+ResNet [43], we use the Hilbert transform to obtain the envelope spectrum as the input for a 1D ResNet. Each sample has 256 data points. A dedicated ResNet has been designed for this input dimension. Apart from the ResNet-based baselines, a classic ML method using manually defined features and gradient boosting decision tree (GBDT) [44] is included as well. 21 statistical features are defined by applying the functions given in Table 2 in the time domain, the envelope spectrum and the scale-averaged wavelet spectrum. The hyperparameters of all baselines are optimized on the M1 dataset using Bayesian optimization for parameter searching and five-fold cross-validation to prevent overfitting [40], ensuring a

high clean accuracy ACC_c^f for the WF detection task. The fine-tuned architecture and hyperparameters of each ResNet and GBDT are given in Table 14, Table 15 and Table 16 in Annex II.

4.4 Data augmentation baselines

The proposed MBS-FWFSA is compared with three state-of-the-art augmentation methods, i.e. C-DCGAN [32], SIM [36], and SIM-GAN [37]. The original SIM uses the pure simulation data as the additional synthetic source domain. In our experiments, SIM is improved by applying window cropping, sign flipping and noise injection on the simulation data to produce an arbitrary number of the augmented simulation data. This augmentation process is actually the second part of MBS-FWFSA. Comparing MBS-FWFSA with SIM can be regarded as an ablation study and reveal the role of FWFSA in the entire framework. C-DCGAN is directly applied to the original source domain to create a new synthetic source domain. Its architecture is adapted to our datasets by hyperparameter tuning. The re-implemented architecture can be found in Table 17 in Annex III. SIM-GAN applies a vanilla GAN to the measurement and simulation data to create more data. The original GAN is modified as a deep convolutional GAN (DCGAN), sharing the same architecture of C-DCGAN, but with different training settings. The details can be found in Table 18 in Annex III.

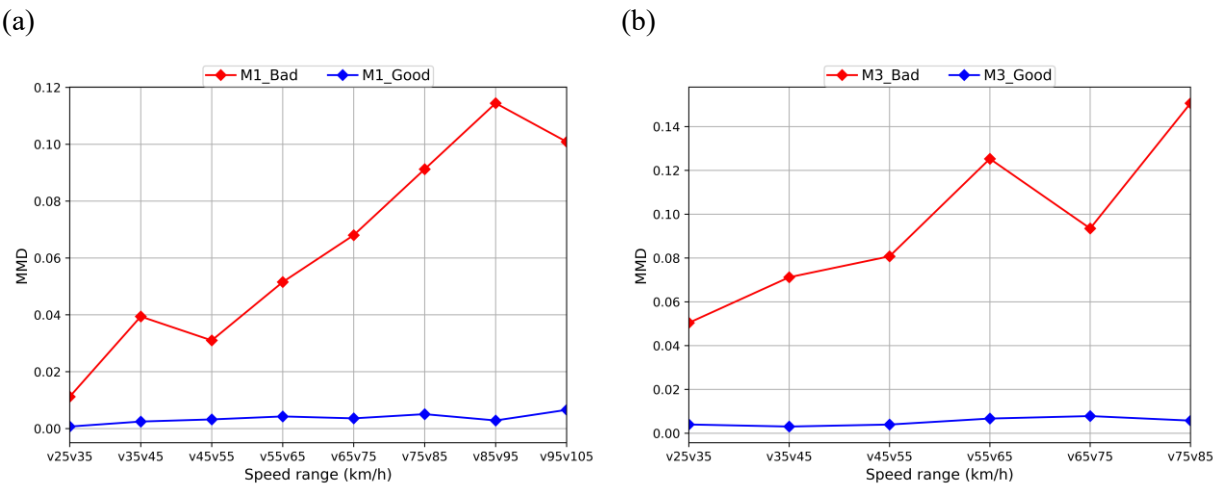


Figure 10. (a) MMD between the source domain D_s^r at 15-25 km/h and the target domains D_{t1}^{rs} at higher speeds in M1; (b) MMD in M3. "Good" refers to data under the healthy condition, while "Bad" refers to data with WFs.

5. Experiment results

5.1 Covariate shift due to condition variations

Covariate shift is the distribution shift of the independent variables, which are the inputs for MFD models. We investigate the influence of speed variations, object variations, and interference variations on covariate shift by measuring MMD between the original and target domains. The high-dimensional vibration data is transferred into the envelope spectrum for MMD calculation.

5.1.1 Speed variations

The M1 and M3 datasets are used in the experiment of speed variations. The data samples are grouped into several speed ranges, as shown in Table 13 in Annex I. The samples at speeds from 15 km/h to 25 km/h are defined as the source domain D_{s1}^{r}. Speed variation results in 8 target domains in M1, i.e. $D_{t1}^{r1}, D_{t1}^{r2}, \cdots, D_{t1}^{r8}$, which refers to the sets of data samples in the speed ranges of 25-35 km/h, 35-45 km/h, ⋯, 95-105 km/h. The covariate shift is quantified by MMD between the source domain D_{s}^{r} and the corresponding target domains D_{t1}^{rs}. Figure 10. illustrates the calculated MMDs for the dataset M1 and M3, respectively. Both diagrams show that the MMD of faulty data (in red) vastly increases with the enlarged speed differences. In contrast, the MMD of healthy data solely has minor changes under speed variations.

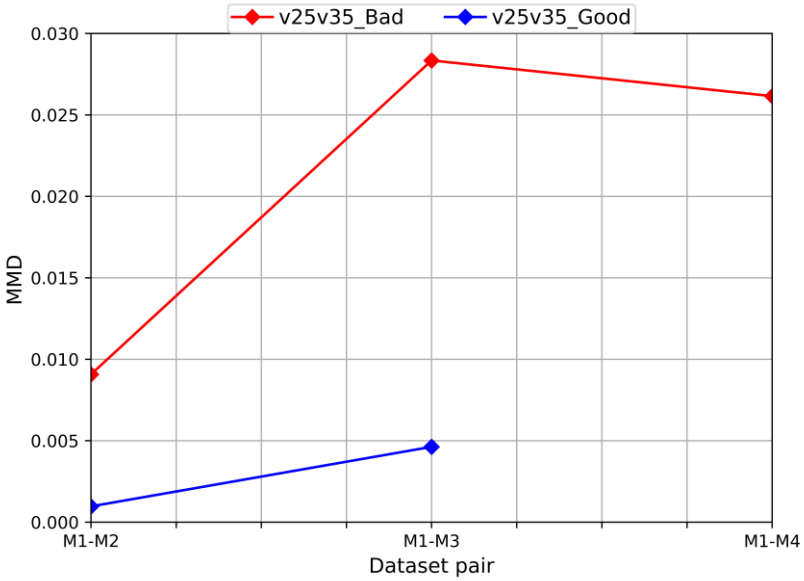


Figure 11. MMD between the source domain M1 and the target domains M2, M3 and M4 at speeds 25-35 km/h. "Good" refers to data under the healthy condition, while "Bad" refers to data with WFs. The health data is not available in M4.

5.1.2 Object variations

The covariate shift due to object variations is measured by MMD among different datasets. The data samples in the speed range between 25 and 35 km/h are used in this experiment, considering the data availability in all datasets (as shown in Table 13). M1 is defined as the source domain D_{s2}^{r}, while M2, M3 and M4 are the target domain D_{t2}^{r1}, D_{t2}^{r2} and D_{t2}^{r3}. Figure 11 shows the calculated MMDs between D_{s}^{r} and D_{t2}^{r1}, D_{t2}^{r2} as well as D_{t2}^{r3} respectively. At a first glance, the disparity between M1 and M2 are much lower than that between M1 and M3/M4. The possible reason may be that both M1 and M2 were measured on synthetic WFs and sampled at 5000 Hz. The measured wagons had the same bogie type, which may result in a similar vehicle dynamic behavior. In M3 and M4, the wagon bogies were different, WFs were naturally generated, and the data was sampled at 500 Hz. Furthermore, the magnitude of MMDs for faulty data is higher than that for healthy data, consistent with the above experiment. Comparing the magnitudes in Figure 11 with the ones in Figure 10, we find that the disparities caused by object variations are much lower than those caused by speed variations.

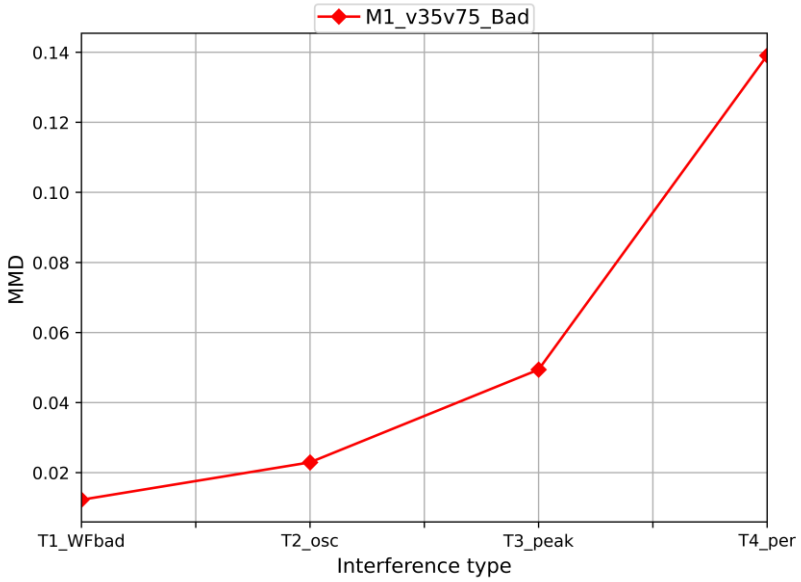


Figure 12. MMD caused by the variation of signal interferences. "T1_WFbad" refers to the MMD between D_s^{r1} and D_{t3}^{r1}. "T2_osc" refers to the MMD between D_s^{r2} and D_{t3}^{r2}. "T3_peak" refers to the MMD between D_s^{r2} and D_{t3}^{r3}. "T4_per" refers to the MMD between D_s^{r2} and D_{t3}^{r4}.

5.1.3 Interference variations

Four target domains are defined according to the four types of signal inferences as shown in Figure 8, i.e. WF signals under severe interferences as the target domain D_{t3}^{r1}, healthy signals with the oscillatory interference as the target domain D_{t3}^{r2}, healthy signals with one impulsive interference as the target domain D_{t3}^{r3}, and healthy signals with several adjacent impulsive interferences as the target domain D_{t3}^{r4}. As D_{t3}^{r1} concerns the faulty signals, the corresponding source domain D_{s3}^{r1} for MMD measure is the clean WF signal without severe signal inferences. The target domains D_{t3}^{r3}, D_{t3}^{r3}, and D_{t3}^{r3} concern the healty signals. Therefore, the corresponding source domain D_{s3}^{r2} for MMD measure is the clean healthy data. All data samples for the experiment are selected from M1 in the speed range between 35 and 75 km/h. Figure 12 shows the calculated MMDs caused by different signal interferences. Healthy signals with several adjacent impulsive interferences have the most considerable disparity from the ones without interferences. The corresponding MMD magnitude is even higher than the largest one caused by speed variations.

5.2 Robustness problems caused by condition variations

Model robustness is reflected by the accuracy impairment caused by condition variations. The baselines are trained and tested on the source domain to obtain the clean accuracy ACC_c^f. Afterward, the trained baseline models are tested on the target domains to obtain the impaired accuracies $ACC_{v,s}^f$, from which the robustness of a classifier mRB_v^f over severities can be derived.

In the robustness experiments, data samples are standardized and divided into 60% training datasets, 20% validation datasets and 20% test datasets, where the faulty and healthy data is kept balanced. Class imbalance can make a classifier learn just the primary class and result in

inaccurate classification results. Especially in MFD, the amount of faulty data is commonly much smaller than that of healthy data. As the class imbalance problem is an extensive research topic and worth making a dedicated study, we exclude this problem from our study. In the training procedure, we use mini-batch stochastic gradient descent with Adam optimization [46] with a batch size of 32 and 30 epochs. Each baseline is trained and tested over ten times. The extraordinary results are excluded, as they may originate from improper initialization of NN weights. The mean values and the standard deviations of WF detection accuracy are recorded and presented.

5.2.1 Speed variations

The robustness against speed variations is evaluated on the M1 dataset. The most critical scenario is to train a diagnostic model at high speeds and test at low speeds. The data samples at speeds from 55 km/h to 105 km/h are defined as the source domain D_{s1}^{r}, ensuring sufficient training data. The target domains $D_{t1}^{r1}, D_{t1}^{r2}, D_{t1}^{r3}$ and D_{t1}^{r4} refer to the sets of data samples in the speed ranges of 45-55 km/h, 35-45 km/h, 25-35 km/h and 15-25 km/h. Table 4 and Figure 13 present the test results. At first glance, three ResNet baselines have similar robustness against speed variations. When the models are trained and tested in the same speed range, they achieve very high hit rates (over 99%). However, their performance degrades significantly with the decrease in vehicle speeds, regardless of pre-processing methods. In comparison, GBDT with the handcrafted features has much better robustness.

Table 4. Diagnostic accuracies in different speed ranges and the mean robustness mRB^{f} of the baselines

Baseline	55-105 km/h	45-55 km/h	35-45 km/h	25-35 km/h	15-25 km/h	mRB_1
ResNet	99.85 ± 0.14%	98.42 ± 0.54%	96.88 ± 1.32%	80.95 ± 4.96%	58.00 ± 4.63%	0.84
CWT+ ResNet	99.83± 0.24%	96.60 ± 0.43%	96.79 ± 1.40%	82.47 ± 6.80%	56.10 ± 4.97%	0.83
HT+ ResNet	99.28 ± 1.28%	93.59 ± 0.64%	94.21 ± 0.63%	78.57 ± 3.50%	59.93 ± 2.05%	0.82
GBDT	99.41 ± 0.31%	95.92 ± 0.31%	96.35 ± 0.20%	92.57 ± 0.76%	88.34 ± 0.61%	0.94

5.2.2 Object variations

The robustness against object variations is evaluated in such a way that the baselines are trained on one dataset and tested on the others. All training and test data samples fall in the speed range between 15 and 45 km/h. The M1 dataset is defined as the source domain D_{s2}^{r}. The target domains D_{t2}^{r1}, D_{t2}^{r2} and D_{t2}^{r3} refer to the M2, M3 and M4 dataset, respectively. Although the above MMD experiment indicates that the domain disparities caused by object variations are much smaller than those caused by speed variations, the baselines' performance impairment does not follow this. As shown in Table 5 and Figure 14, all baselines have a similar performance on M1 and M2, however appreciably different on M3 and M4. In particular, ResNet fails to detect WFs on M3 and delivers the opposite detection results on M4. Pre-processing using CWT and HT helps overcome this issue, resulting in better mean robustness of CWT+ResNet, HT+ResNet and GBDT.

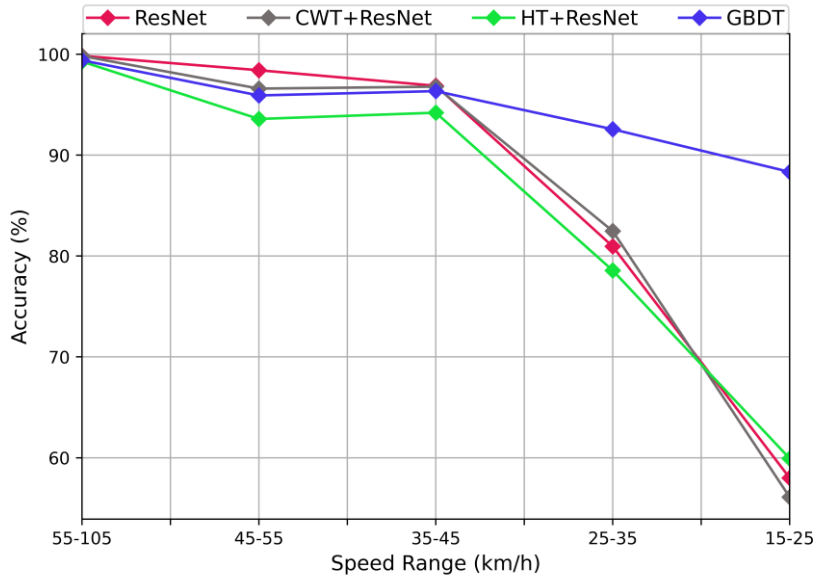


Figure 13. Diagnostic accuracies of the baselines tested in different speed ranges. Red line: ResNet; gray line: CWT+ResNet; green line: HT+ResNet; blue line: GBDT

Table 5. Diagnostic accuracies tested in different datasets and the mean robustness mRB^f of the baselines

Baseline	M1	M2	M3	M4	mRB_2
ResNet	99.01 ± 0.85%	81.06 ± 2.95%	56.89 ± 5.32%	27.04 ± 5.55%	0.56
CWT+ResNet	93.74 ± 4.04%	76.11 ± 3.24%	75.27 ± 8.09%	97.01 ± 4.61%	0.89
HT+ResNet	97.55 ± 1.23%	76.43 ± 1.52%	91.01 ± 1.72%	86.24 ± 7.60%	0.87
GBDT	95.72 ± 0.71%	76.21 ± 1.83%	96.66 ± 0.43%	61.72 ± 0.57%	0.82

5.2.3 Interference variations

The robustness against interference variations is evaluated by training the baselines on the clean dataset and testing the trained models on the corrupted datasets. The datasets for the source domains and the target domains are identical to those used in the MMD experiment. The experimental results are shown in Table 6. Diagnostic accuracies under different signal interferences and the mean robustness mRB^f of the baselines and Figure 15, where T0 refers to the clean datasets, T1 refers to the faulty signals distorted by significant interferences, T2 refers to the healthy signals with the oscillatory interferences, T3 refers to the healthy signals with the single impulsive interferences, and T4 refers to the healthy signals with the adjacent impulsive interferences. In general, the ResNet baselines have a similar performance against signal interferences. They are prone for miss detection than false alarms. Among them, ResNet without pre-processing has the best robustness. In contrast, GBDT based on handcrafted features is very sensitive to signal interferences. It fails to distinguish a signal interference from a WF signal.

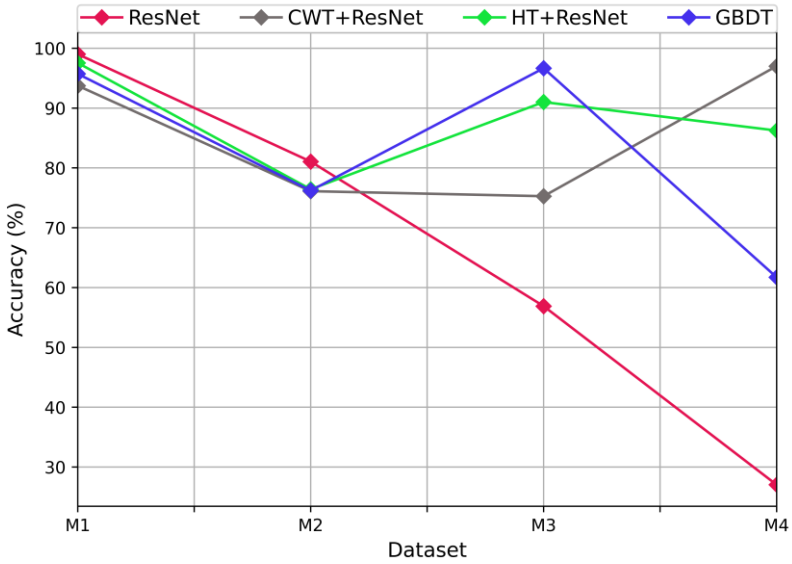


Figure 14. Diagnostic accuracies of the baselines tested in different datasets. Red line: ResNet; gray line: CWT+ResNet; green line: HT+ResNet; blue line: GBDT

Table 6. Diagnostic accuracies under different signal interferences and the mean robustness mRB^f of the baselines

Baseline	T0	T1	T2	T3	T4	mRB_3
ResNet	98.95 ± 1.71%	89.70 ± 4.62%	99.01 ± 2.16%	98.42 ± 4.10%	98.83 ± 2.91%	0.98
CWT+ ResNet	98.20 ± 1.07%	80.77 ± 8.66%	98.07 ± 3.78%	98.61 ± 2.43%	94.70 ± 7.07%	0.94
HT+ ResNet	99.06 ± 1.12%	69.40 ± 3.53%	98.29 ± 0.84%	99.42 ± 0.43%	96.28 ± 1.58%	0.92
GBDT	98.13 ± 0.71%	54.55 ± 2.94%	53.09 ± 2.69%	54.23 ± 2.39%	52.02 ± 1.77%	0.55

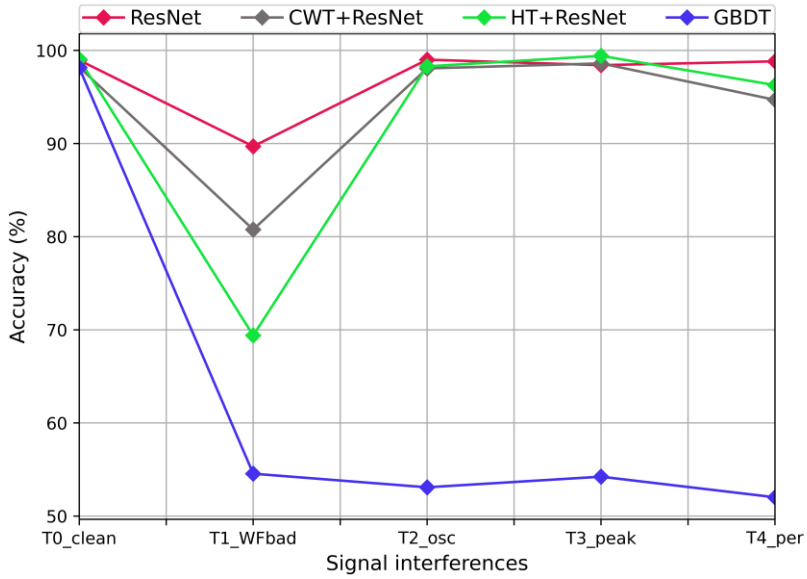


Figure 15. Diagnostic accuracies of the baselines tested under different signal interferences. Red line: ResNet; gray line: CWT+ResNet; green line: HT+ResNet; blue line: GBDT

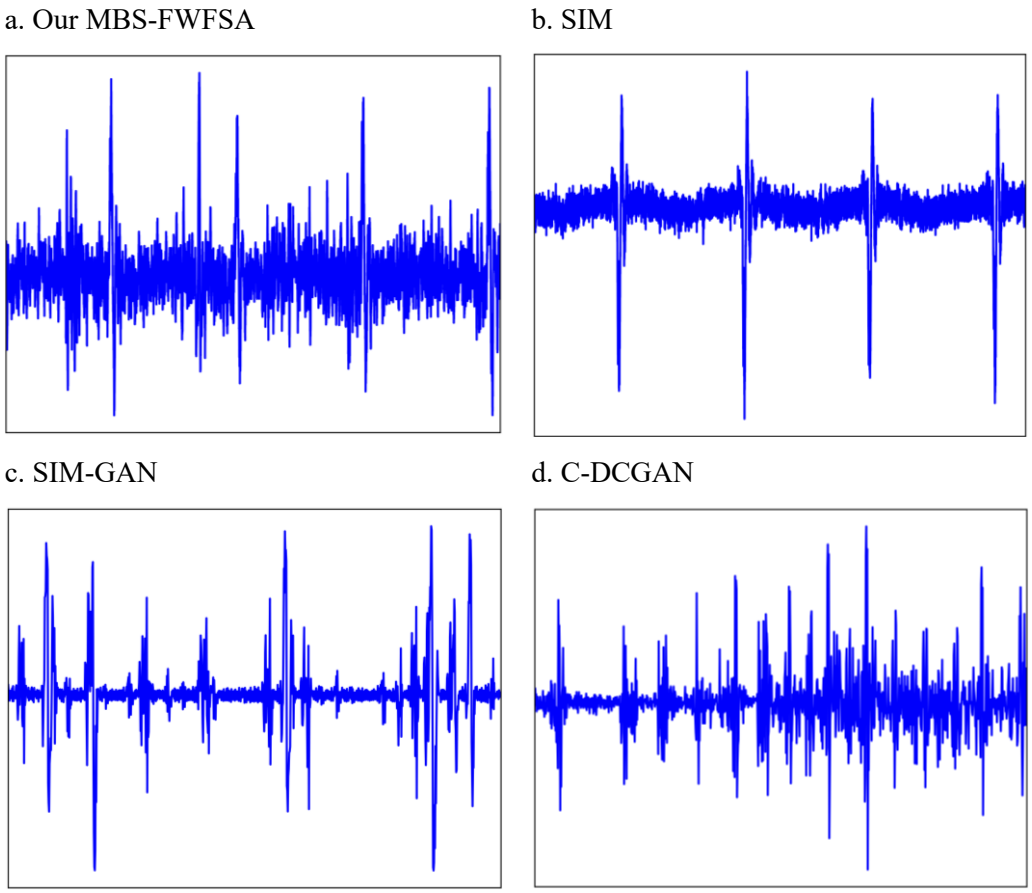


Figure 16. Synthetic samples generated by MBS-FWFSA, SIM, SIM-GAN and C-DCGAN, towards the low-speed target domain

5.3 Effectiveness of data augmentation

Data augmentation extends the diversity and thus the covariate distribution within the training dataset, preventing the out-of-distribution problem from the root. The proposed MBS-FWFSA framework produces infinitive RASF data as the additional synthetic source domains. In the following experiments, the effect of MBS-FWFSA on covariate shift and robustness enhancement is compared to three augmentation methods, i.e. C-DCGAN [33], SIM [37], and SIM-GAN [38].

5.3.1 Speed variations

The original source domain merely contains the vibration data at high speeds. Except C-DCGAN, our MBS-FWFSA, SIM and SIM-GAN can directly introduce low-speed simulation data for augmentation. The simulation data is generated by the MBS vehicle model with a 20 mm WF running at 20, 30 and 40 km/h on a straight line with track irregularities (as described by ERRI B176). Figure 16 illustrates the exemplary synthetic samples generated by MBS-FWFSA, SIM, SIM-GAN, and C-DCGAN, aiming to mimic the low-speed target domain. SIM generates the simulated faulty data with the injected noise. MBS-FWFSA augments the simulation data with the realistic interferences learned from the real healthy data. SIM-GAN takes both low-speed simulation data and the high-speed source domain data for training.

Although the generated synthetic data has an impulsive pattern, the peaks appear randomly without a periodic pattern. C-DCGAN generates the synthetic data solely from the high-speed source domain.

Table 7. Robustness of each MFD baseline averaged over speed ranges with and without data augmentation. Red figures indicate improved robustness. Green figures indicate decreased robustness.

MFD Baseline	Orig. mRB_1	mRB_1 after data augmentation			
		MBS-FWFSA	SIM	SIM-GAN	C-DCGAN
ResNet	0.84	0.90	0.89	0.86	0.85
CWT+ResNet	0.83	0.90	0.85	0.81	0.83
HT+ResNet	0.82	0.87	0.74	0.81	0.82
GBDT	0.94	0.92	0.92	0.92	0.91

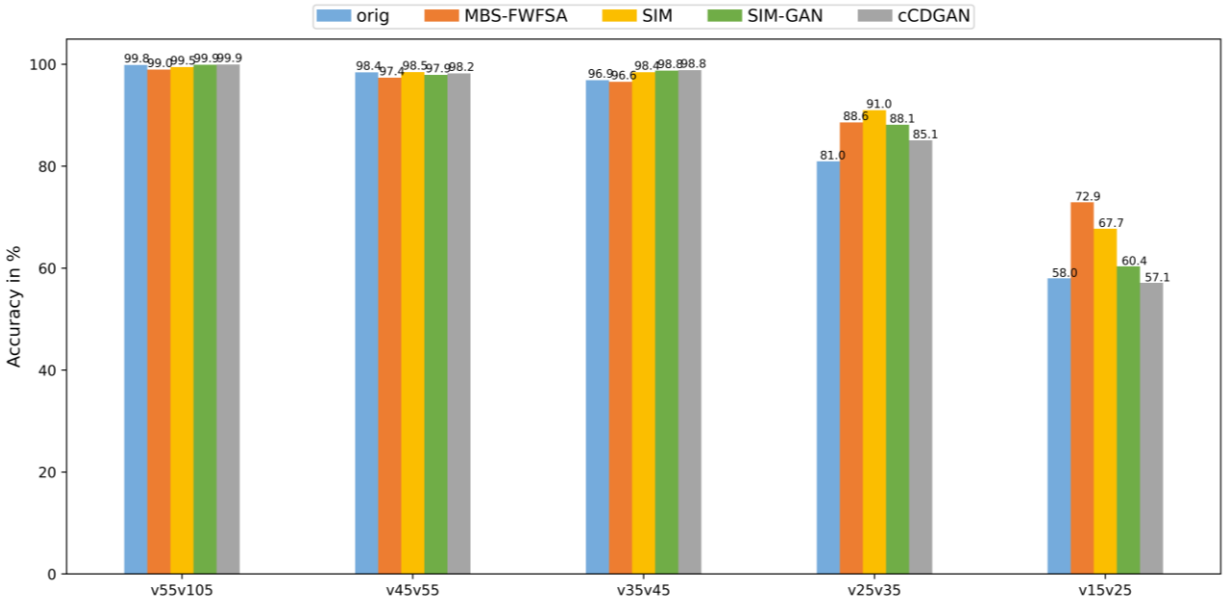


Figure 17. Effect of data augmentation on ResNet in the case of speed variations. “orig” denotes the test accuracy of ResNet on the original source domain. “MBS-FWFSA”, “SIM”, “SIM-GAN” and “C-CDGAN” denote the test accuracy of ResNet on the dataset augmented by the corresponding data augmentation methods, where MBS-FWFSA is our proposed approach.

The baseline MFD methods are trained on the augmented dataset consisting of the original and synthetic source domains to evaluate the effect of data augmentation. The amount of synthetic data samples is kept balanced with the original ones. The results are summarized in Table 7. As the synthetic data is generated in the time domain, all the data augmentation methods can improve the robustness of ResNet. However, they may have no effect or even a negative effect on the envelope and wavelet spectrum. The WF detection accuracies of ResNet on each speed range are further illustrated in Figure 17. At first glance, the more condition varies, the better the data augmentation works. In the lowest speed range, the proposed MBS-FWFSA outperforms the other data augmentation methods and improves the

detection accuracy from 58.9% to 72.9%. SIM also achieves a considerable improvement, whereas SIM-GAN and C-CDGAN have negligible effects. In the speed range between 25 and 35 km/h, all data augmentation methods improve the robustness, and SIM has a relatively better performance. The reason is that MBS-FWFSA and SIM controllably introduce an adequate amount of low-speed vibration signals into the source domain. However, the synthetic samples generated by SIM-GAN and C-CDGAN are randomly distributed and thus insufficient to force Resnet to learn the low-speed patterns. In addition, the WF patterns are prone to be distorted by impulsive interferences in the lowest speed range, which can be simulated by MBS-FWFSA and included in the synthetic source domain. Therefore, MBS-FWFSA overperforms SIM in the speed range between 15 and 25 km/h.

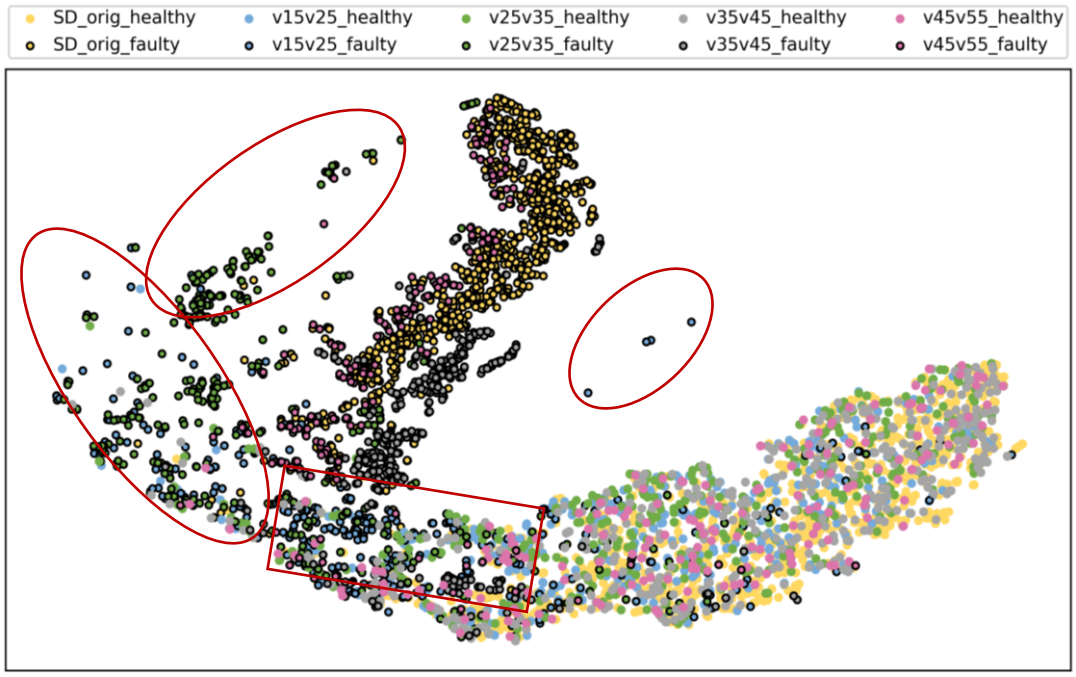
For a more in-depth analysis, we visualize the covariate distribution of the original and synthetic source domain as well as the target domains in Figure 18. The covariate distribution maps are 2D embedded space converted by t-distributed Stochastic Neighbor Embedding (t-SNE) from the features learned by ResNet in the corresponding domains. To be more specific, 16 features (outputted by the global average pooling layer) are learned by ResNet from the original 10240 data points per sample. They are further embedded by t-SNE in a 2D space. Figure 18 (a) shows the distribution of the original source domain (in yellow) and the target domains. The healthy data points (markers without the black edge), regardless of vehicle speeds, are aggregated well in one area. This observation is consistent with the MMD measure in Figure 10. Regarding the faulty data points (markers with the black edge), the source domain covers or is close to the target domain 35-45 km/h (in gray) and 45-55 km/h (in pink). Therefore, the WF detection accuracy in these two domains remains very high. However, the low-speed target domain 25-35 km/h (in green) and 15-25 km/h (in blue) are far from the original source domain. This leads to performance impairment. They are mainly distributed in four areas. Three of them (highlighted by three red circles) are covered by the synthetic source domains generated by MBS-FWFSA (in red), as shown in Figure 18 (b). The area highlighted by a red rectangle is very close to the healthy data and not covered by any source domain. It mainly consists of the blue points at 15-25 km/h. Therefore, the accuracy in this target domain can only be improved to 72.9% by MBS-FWFSA. To compare different augmentation methods, we visualize the faulty data distribution at 15-25 km/h in the target domain and different synthetic source domains in Figure 19. Obviously, the red points generated by MBS-FWFSA are the closest to the target domain in blue. As MBS-FWFSA stems from SIM, the red points have a small overlap with the yellow points generated by SIM. The points generated by the two GAN-based methods are close to each other.

5.3.2 Object variations

The original source domain merely contains the M1 dataset at speeds between 15 and 45 km/h. To simulate object variation, we change some parameters of the vehicle and WF models, such as load conditions, suspension parameters, WF sizes and WF distributions. Figure 20 shows the exemplary synthetic samples generated by MBS-FWFSA, SIM, SIM-GAN and C-DCGAN, simulating two adjacent 30mm WFs at 20 km/h. Comparing the signal generated by SIM in Figure 20 with that in Figure 16, we readily observe the difference between the single-WF signal and the two-WF signal. MBS-FWFSA adds additional background information to the SIM data. The SIM-GAN signals stem from the SIM data and

the original low-speed M1 data. C-DCGAN generates the synthetic data solely from the low-speed M1 data.

(a)



(b)

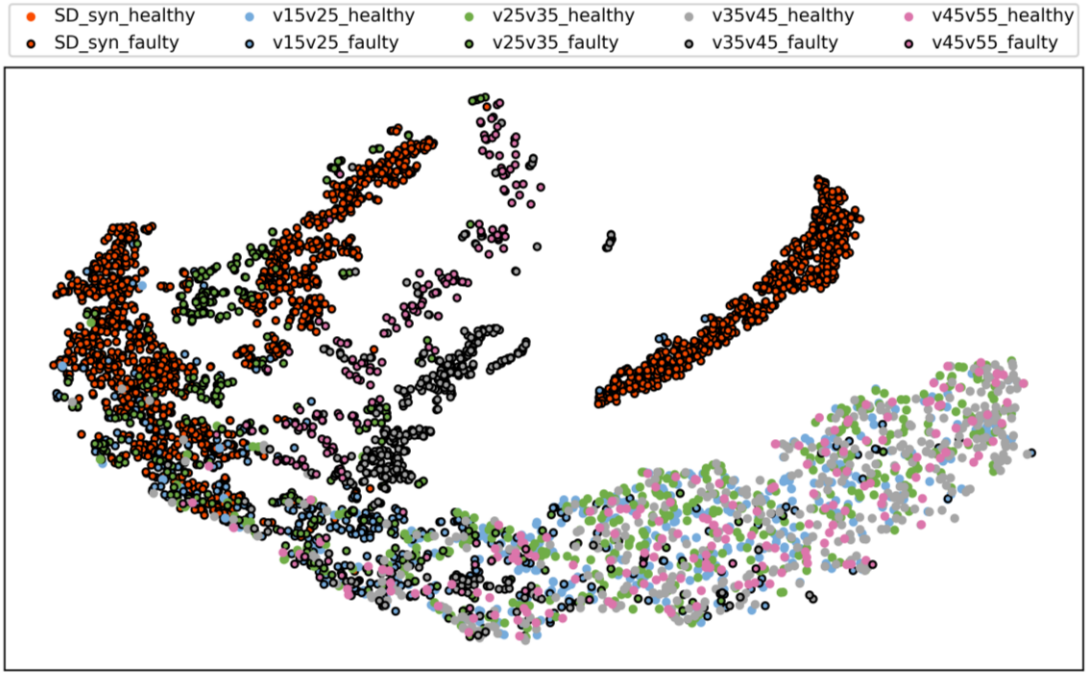


Figure 18. Covariate distribution in the case of speed variations (a) between the original source domain and the target domains; (b) between the synthetic source domain and the target domains. “SD” in the legend is the abbreviation of “source domain”. The yellow markers are the original SD. The red ones are the synthetic SD generated by “MBS-FWFSA”. The blue ones are the target domain for the speed range between 15 and 25 km/h. The green ones are for 25-35 km/h. The gray ones are for 35-45 km/h. The pink ones are for 45-55 km/h. The healthy signals are the markers without the edge. The faulty signals are with the black edge.

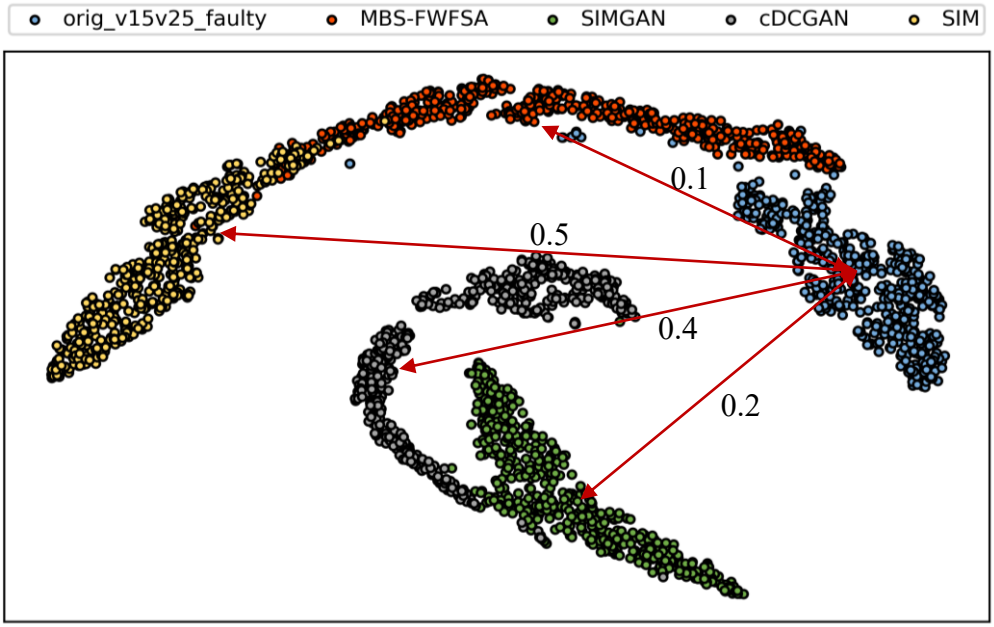


Figure 19. Covariate distribution of the target domain 15-25 km/h (in blue) and the different synthetic domains generated by MBS-FWFSA (in red), SIMGAN (in green), cDCGAN (gray) and SIM (in yellow). The displayed distances are the calculated MMDs between the target domain and the corresponding synthetic source domains.

Table 8 summarizes the effect of different data augmentation methods on robustness enhancement for the MFD baselines under object variations. ResNet's robustness is significantly improved by all data augmentation methods. Figure 21 shows the detailed detection accuracy of ResNet in the target domains. The proposed MBS-FWFSA outperforms the others in all the target domains. In particular, it improves the accuracy from 27% to 74.5% on the M4 dataset due to the reality-augmented simulation of the two-WF signals. Moreover, the detection accuracy on the M3 dataset is improved from 56.9% to 79.6% by MBS-FWFSA, much more than that by the others.

Table 8. Robustness of each MFD baseline averaged over datasets with and without data augmentation. Red figures indicate improved robustness. Green figures indicate decreased robustness.

MFD Baseline	**Orig. mRB_1**	**mRB_2 after data augmentation**			
		MBS-FWFSA	SIM	SIM-GAN	C-DCGAN
ResNet	0.56	0.82	0.70	0.74	0.67
CWT+ResNet	0.89	0.88	0.94	0.87	0.89
HT+ResNet	0.87	0.90	0.89	0.93	0.91
GBDT	0.82	0.85	0.85	0.84	0.84

Figure 22 (a) displays the distribution of the original source domain M1 and the target domains. The target domain M2 is highly overlapped with the original source domain M1, in terms of both healthy and faulty data. Therefore, the diagnostic performance in the target

domain M2 remains 81.1% without data augmentation. The green M3 faulty data points are widely distributed, mainly in area 3 and partly in area 1 and 2, where only a few M1 points fall. This explains the accuracy of 56.9% in the target domain M3. In M4 (gray), only the faulty data is available, which is distributed in area 1 and 2. A few M1 faulty data points are close to or within area 1, whereas area 2 is nearly not covered. In contrast, we observe some M1 healthy data points in area 2. This may be the reason why ResNet trained in M1 delivers the opposite detection results in the target domain M4. In Figure 22 (b), the synthetic healthy data generated by MBS-FWFSA are very close to area 4, covering a part of the M3 healthy data. This reduces the possibility of false alarms in the target domain M3. The synthetic faulty data points mainly lie around area 2, covering a part of the M3 and M4 faulty data points. However, areas 1 and 3 are not covered by the synthetic data. This limits the robustness enhancement. Figure 23 compares the MMD between the target domain M4 and the synthetic source domains generated by the four data augmentation. We observe two aggregations of the blue M4 points. The primary aggregation lies on the left, close to the gray C-DCGAN points. The secondary M4 aggregation lies in the middle, covered by the red MBS-FWFSA points. In terms of MMD, C-DCGAN is closer to the target domain M4 than MBS-FWFSA. This result is not consistent with that shown in Figure 21, where C-DCGAN is the most ineffective for robustness enhancement of ResNet in the target domain M4.

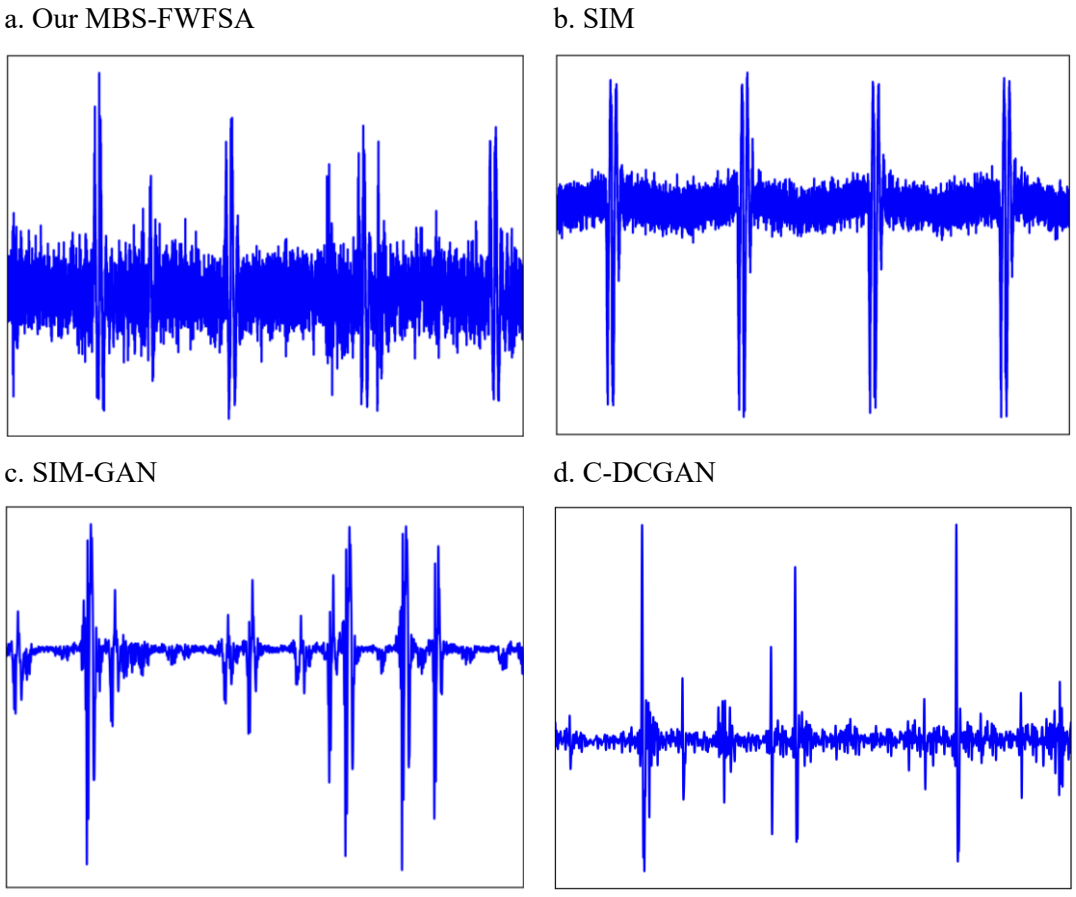


Figure 20. Synthetic samples generated by MBS-FWFSA, SIM, SIM-GAN and C-DCGAN, simulating the variation of the wagon and WF conditions

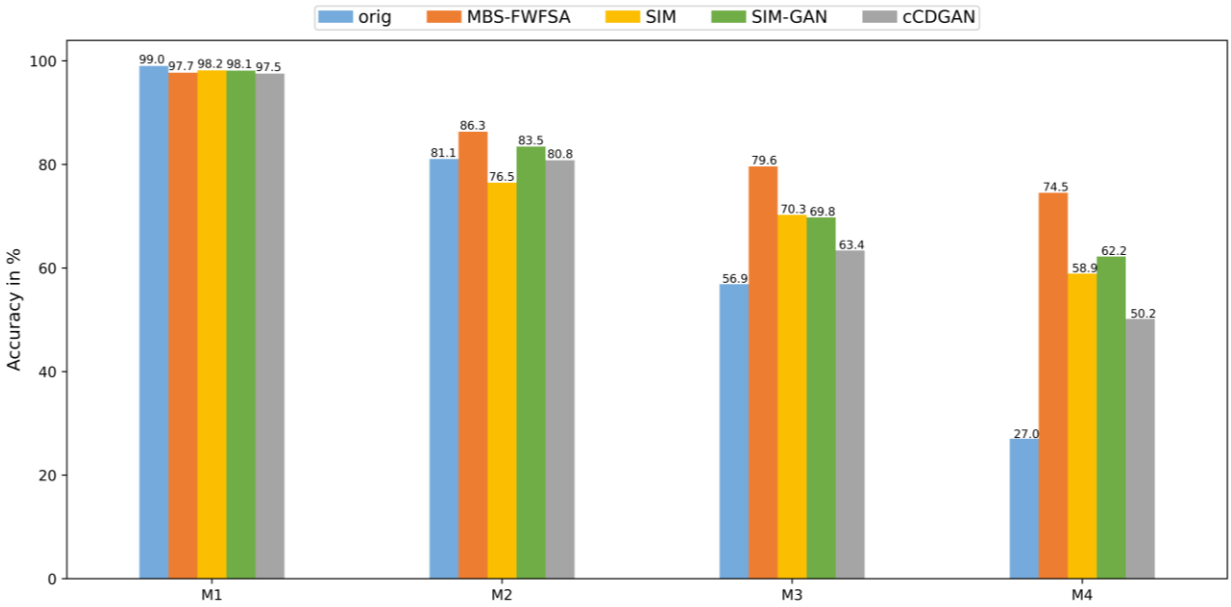


Figure 21. Effect of data augmentation on ResNet in the case of object variations

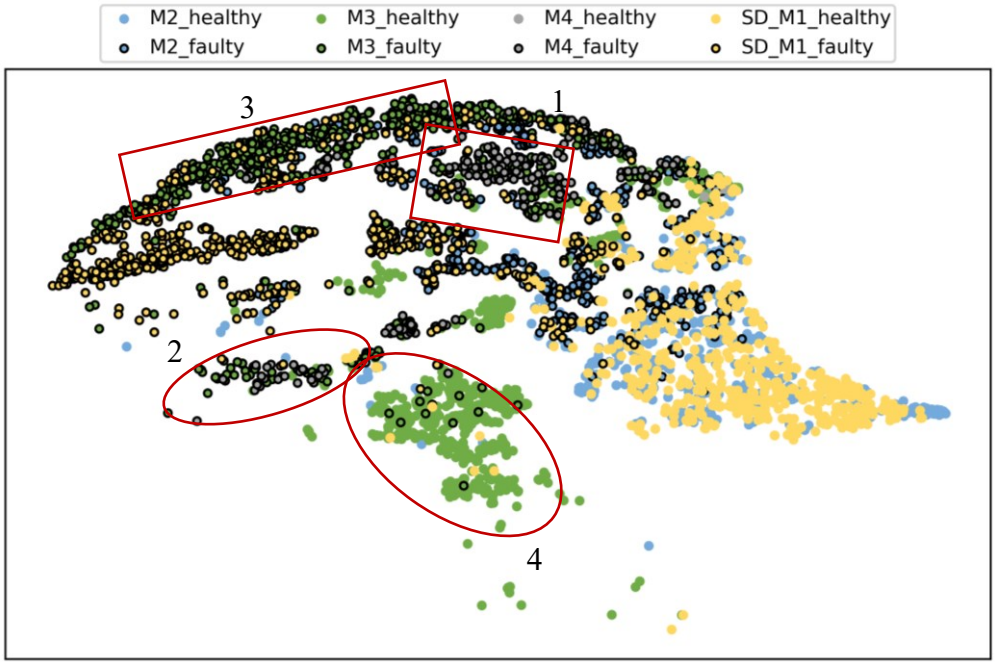
5.3.3 Interference variations

The original source domain solely contains the clean faulty and healthy signals in the speed range between 35 and 75 km/h from the M1 dataset. The data augmentation setting is similar to that in the speed-variation experiment. The only difference is that the vehicle speeds in the simulations are set as 40, 55 and 70 km/h. The generated synthetic data samples are similar to those shown in Figure 16. Table 9 summarizes the effect of different data augmentation methods on robustness enhancement for the MFD baselines under interference variations. Three ResNet-based baselines have already high robustness against signal interferences, which cannot be further improved by data augmentation. GBDT has the lowest robustness. However, all the data augmentation methods unexpectedly have nearly no effect on GBDT.

The distribution of GBDT features is visualized in Figure 24 (a), where 21 statistic features per sample are embedded by t-SNE. Several clusters of the original and synthetic source domains are clearly observed, representing faulty and healthy data at different vehicle speeds. In contrast, the four target domains are widely distributed. The healthy data points are partly mixed with the faulty ones from the source domain. A number of the blue T1 faulty data points hide among the healthy ones from other target domains, making them indistinguishable. This observation explains why the WF detection by GBDT fails in the target domains, and cannot be improved by data augmentation. For comparison, we visualize the distribution of HT+ResNet features in Figure 24 (b). In this case, 16 features (outputted by the global average pooling layer) are learned by HT+ResNet from the original 256 data points per sample. They are further embedded by t-SNE in a 2D space. We readily observe that the healthy data points from the target domains T2, T3 and T4 are aggregated in one area and fully covered by the original and synthetic source domains. This reflects the high diagnostic accuracy of HT+ResNet in these three target domains. Only a tiny part of the blue T1 points are mixed with the yellow source domain points. This leads to an accuracy of 69.40% without data augmentation. By adding the red synthetic points, we improve the HT+ResNet accuracy

in the target domain T1 from 69.40% to 76.91%. The rest blue points are buried in the healthy ones and thus can hardly be detected.

(a)



(b)

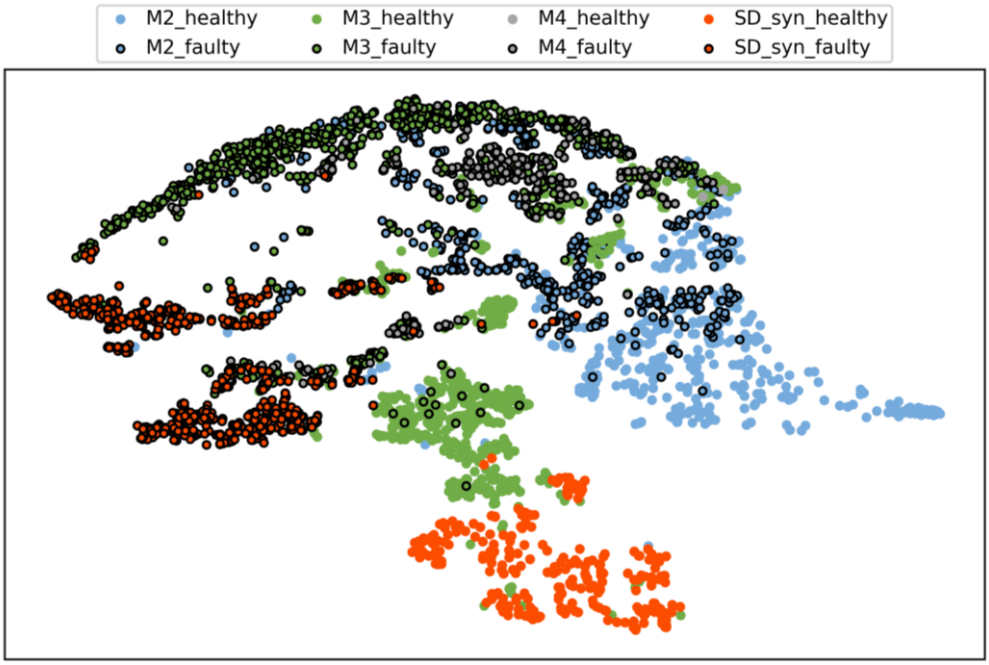


Figure 22. Covariate distribution in the case of object variations (a) between the original source domain and the target domains; (b) between the synthetic source domain and the target domains. The yellow markers are the original SD. The red ones are the synthetic SD generated by "MBS-FWFSA". The blue ones are the M2 target domain. The green ones are M3. The gray ones are M4. The healthy signals are the markers without the edge. The faulty signals are with the black edge.

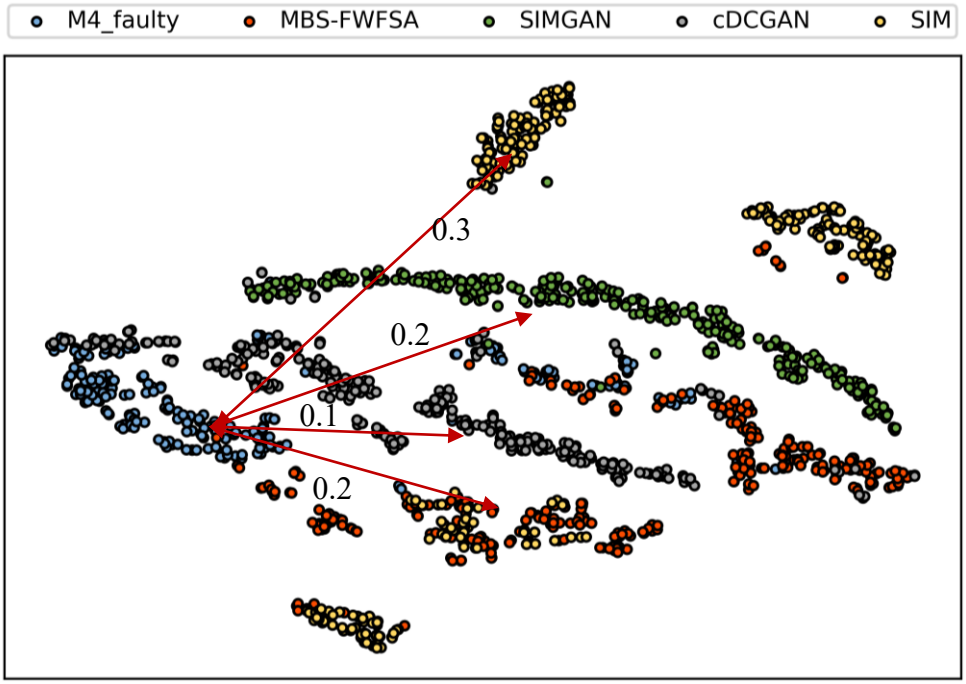


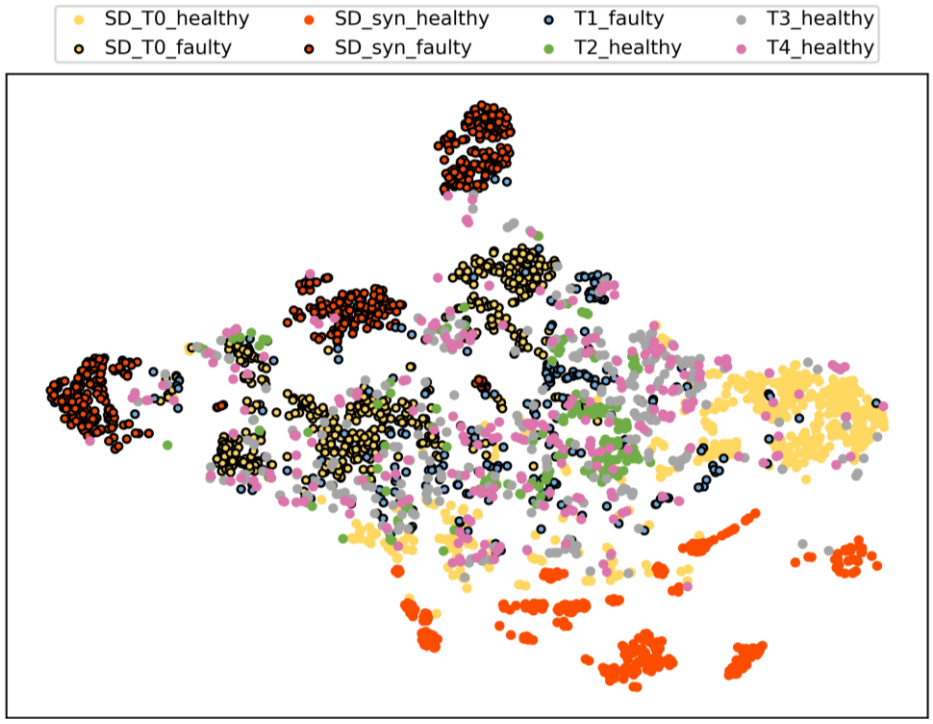
Figure 23. Covariate distribution of the target domain M4 (in blue) and the different synthetic domains generated by MBS-FWFSA (in red), SIMGAN (in green), cDCGAN (gray) and SIM (in yellow). The displayed distances are the calculated MMDs between the target domain and the corresponding synthetic source domains.

The MMD distances between the blue target domain T1 and the four synthetic source domains are compared in Figure 25. SIMGAN is the closest to T1 and achieves the highest accuracy of 81.24%. The second one is MBS-FWFSA. There are three clusters of the MBS-FWFSA red points in Figure 25, representing three simulation speeds. One of them (at 55 km/h) is within the T1 cluster and closer than SIMGAN, although the averaged MMD of MBS-FWFSA is larger than SIMGAN.

6. Discussions

In the above experiments, the consequences of condition variations on covariate shift and model robustness have been empirically studied at first. Comparing the observations in Section 5.1 with the visualization of the exemplary signals in Figure 6-8, we conclude that the covariate shift measured by MMD is consistent with the visual observation of the vibration signals. The speed variations and severe signal interferences establish distinctions of the signal manifestation, resulting in a significant covariate shift between the source and target domain, which can be visually recognized. In comparison, the signal disparities induced by object variations are much smaller, and the faulty signals from different datasets look similar. In an MFD task, the distance between the source and target domain within one class belongs to the intra-class distance, while the distance between the healthy and faulty data refers to the inter-class distance. MFD models may not work correctly when the intra-class distance is larger than the inter-class distance due to covariate shift. Figure 26 compares the inter-class distance and the maximum intra-class distances caused by three types of condition variations. Based on this, speed variations and interference variations were expected to result in more severe diagnostic performance impairment.

(a)



(b)

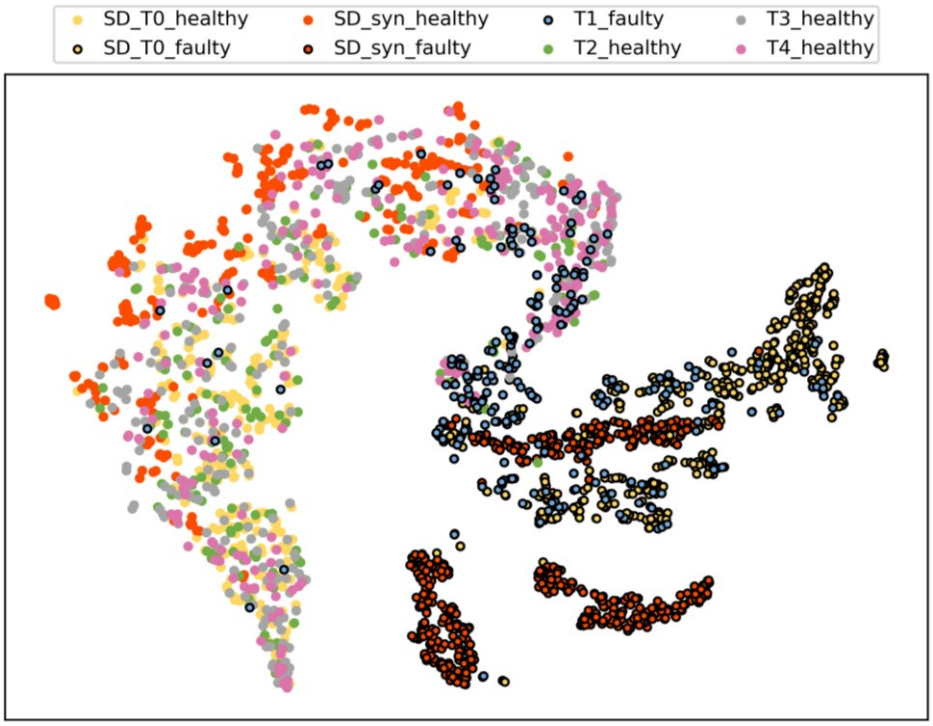


Figure 24. Covariate distribution in the case of interference variations (a) 21 statistic features; (b) envelope features learned by HT+ResNet. The yellow markers are the original SD. The red ones are the synthetic SD generated by "MBS-FWFSA". The blue ones are the T1 target domain. The green ones are T2. The gray ones are T3. The pink ones are T4. The healthy signals are the markers without the edge. The faulty signals are with the black edge.

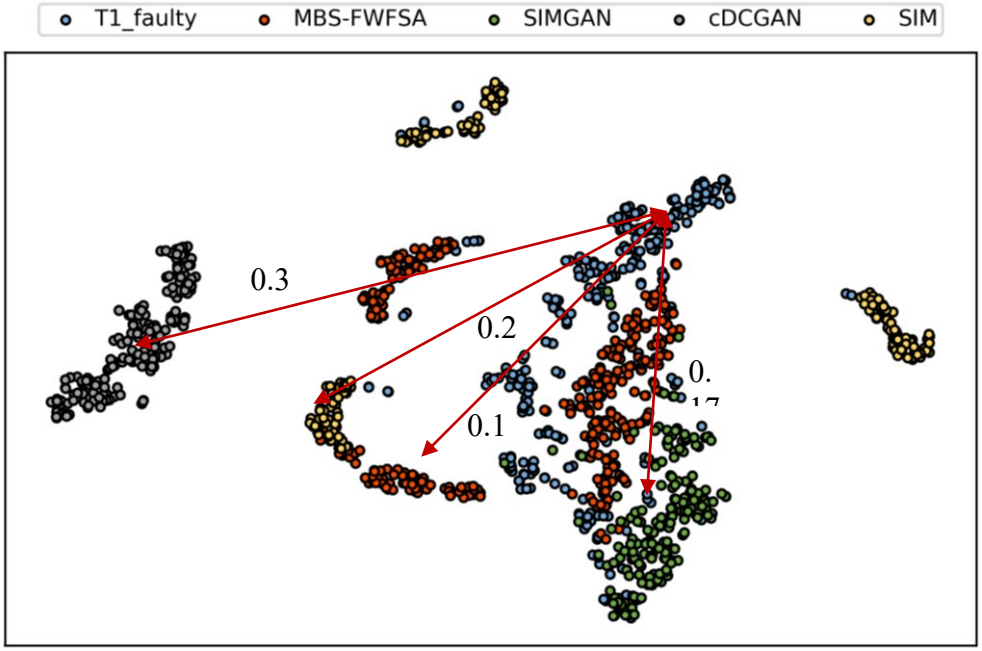


Figure 25. Covariate distribution of the target domain T1 (in blue) and the different synthetic domains generated by MBS-FWFSA (in red), SIMGAN (in green), C-DCGAN (gray) and SIM (in yellow). The displayed distances are the calculated MMDs between the target domain and the corresponding synthetic source domains.

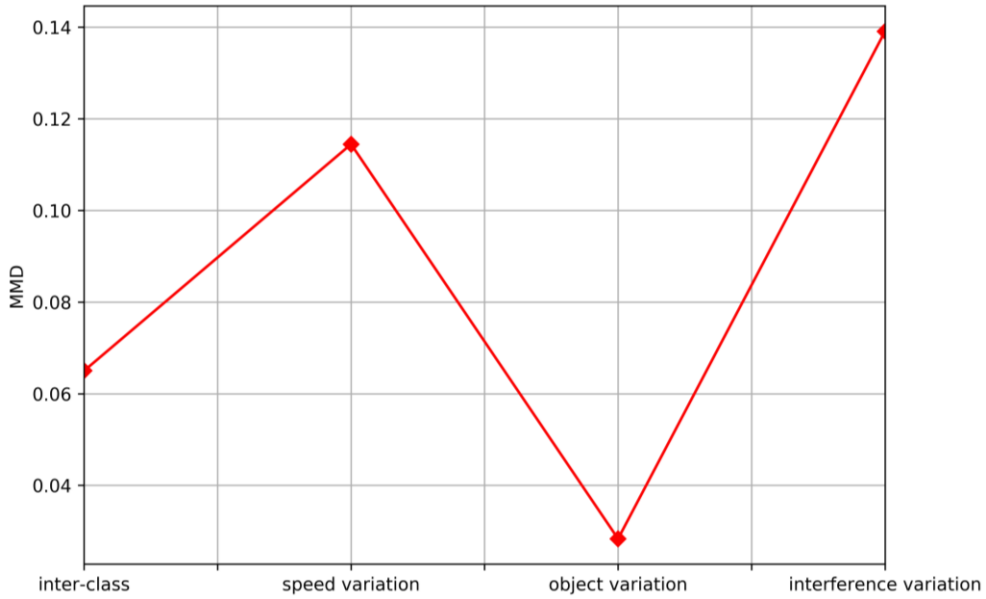


Figure 26. Comparison between the inter-class distance and the intra-class distances caused by three types of condition variations in M1

However, this is disproved by the mean robustness of the four MFD baselines presented in Table 4-6. The performance impairment does not directly correspond to the covariate shift measured by MMD. The minor disparities in vibration signals due to object variations may lead to the complete malfunction of diagnostic models. A comparable case is an adversarial example where an imperceptible perturbation in the data results in an incorrect prediction of a DL model. In contrast, the visible signal interferences have little influence on a DL model, even though they result in a significant covariate shift and domain disparity.

Table 9. Robustness of each MFD baseline averaged over interference types with and without data augmentation. Red figures indicate improved robustness. Green figures indicate decreased robustness.

MFD Baseline	Orig. mRB_1	mRB_1 after data augmentation			
		MBS-FWFSA	SIM	SIM-GAN	C-DCGAN
ResNet	0.98	0.98	0.98	0.90	0.98
CWT+ResNet	0.95	0.95	0.91	0.94	0.84
HT+ResNet	0.92	0.93	0.92	0.90	0.92
GBDT	0.55	0.57	0.53	0.53	0.58

Another noticeable issue arising in the robustness experiments in Section 5.2 is that an MFD method may have very different robustness levels against different types of condition variations. For instance, GBDT with the statistic features is robust against speed variations but fragile due to signal interferences. Therefore, the mean robustness averaged over different variations plays an essential role in evaluating a diagnosis model. Table 10 summarizes the mean robustness of each baseline. CWT+ResNet has achieved the best performance.

Table 10. Mean robustness of each MFD baseline averaged over three condition variation types

Baseline	mRB_1	mRB_2	mRB_3	mRB
ResNet	0.84	0.56	0.98	0.79
CWT+ResNet	0.83	0.89	0.94	0.89
HT+ResNet	0.82	0.87	0.92	0.87
GBDT	0.94	0.82	0.55	0.77

The effect of data augmentation on the covariate distribution and robustness enhancement have been investigated under three types of condition variations for four MFD baselines. In addition, the proposed MBS-FWFSA was compared to three augmentation methods. The detailed analysis in Section 5.3 indicates that the synthetic data generated by an effective data augmentation strategy extends the source domain distribution. When the distribution of the combined original and synthetic source domain can cover or get closer to the target domains, the model robustness will be improved. The only exception occurs in object variations. C-DCGAN has achieved the lowest accuracy improvement, but its distribution is the closest to the target domain. Table 15 summarizes the final comparison results. The mean robustness is averaged over three condition variation types and the corresponding severities. The proposed MBS-FWFSA achieves the robustness enhancement for all the MFD baselines and overperforms other data augmentation methods.

Table 11. Effect of the four data augmentation strategies on robustness enhancement averaged over different condition variation types and severities. Red figures indicate improved robustness. Green figures indicate decreased robustness.

MFD Baseline	Orig. mRB	mRB after data augmentation			
		MBS-FWFSA	SIM	SIM-GAN	C-DCGAN
ResNet	0.79	0.90	0.86	0.83	0.83
CWT+ResNet	0.89	0.91	0.90	0.87	0.85
HT+ResNet	0.87	0.90	0.85	0.88	0.88
GBDT	0.77	0.78	0.77	0.80	0.78

The computation complexity of the four methods is hardly fairly compared. Instead, Table 12 lists the required laborious procedures of each data augmentation methods. C-DCGAN requires adequate real faulty data under various operating conditions to ensure the quality of the generated synthetic data. MBS-FWFSA, SIM and SIM-GAN rely on the simulations to create the synthetic faulty data. In terms of data processing, the most time-consuming step of MBS-FWFSA is feature extraction. As the algorithm FWFSA in Table 1 indicates, feature extraction is applied for the updated average time series in each updating iteration and each sample in the source domain D. The signal processing techniques used in this work is HT and CWT, both of which has the computational complexity of $O(N)$. The total complexity of FWFSA is $O(I^2N)$, much faster than the original WDBA $O(I^2N^2)$, where I is the total number of time series samples within the source domain D and N is the dimension of time series. FWFSA leans the healthy vibration manifestation in the railway environment. The procedure of FWFSA has merely to be executed for one time in the entire experiments. Once the learning process is done, the RASF data under various operating conditions can be generated by simply executing eq. (10). In the GAN-based methods, the procedures for hyperparameter tuning are incredibly laborious. For each operating condition, the training process of GAN has to be repeated, which is also time-consuming. In addition, GANs have a much higher requirement on the training data availability than FWFSA. The latter can work on an arbitrary amount of data.

7. Conclusions

We proposed a novel data augmentation framework MBS-FWFSA. MBS simulates the faulty and operating conditions, which can hardly be measured in practice. FWFSA has been developed by us as a new time series averaging method. It learns the realistic background information in the regular operating environment. Combining FWFSA and MBS can generate reality-augmented simulation faulty data. It extends the training data distribution and improves the robustness of ML-based fault diagnostic methods under various condition variations. The proposed FWFSA-MBS has been validated by extensive experiments based on the real-world measurement data in a railway application (i.e. wheel flat detection) and overperforms the state-of-the-art data augmentation methods.

Table 12. Required laborious procedures of each data augmentation methods.

Laborious procedure	MBS-FWFSA	SIM	SIM-GAN	C-DCGAN
Modeling and simulation	√	√	√	
Collection of adequate real faulty data			√	√
Feature extraction	√			
Hyperparameter tuning			√	√

Moreover, the experiments reveal that condition variations may result in either a significant or imperceptible covariate shift. The former can be readily observed and quantified by a common discrepancy measure such as maximum mean discrepancy. The latter caused by object variations cannot be reflected by maximum mean discrepancy. It is more similar to an adversarial attack. Both significant and imperceptible covariate shifts can result in severe performance impairment. Since the diagnostic methods behave differently in terms of robustness under different condition variations, we recommend using the proposed mean robustness for a fair evaluation.

Acknowledgments

The experiment data used in this paper is supported by the previous projects of Chair of Rail Vehicles TUB. The research is funded by the EU Shift2Rail project Assets4Rail (Grand number: 826250) under Horizon 2020 Framework Programme.

References

[1] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, A.K. Nandi, Applications of machine learning to machine fault diagnosis: A review and roadmap, Mechanical Systems and Signal Processing. 138 (2020) 106587. doi:10.1016/j.ymssp.2019.106587.

[2] M.C. Nakhaee, D. Hiemstra, M. Stoelinga, M.V. Noort, The Recent Applications of Machine Learning in Rail Track Maintenance: A Survey, Reliability, Safety, and Security of Railway Systems. Modelling, Analysis, Verification, and Certification Lecture Notes in Computer Science. (2019) 91–105. doi:10.1007/978-3-030-18744-6_6.

[3] G.D. Dharani, N.G. Nair, P. Satpathy, J. Christopher, Covariate Shift: A Review and Analysis on Classifiers, 2019 Global Conference for Advancement in Technology (GCAT). (2019). doi:10.1109/gcat47503.2019.8978471.

[4] S. Ma, B. Cheng, Z. Shang, G. Liu, Scattering transform and LSPTSVM based fault diagnosis of rotating machinery, Mechanical Systems and Signal Processing. 104 (2018) 155–170. doi:10.1016/j.ymssp.2017.10.026.

[5] Y. Hu, X. Tu, F. Li, High-order synchrosqueezing wavelet transform and application to planetary gearbox fault diagnosis, Mechanical Systems and Signal Processing. 131 (2019) 126–151. doi:10.1016/j.ymssp.2019.05.050.

[6] Z. Zilong, Q. Wei, Intelligent fault diagnosis of rolling bearing using one-dimensional multiscale deep convolutional neural network based health state classification, 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC). (2018). doi:10.1109/icnsc.2018.8361296.

[7] H. Zhu, X. Wang, Y. Zhao, Y. Li, W. Wang, L. Li, Sparse representation based on adaptive multiscale features for robust machinery fault diagnosis, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science. 229 (2014) 2303–2313. doi:10.1177/0954406214557343.

[8] G. Xu, M. Liu, Z. Jiang, D. Söffker, W. Shen, Bearing Fault Diagnosis Method Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning, Sensors. 19 (2019) 1088. doi:10.3390/s19051088.

[9] A. Yunusa-Kaltungo and J. K. Sinha, Sensitivity analysis of higher order coherent spectra in machine faults diagnosis, Structural Health Monitoring: An International Journal, 15(2016) 555–567. doi: 10.1177/1475921716651394.

[10] Y. Li, N. Wang, J. Shi, J. Liu, X. Hou, Revisiting Batch Normalization for Practical Domain Adaptation, 2017 International Conference on Learning Representations (ICLR). (2017).

[11] T.-W. Ke, M. Maire, S.X. Yu, Multigrid Neural Architectures, ArXiv.org. (2017). https://arxiv.org/abs/1611.07661 (accessed August 30, 2020).

[12] G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, K.Q. Weinberger, Multi-Scale Dense Networks for Resource Efficient Image Classification, ArXiv.org. (2018). https://arxiv.org/abs/1703.09844 (accessed August 30, 2020).

[13] A. Farahani, S. Voghoei, K. Rasheed, H.R. Arabnia, A Brief Review of Domain Adaptation, ArXiv.org. (2018). https://arxiv.org/abs/1703.09844 (accessed January 23, 2021)

[14] W. Qian, S. Li, J. Wang, A New Transfer Learning Method and its Application on Rotating Machine Fault Diagnosis Under Variant Working Conditions, IEEE Access. 6 (2018) 69907–69917. doi:10.1109/access.2018.2880770.

[15] Z. Chai, C. Zhao, A Fine-Grained Adversarial Network Method for Cross-Domain Industrial Fault Diagnosis, IEEE Trans. Automat. Sci. Eng. 17 (2020) 1432–1442. https://doi.org/10.1109/TASE.2019.2957232.

[16] X. Li, W. Zhang, H. Ma, Z. Luo, X. Li, Domain generalization in rotating machinery fault diagnostics using deep neural networks, Neurocomputing 403 (2020) 409–420. https://doi.org/10.1016/j.neucom.2020.05.014.

[17] Y. Liao, R. Huang, J. Li, Z. Chen, W. Li, Deep Semi-supervised Domain Generalization Network for Rotary Machinery Fault Diagnosis under Variable Speed, IEEE Trans. Instrum. Meas. (2020) 1. https://doi.org/10.1109/TIM.2020.2992829.

[18] H. Zheng, R. Wang, Y. Yang, Y. Li, M. Xu, Intelligent Fault Identification Based on Multisource Domain Generalization Towards Actual Diagnosis Scenario, IEEE Trans. Ind. Electron. 67 (2020) 1293–1304. https://doi.org/10.1109/TIE.2019.2898619.

[19] Y. Ye, D. Shi, P. Krause, M. Hecht, A data-driven method for estimating wheel flat length, Vehicle System Dynamics. 58 (2019) 1329–1347. doi:10.1080/00423114.2019.1620956.

[20] C. Shorten, T.M. Khoshgoftaar, A survey on Image Data Augmentation for Deep Learning, Journal of Big Data. 6 (2019). doi:10.1186/s40537-019-0197-0.

[21] D. Hendrycks, T. Dietterich, Benchmarking Neural Network Robustness to Common Corruptions and Perturbations, ArXiv.org. (2019). https://arxiv.org/abs/1903.12261 (accessed August 30, 2020).

[22] J.M. Ramirez, A. Montalvo, J.R. Calvo, A Survey of the Effects of Data Augmentation for Automatic Speech Recognition Systems, Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications Lecture Notes in Computer Science. (2019) 669–678. doi:10.1007/978-3-030-33904-3_63.

[23] Q. Wen, L. Sun, X. Song, J. Gao, X. Wang, H. Xu, Time Series Data Augmentation for Deep Learning: A Survey, ArXiv.org. (2020). https://arxiv.org/abs/2002.12478 (accessed August 31, 2020).

[24] G. Forestier, F. Petitjean, H.A. Dau, G.I. Webb, E. Keogh, Generating Synthetic Time Series to Augment Sparse Datasets, in: 2017 IEEE International Conference on Data Mining (ICDM), New Orleans, LA, IEEE, 18.11.2017 - 21.11.2017, pp. 865–870.

[25] T. Hu, T. Tang, R. Lin, M. Chen, S. Han, J. Wu, A simple data augmentation algorithm and a self-adaptive convolutional architecture for few-shot fault diagnosis under different working conditions, Measurement 156 (2020) 107539. https://doi.org/10.1016/j.measurement.2020.107539.

[26] X. Li, W. Zhang, Q. Ding, J.-Q. Sun, Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation, J Intell Manuf 31 (2020) 433–452. https://doi.org/10.1007/s10845-018-1456-1.

[27] Z. Meng, X. Guo, Z. Pan, D. Sun, S. Liu, Data Segmentation and Augmentation Methods Based on Raw Data Using Deep Neural Networks Approach for Rotating Machinery Fault Diagnosis, IEEE Access 7 (2019) 79510–79522. https://doi.org/10.1109/ACCESS.2019.2923417.

[28] K. Yu, T.R. Lin, H. Ma, X. Li, X. Li, A multi-stage semi-supervised learning approach for intelligent fault diagnosis of rolling bearing using data augmentation and metric learning, Mechanical Systems and Signal Processing 146 (2021) 107043.

[29] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., Generative Adversarial Networks, ArXiv.org. (2014). https://arxiv.org/abs/1406.2661 (accessed August 31, 2020).

[30] X. Gao, F. Deng, X. Yue, Data augmentation in fault diagnosis based on the Wasserstein generative adversarial network with gradient penalty, Neurocomputing 396 (2020) 487–494. https://doi.org/10.1016/j.neucom.2018.10.109.

[31] T. Zheng, L. Song, J. Wang, W. Teng, X. Xu, C. Ma, Data synthesis using dual discriminator conditional generative adversarial networks for imbalanced fault diagnosis of rolling bearings, Measurement 158 (2020) 107741. https://doi.org/10.1016/j.measurement.2020.107741.

[32] J. Luo, J. Huang, H. Li, A case study of conditional deep convolutional generative adversarial networks in machine fault diagnosis, J Intell Manuf (2020). https://doi.org/10.1007/s10845-020-01579-w.

[33] Y. Wang, G. Sun, Q. Jin, Imbalanced sample fault diagnosis of rotating machinery using conditional variational auto-encoder generative adversarial network, Applied Soft Computing 92 (2020) 106333. https://doi.org/10.1016/j.asoc.2020.106333.

[34] B. Han, X. Wang, S. Ji, G. Zhang, S. Jia, J. He, Data-Enhanced Stacked Autoencoders for Insufficient Fault Classification of Machinery and its Understanding via Visualization, IEEE Access 8 (2020) 67790–67798. https://doi.org/10.1109/ACCESS.2020.2985769.

[35] X. Liu, S. Alfi, S. Bruni, An efficient recursive least square-based condition monitoring approach for a rail vehicle suspension system, Vehicle System Dynamics. 54 (2016) 814–830. doi:10.1080/00423114.2016.1164869.

[36] C. Sobie, C. Freitas, and M. Nicolai, Simulation-driven machine learning: Bearing fault classification, Mechanical Systems and Signal Processing, 99 (2017) 403–419. doi: 10.1016/j.ymssp.2017.06.025.

[37] Y. Gao, X. Liu, J. Xiang, FEM Simulation-Based Generative Adversarial Networks to Detect Bearing Faults, IEEE Trans. Ind. Inf. 16 (2020) 4961–4971. https://doi.org/10.1109/TII.2020.2968370.

[38] A. Gretton, K. Borgwardt, M.J. Rasch, B. Scholkopf, A.J. Smola, A Kernel Method for the Two-Sample Problem, ArXiv e-prints, 2008.

[39] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016). doi:10.1109/cvpr.2016.90.

[40] D. Shi, Y. Ye, M. Gillwald, M. Hecht, Designing a lightweight 1D convolutional neural network with Bayesian optimization for wheel flat detection using carbody accelerations, International Journal of Rail Transportation. (2020) 1–31. doi:10.1080/23248378.2020.1795942.

[41] H.I. Fawaz, G. Forestier, J. Weber, P. Muller, J. Idoumghar, Deep learning for time series classification: a review. ArXiv.org. (2019). https://arxiv.org/pdf/1809.04356v4.pdf (accessed August 31, 2020).

[42] D.K. Appana, A. Prosvirin, J. Kim, Reliable fault diagnosis of bearings with varying rotational speeds using envelope spectrum and convolution neural networks, Soft Comput., 22 (20) (2018) 6719-6729.

[43] S. Guo, T. Yang, W. Gao, C. Zhang, Y. Zhang, An intelligent fault diagnosis method for bearings with variable rotating speed based on Pythagorean spatial pyramid pooling CNN, Sensors-Basel, 18 (11) (2018) 3857.

[44] L. Kou, Y. Qin, X. Zhao, Y. Fu, Integrating synthetic minority oversampling and gradient boosting decision tree for bogie fault diagnosis in rail vehicles, Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit. 233 (2018) 312–325. doi:10.1177/0954409718795089.

[45] R. Zunsong, An investigation on wheel/rail impact dynamics with a three-dimensional flat model, Vehicle System Dynamics. 57 (2018) 369–388. doi:10.1080/00423114.2018.1469774.

[46] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, ArXiv.org. (2017). https://arxiv.org/abs/1412.6980 (accessed August 31, 2020).

Annex I

Table 13. Number of the available data samples in each speed range within each dataset. “Good” refers to data under the healthy condition, while “Bad” refers to data with wheel flats.

Dataset	15-25 km/h	25-35 km/h	35-45 km/h	45-55 km/h	55-65 km/h	65-75 km/h	75-85 km/h	85-95 km/h	95-105 km/h
M1_Good	337	637	860	514	320	257	904	1312	1802
M1_Bad	567	337	910	184	321	382	1983	159	31
M2_Good	264	141	38	-	-	-	-	-	-
M2_Bad	328	134	6	-	-	-	-	-	-
M3_Good	67	192	337	183	177	1994	1331	-	-
M3_Bad	234	220	273	202	454	457	1412	-	-
M4_Good	-	-	-	-	-	-	-	-	-
M4_Bad	139	55	33	4	-	-	-	-	-

Annex II

Table 14. ResNet architecture within the baseline ResNet and CWT+ResNet. “Conv” refers to the convolutional layer. “BN” denotes batch normalization layer. “ReLu” is the rectified linear activation function. “Route” is the skip connection.

Block	Type	Filters	Size/Stride	Input	Output
0	Conv + BN +ReLu	8	3/2	10240 × 1	5120 × 8
1	Conv + BN +ReLu	8	3/2	5120 × 8	2560 × 8
2	Maxpooling	-	-/10	2560 × 8	256 × 8
3	Conv + BN + ReLu	16	3/2	256 × 16	128 × 16
4	Conv + BN + ReLu	16	3/2	128 × 16	64 × 16
5	Conv + BN + ReLu	16	3/1	64 × 16	64 × 16
6	Conv + BN + ReLu	16	3/1	64 × 16	64 × 16
7	Route to 5	-	-	-	
8	Conv + BN + ReLu	16	3/1	64 × 16	64 × 16

9	Conv + BN + ReLu	16	3/1	64 × 16	64 × 16
10	Route to 8	-	-	-	
11	Maxpooling	-	-/2	64 × 16	32 × 16
12	GlobalAveragePooling	-	-	32 × 16	16
13	Fully connected layer	-	-	16	2

Table 15. ResNet architecture within the baseline HT+ResNet. “Conv” refers to the convolutional layer. “BN” denotes batch normalization layer. “ReLu” is the rectified linear activation function. “Route” is the skip connection.

Block	Type	Filters	Size/Stride	Input	Output
0	Conv + BN +ReLu	8	3/1	256 × 1	256 × 8
1	Conv + BN +ReLu	8	3/1	256 × 8	256 × 8
2	Maxpooling	-	-/2	256 × 8	128 × 8
3	Conv + BN + ReLu	16	3/2	128 × 16	64 × 16
4	Conv + BN + ReLu	16	3/2	64 × 16	32 × 16
5	Conv + BN + ReLu	16	3/1	32 × 16	32 × 16
6	Conv + BN + ReLu	16	3/1	32 × 16	32 × 16
7	Route to 5	-	-	-	
8	Conv + BN + ReLu	16	3/1	32 × 16	32 × 16
9	Conv + BN + ReLu	16	3/1	32 × 16	32 × 16
10	Route to 8	-	-	-	
11	Maxpooling	-	-/2	32 × 16	16 × 16
12	GlobalAveragePooling	-	-	16 × 16	16
13	Fully connected layer	-	-	16	2

Table 16. Essential hyperparameters of the baseline GBDT

Method	Learning rate	Num. of estimators	Criterion	Max. depth
GBDT	0.05	200	Friedman_mse	8

Annex III

Table 17. Architecture and hyperparameters of cDCGAN

Module	Learning rate	Network layer	Filters	Size/Stride	Output
Discriminator	0.003	Conv + BN + LeakyReLu	16	7/2	512 × 16
		Conv + BN + LeakyReLu	16	7/2	256 × 16
		Conv + BN + LeakyReLu	16	5/2	128 × 16
		Conv + BN + LeakyReLu	16	5/2	64 × 16
		Fully connected layer	-	-	2
		Sigmoid_output1	-	-	1
		Softmax_output2	-	-	1
Generator	0.0008	Embedding_input1	-	-	50
		Fully connected layer_input1	-	-	64
		Reshape_input1	-	-	64 × 1
		Fully connected layer_input2	-	-	1024
		LeakyReLu_input2	-	-	1024
		Reshape_input2	-	-	64 × 16
		Concatenate_input1_input2	-	-	64 × 17
		Deconv+ BN + LeakyReLu	16	4/2	128 × 16
		Deconv+ BN + LeakyReLu	16	4/2	256 × 16
		Deconv+ BN + LeakyReLu	16	4/2	512 × 16
		Deconv+ BN + LeakyReLu	16	4/2	1024 × 16
		Conv + tanh	1	3/1	1024 × 1

Table 18. Architecture and hyperparameters of GAN

Module	Learning rate	Network layer	Filters	Size/Stride	Output
Discriminator	0.001	Conv + BN + LeakyReLu	16	7/2	512 × 16
		Conv + BN + LeakyReLu	16	7/2	256 × 16
		Conv + BN + LeakyReLu	16	5/2	128 × 16
		Conv + BN + LeakyReLu	16	5/2	64 × 16
		Fully connected layer	-	-	1
		Sigmoid	-	-	1
Generator	0.0002	Fully connected layer	-	-	1024
		LeakyReLu	-	-	1024
		Reshape	-	-	64 × 16
		Deconv+ BN + LeakyReLu	16	4/2	128 × 16
		Deconv+ BN + LeakyReLu	16	4/2	256 × 16
		Deconv+ BN + LeakyReLu	16	4/2	512 × 16
		Deconv+ BN + LeakyReLu	16	4/2	1024 × 16
		Conv + tanh	1	3/1	1024 × 1

6. Dynamic Displacement Measurement

This publication presents the proposed virtual point tracking for real-time target-less dynamic displacement measurement to support track geometry monitoring. It addresses the specific challenge of track geometry monitoring, where track alignment cannot be accurately derived by pure vibration monitoring. We hypothesized that track alignment can be estimated by combing the lateral acceleration and displacement of the wheel on the rail. Therefore, we proposed to use an off-the-shelf camera to film the motion of the wheels and developed an intelligent algorithm to derive the lateral displacements of the wheels relative to the rails in real time. In addition, the general robustness problem of optical sensing was investigated under different weather conditions by image corruption techniques. The proposed algorithm was validated on three field tests under different conditions. We found that the measurement uncertainty of the derived lateral wheel displacement is larger than that of track alignment prerequired in the track inspection standard [42]. For further improvement, a new camera with a narrow angle of field should be applied.

The accepted manuscript below is the article published by Elsevier in Mechanical Systems and Signal Processing on 6th October 2021, available online:

https://doi.org/10.1016/j.ymssp.2021.108482

Deep Learning Based Virtual Point Tracking for Real-Time Target-less Dynamic Displacement Measurement in Railway Applications

Dachuan Shi *a, Eldar Šabanovičb, Luca Rizzettoc, Viktor Skrickijb, Roberto Oliverioc, Nadia Kavianic, Yunguang Ye a, Gintautas Bureikab, Stefano Riccic, Markus Hechta

a Institute of Land and Sea Transport Systems, Technical University of Berlin, Berlin 10587, Germany

b Faculty of Transport Engineering, Vilnius Gediminas Technical University, LT-10223 Vilnius

c Department of Buildings and Environmental Engineering, Sapienza University of Rome, 00185 Roma

* Corresponding E-mail: dachuan.shi@tu-berlin.de, Tel.: +49 030 314 79806 and Fax: +49 030 314 22529

Abstract

In the application of computer-vision-based displacement measurement, an optical target is usually required to prove the reference. If the optical target cannot be attached to the measuring objective, edge detection, feature matching, and template matching are the most common approaches in target-less photogrammetry. However, their performance significantly relies on parameter settings. This becomes problematic in dynamic scenes where complicated background texture exists and varies over time. We propose virtual point tracking for real-time target-less dynamic displacement measurement, incorporating deep learning techniques and domain knowledge to tackle this issue. Our approach consists of three steps: 1) automatic calibration for detection of region of interest; 2) virtual point detection for each video frame using deep convolutional neural network; 3) domain-knowledge based rule engine for point tracking in adjacent frames. The proposed approach can be executed on an edge computer in a real-time manner (i.e. over 30 frames per second). We demonstrate our approach for a railway application, where the lateral displacement of the wheel on the rail is measured during operation. We also implemented an algorithm using template matching and line detection as the baseline for comparison. The numerical experiments have been performed to evaluate our approach's performance and latency in a harsh railway environment with dynamic complex backgrounds. We make our code and data available at https://github.com/quickhdsdc/Point-Tracking-for-Displacement-Measurement-in-Railway-Applications.

Keywords: Point tracking; Computer vision; Displacement measurement; Photogrammetry; Deep learning; Railway

1. Introduction

1.1 Background and motivation

Thanks to the rapid advance in computer vision (CV) in the last decade, there is a noticeable increase in many sectors applying photogrammetry to inspect structures. A typical photogrammetry application is the deformation measurement of large structures such as bridges in civil engineering [1]. In the railway sector, Zhan et al. [2] proposed to use high-speed line scan cameras to measure catenary geometry parameters, calibrated by a 1-D optical target. Li et al. [3] used CV to monitor track slab deformation. Two optical targets were attached to the track slab to extract region of interest (RoI). In the aforementioned applications, optical targets are required to provide measurement references. When optical targets cannot be attached to the structure, edge detection, digital image correlation, template

matching and template matching are the most common solutions [4,5]. However, they may suffer from robustness problems due to complex backgrounds. Wang et al. [6] combined a deep learning model and a template-matching driven tracking algorithm for recognition and tracking of rail profiles from laser fringe images. Jiang et al. [7] proposed a robust line detection workflow for the uplift measurement of railway catenary, addressing the problem caused by noisy background. The measurement was done in a static condition by fixing the camera system next to the railway. The challenge we are facing is more complex. We are addressing the issue of real-time target-less dynamic displacement measurement in front of noisy and varying backgrounds. In the context of the railway, we aim to monitor the wheels' lateral motion of a railway vehicle relative to the rail in regular railway operation. It tackles an unsolved railway issue related to track geometry (TG) monitoring.

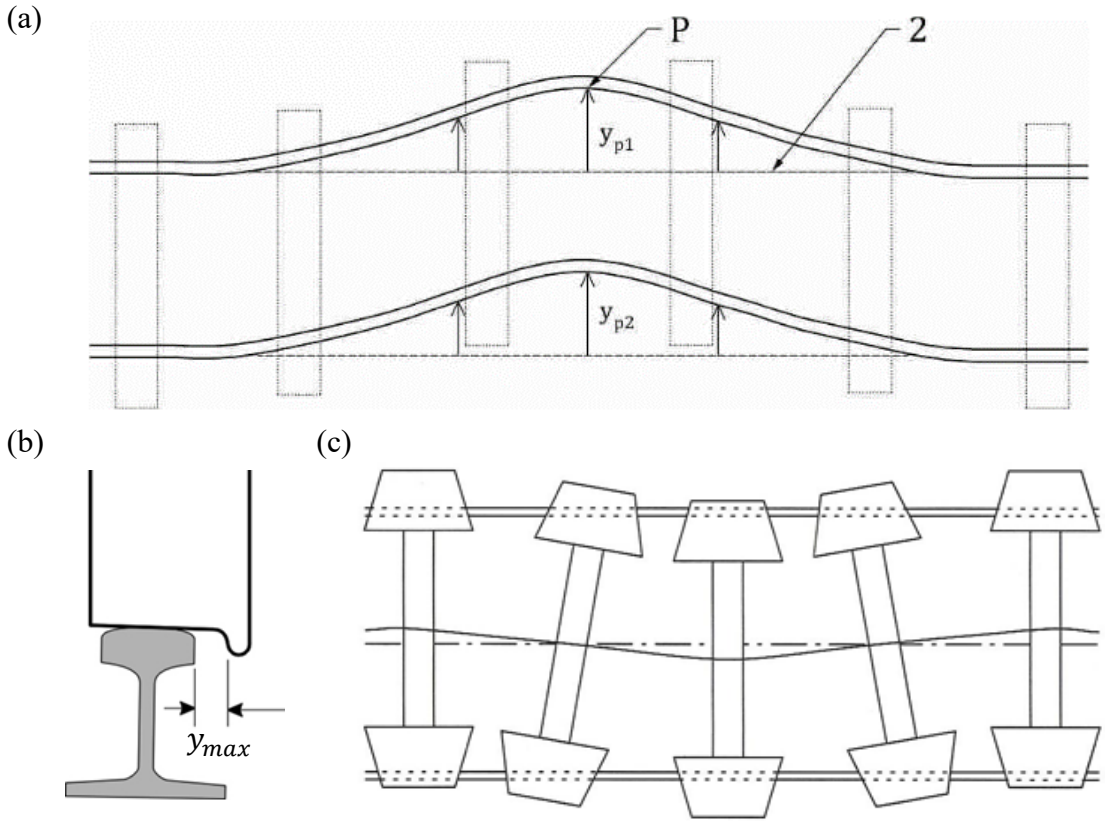


Figure 1. (a) lateral alignment of the left rail y_{p1} and right rail y_{p2}, where P denotes the rail reference point and "2" denotes the reference rail line [8]; (b) illustration of wheel/rail gauge clearance [9]; (c) schematic view of hunting motion [9]

TG parameters are defined as the indicators for track maintenance in the European railway standard EN 13848-1 [8], namely track gauge, cross-level, longitudinal level, lateral alignment, twist. Based on the statistical study in a European project [10], longitudinal level and lateral alignment are the most critical parameters for maintenance decisions. Other parameters are either highly linearly correlated to longitudinal level or degrading slower than longitudinal level. Traditionally, TG parameters are measured by the dedicated TG inspection systems in regular inspections, which are typically based on high-value laser triangulation sensors or/and inertial measurement units (IMU). The inspection interval is usually defined as several months, which results in the lack of up-to-date information on track conditions [11].

In order to improve the information availability and enable efficient maintenance decisions, TG monitoring on in-service vehicles was proposed. TG monitoring has been studied extensively in the last two decades [12]. Accelerometers have been commonly accepted as the most promising sensor for TG monitoring due to their low cost and robustness. It has been validated in previous studies that longitudinal level can be accurately reconstructed from vertical accelerations [12]. However, lateral alignment cannot be accurately derived from lateral accelerations due to railway vehicle dynamics. As shown in Figure 1 (a), lateral alignment is defined as the lateral deviation y_p between the actual and reference rail line in the horizontal plane at the point *P* on the each rail, being at the position 14 mm below the top of the railhead for the standard rail profile UIC 60E1 [8]. It is expected that the vehicle wheels follow the excitation of lateral alignment in the lateral direction so that lateral alignment can be estimated by accelerations. However, the wheels do not follow lateral alignment exactly as the vertical one. One reason is that the wheel has a freedom of movement in the lateral direction in a clearance y_{max}, which refers to the clearance between the wheel flange and the railhead edge, as shown in Figure 1 (b). Another reason given by True et al. [13] is that the lateral irregularities simultaneously act on the lateral force and the spin torque of the wheel-rail contact force, which are nonlinearly coupled. This indicates that lateral alignment cannot be accurately derived from the perspective of vehicle dynamics.

To tackle this issue, Ripke et al. [10] combined acceleration measurements with a multi-body dynamic simulation (MBS) model of the vehicle, on which the accelerometers were installed. The alignment was estimated by accelerations and then corrected by the MBS model using a dedicated correction mechanism. The estimated alignment was compared with the one measured by a commercial TG inspection system. However, this approach was vitiated by the comparison results. Rosa et al. [14] proposed a model-based method, combing MBS and Kalman filter, to estimate lateral alignment. However, a critical issue of a model-based method is that the model cannot take into account the wear process of wheel profile, which has significant effects on vehicle dynamics. Rosa et al. [15] proposed to train a machine learning (ML) based classifier to detect large track lateral irregularities. From a maintenance perspective, two classes of alignment values have been defined by thresholding. The measured alignment values in class 1 indicate the normal track condition, and no specific maintenance measure has to be taken. Class 2 indicates severe track degradation, requiring short-term maintenance measures. As well known, the features as the input for the classifier are essential for classification performance. In [15], only standard deviations of accelerations were defined as the features, which may not contain abundant classification information. The test accuracy was under 90%. This approach also evaded the reconstruction of alignment values.

Based on the previous studies, we conclude that the wheels' lateral displacement on the rail (LDWR) is indispensable to estimate the accurate lateral alignment. Therefore, we propose deep-learning (DL) based virtual point tracking to measure LDWR in a real-time manner. Combined with an accelerometer, the proposed system can be used to reconstruct the alignment for a massive deployment on in-service trains.

Our approach can also be used for hunting detection, as shown in Figure 1 (c), which indicates the dynamic instability of railway vehicles and is thus safety-relevant. The current detection methods are based on acceleration measurements. Detection performance may suffer due to alignment, particularly when detecting small-amplitude hunting instability [16].

Monitoring LDWR can fundamentally solve this problem. Furthermore, monitoring LDWR is a central part of the active wheelset steering systems using lateral displacement control strategy [17]. LDWR can express the rolling radius of the wheels. If the lateral displacement can satisfy a specific condition, the wheelset will be in pure rolling condition, resulting in minimal wear in a curve. Within the control chain, the measured LDWR provides feedback to the control system [16].

1.2 Related work

Our task is to detect and track the virtual points for target-less dynamic displacement measurement in front of noisy and varying backgrounds. We introduce DL approaches for human pose estimation (HPE) for point detection. In the following, we review the related work for the measurement of LDWR, photogrammetry for displacement measurement and DL-based HPE, respectively.

A commercial system based on laser triangulation sensors was used to measure LDWR for active wheel control [18]. The laser sensors were mounted on the wheelset axle, closely pointing at the railhead. The accuracy of the laser sensors is of the order of 0.1 mm. However, the sensors are subject to high vibrations at the wheelset level, which could degrade their lifetime and performance. Kim [17] used a charge-coupled device camera to measure LDWR for active wheel control. LDWR was measured by tracking the rail line and the wheel template. The proposed algorithm was mainly based on conventional image processing techniques of filtering, edge detection, template matching and line detection. However, it requires parameter tuning as a part of calibration for different environmental conditions, which is a laborious and time-consuming process. Yamamoto [19] used a thermographic camera installed on the bogie frame to view the wheel-rail contact area. Despite successful localization, the thermographic camera has a low resolution of 320×240 pixels and thus a low measurement resolution in millimeters. It cannot fulfill the requirements of TG monitoring.

Photogrammetry for displacement measurement is typically divided into five steps: camera calibration, RoI selection, feature extraction, visual tracking and displacement calculation. The applicable methods for each step have been reviewed by Baqersad et al. [4] and Dong et al. [5]. Edge detection and template matching algorithms are frequently applied in target-less photogrammetry, where structures' inner edges or features are extracted for object detection and tracking. Guo et al. [20] introduced Lucas-Kanade template tracking algorithm for dynamic displacement measurement. This algorithm was able to process images from high-speed cameras. However, it requires a pre-defined template that remains visually stable within the measurement. This prerequisite may not be fulfilled in the case of noisy and dynamic backgrounds. Cha et al. [21] applied a phased-based optical flow algorithm for motion tracking of structures. However, optical flow approaches are sensitive to the variation of illumination and backgrounds. Dong et al. [22] applied spatio-temporal context learning for RoI tracking and Taylor approximation for subpixel motion estimation. The robustness of this approach has been validated for small motion tracking in laboratory experiments by varying the illumination and humidity. Apart from the conventional image processing techniques, DL has been introduced in photogrammetry. Yang et al. [23] combined convolutional neural network (CNN) and recurrent neural network for modal analysis of the structures. A vanilla CNN model was used for spatial feature extraction, while a long short-term memory network was used to model the temporal dependency over the measurement period. The outputs were

the natural frequencies. In the images, the specimens were highlighted through the laser point of a laser vibrometer, which was intended to provide the ground-truth natural frequencies. This laser point may unexpectedly become the optical target and lead to success. However, this was not analyzed in the paper. Liu et al. [24] used CNN for vibration frequency measurement of bridges. The 9×9 RoI in the frames was manually selected and flattened as 1D sequences fed into CNN as the inputs. The CNN outputted vibration frequencies. The manual selection of RoI played an essential role. RoI must contain an objective with clear edges and a clear background.

Displacement could be measured by tracking reference points, which conventionally refer to optical targets. Alternatively, virtual points can be defined in measuring objectives and automatically detected by employing advanced CV techniques. A successful application of virtual point detection/tracking is HPE. HPE is a fundamental CV task, aiming to estimate the posture of human bodies. In the last decade, CV-based HPE has been under rapid development thanks to DL techniques [25]. For HPE, the virtual points are defined as a series of points at a human body's kinematic joints [26], such as eyes, neck, elbows and ankles. In terms of problem formulation, the methods for 2D HPE fall into two categories, namely regression-based and detection-based methods [25]. Detection-based methods transfer the virtual points into 2D representations (e.g. heatmaps) and then map the input image to these heatmaps. This method is commonly used in the modern CNN architectures for HPE, such as the stacked hourglass network [27], the encoder-decoder network [28] and the high-resolution network [29]. In contrast, regression-based methods directly output the coordinates of the virtual points from a given image. It is much harder to map the input 2D image directly to the point coordinates than to the 2D heatmaps. Therefore, a more powerful backbone architecture is required. The CNN network architecture proposed by Luvizon et al. [30] consisted of Inception-V4 for feature extraction and multiple prediction blocks to predict the heatmap of each point. Finally, the Soft-argmax layer was added to regress the coordinates of a keypoint from the heatmap. In recent work, Bazarevsky et al. [31] combined both methods in one network. The network has two heads in the training process, one for prediction of the heatmap and the other for regression of the coordinates. Only the regression head is kept for online inference, while the heatmap prediction head is removed.

1.3 Challenges and contributions

In our railway application for dynamic displacement measurement, we are facing the following challenges. Firstly, it is a monitoring task, rather than an inspection. Monitoring devices are typically developed for massive deployment and full automation during operation. Therefore, Monitoring devices are expected to have high automation and low investment costs. Secondly, the CV system is installed on the railway vehicle facing a wheel, moving along the railway track. An optical target cannot be attached to the rotating wheel. The common target-less approaches, such as edge detection, template matching and line detection, are prone to performance losses in front of dynamic complex backgrounds, where complex textures such as ballast, sleepers and plants exist and vary over time. Thirdly, the images should be processed in a real-time manner, as the calculated LDWR has to be fused with the acceleration measurements to reconstruct track lateral alignment. To address these challenges, we propose a novel approach to virtual point tracking. To our best knowledge, our work is the first attempt to combine HPE and domain knowledge for displacement measurement.

In this paper, we mainly focus on the proposed algorithm for virtual point tracking. The calculation of displacement between the virtual points has been introduced and validated in [32]. The fusion of CV and accelerometers will be addressed in future work. Our main contributions are summarized as follows:

1. A novel approach of virtual point tracking for target-less displacement measurement is proposed, consisting of RoI detection, point detection and point tracking.
2. A lightweight CNN architecture is proposed for real-time point detection in each video frame.
3. A rule engine based on railway domain knowledge is defined for point tracking.
4. Implementation of the proposed approach for real-time edge computing
5. We make our code and data available at https://github.com/quickhdsdc/Point-Tracking-for-Displacement-Measurement-in-Railway-Applications

The structure of the paper is as follows. Section 2 briefly introduces the hardware of the designed monitoring system. Section 3 describes the proposed approach for virtual point tracking in detail, the implemented baseline method, and the image corruption methods for data augmentation. In Section 4, extensive experiments are conducted to evaluate as well as validate each step in our approach and demonstrate the entire approach. In addition, computational complexity and robustness are discussed. Section 5 draws the conclusions.

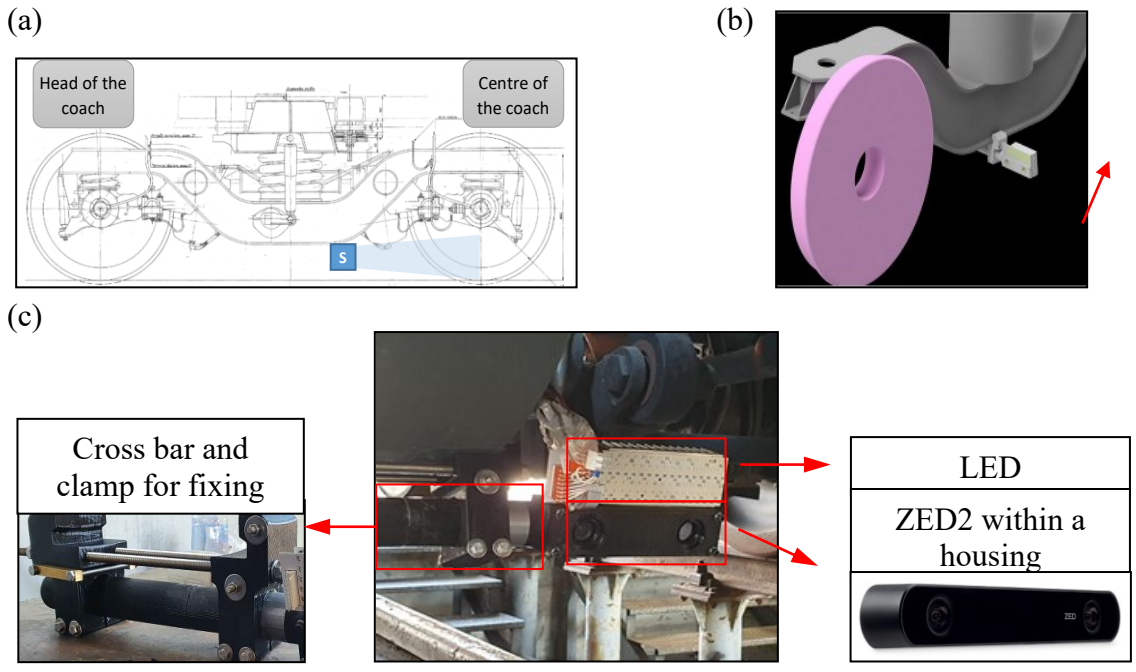


Figure 2. (a) drawing of the camera position on the bogie frame; (b) CAD model of the camera installation; (c) monitoring system installed on the bogie frame

2. Hardware components of the monitoring system

The proposed monitoring system consists of an off-the-shelf stereo camera, an air cleaning system, a processing unit, a lighting system, and a mounting system with dampers. The air cleaning system aims to clean the camera lens by blowing compressed air regularly. This is a

standard solution to avoid dirt in the optical systems [12]. From the software perspective, we enhance the robustness of the algorithm against the image's visual corruptions. This will be introduced in Section 4. For optical sensing, ZED2 stereo camera is used in our system [33], which is configured to output videos with the resolution of 1920×1080 pixels at the sample rate of 30 frames per second (fps). Any comparable cameras can be used as well. The depth information is merely used for displacement measurement. The algorithms described in this are directly applicable for 2D images obtained by regular CCD cameras. As the processing unit, Nvidia Jetson Tx2 has 256 core NVIDIA Pascal architecture and ARMv8 6 core multi-processor CPU complex, enabling real-time execution of DL models [34]. The mounting system consisting of the vibration dampers, a crossbar and a clamp can be easily installed on different bogies types. The camera housing is equipped with an external lighting system, which consists of a series of LEDs. The entire system is installed on the bottom of the bogie frame, facing the wheel, as shown in Figure 2. Two systems are required to monitor the wheel-rail pair on the left and right sides simultaneously. In the current hardware implementation, the cleaning system is not included. The processing unit is inside the vehicle cabin, connecting to and powering the camera. The hardware of the monitoring system will be further improved for long-term monitoring.

3. Approach for virtual point tracking

We formulate the task of dynamic displacement measurement to track virtual reference points and calculate the distance between two virtual reference points. This paper focuses on virtual point tracking. The displacement calculation method has been introduced and validated in our previous work [32]. We define three reference points on the wheel (P_w) and rail (P_{r1} and P_{r2}) respectively. P_{r1} refers to the reference point P for lateral alignment [8], as introduced in Section 1.1. P_{r2} is the symmetry point of P_{r1} on the other side of the railhead edge. The distance between P_{r1} and P_{r2} is the width of the railhead. The lateral displacement D of the wheel on the rail (LDWR) is represented by the lateral distance between P_w and P_{r1}, see Figure 3. The relative lateral motion of the wheel is represented by the changes of D (i.e. ΔD) over time, which is the output of the monitoring system. The point P_{r2} is defined for tracking mechanism, which is explained in Section 3.3.

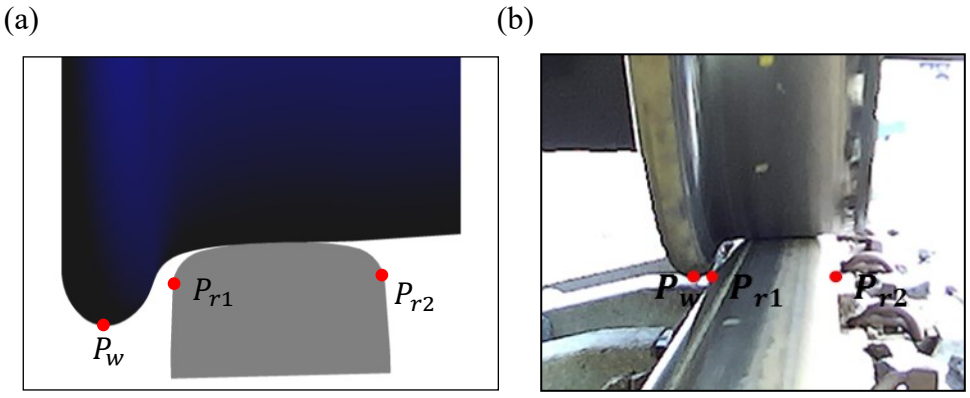


Figure 3. (a) defined keypoints P_w, P_{r1} and P_{r2} illustrated in the animation; (b) marked in the real photo (right)

Virtual point tracking consists of three steps, as shown in Figure 4. The first step is the calibration, executed for the first-time installation. This calibration process detects RoI, which refers to the wheel-rail contact area. The outputs are the coordinates of the centre point of RoI. Moreover, the distance between the camera and the wheel is obtained as the stereo camera's depth. The distance is an input parameter for displacement calculation. In the case of using a CCD camera, the distance has to be manually measured. The next steps are executed to detect and track virtual points in real-time. Next, we will introduce each step in detail.

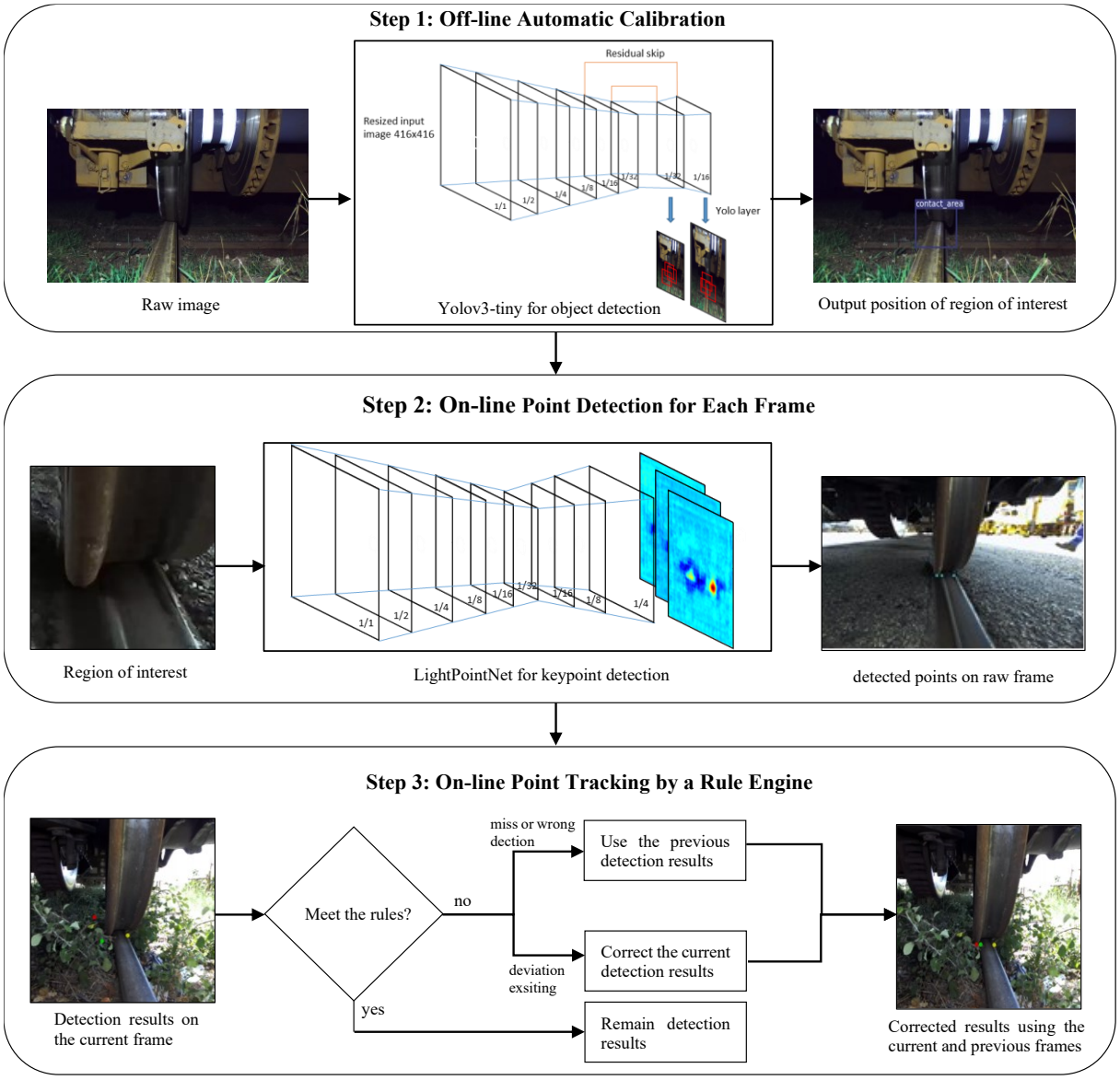


Figure 4. Approach to track virtual points on the wheel and rail

3.1 Step 1: Off-line automatic calibration

As the resolution of each frame is 1920×1080 pixels, it is necessary to resize the image prior to feeding it to CNN. However, resizing and restoring the image cause additional measurement errors for point detection. To avoid the step of image resizing, we propose cropping the RoI from the raw image. We choose a mature object detection technique based

YOLOv3 [35], which is a CNN architecture and has been widely deployed for diverse applications. We adopt a modified version of YOLOv3 for RoI detection, called YOLOv3-tiny. The architecture of YOLOv3-tiny is shown in Figure 4 and Table 1. The first 13 layers are used for feature extraction, known as Darknet. The input for Darknet is the images with 416×416 pixels downsized from the original 1920×1080 ones. The output 1024 feature maps of Darknet have the dimension of 13 × 13 pixels. The layers 13-16 are to make predictions. YOLOv3-tiny pre-defines 3 anchor boxes to predict the objects' bounding boxes and generates 6 parameters for each bounding box, i.e. x and y coordinate of its center point, its width and height, a prediction score, and the object class. In our case, only one class is defined for the wheel-rail contact area, while others are backgrounds. The 14 and 15 layers downsized the number of the feature maps to 18. Each 13 × 13 feature map regresses one parameter. The 16th layer compares the prediction with the ground truth to calculate the loss. The loss consists of classification loss, localization loss, and confidence loss. The detailed loss functions can be found in the original paper [35]. YOLOv3-tiny predicts at two different scales. The first scale uses the aforementioned 13 × 13 feature maps and passes the feature maps to the second scale, which is implemented in the layers 17-23. The outputs of YOLOv3-tiny are the candidates of RoI with the dimension of $N \times 18$, where N denotes the number of the candidates. The final prediction is selected by objectness score thresholding and non-maximal suppression [35].

Table 1. Adapted YOLOv3-tiny architecture

Layer Index	Type	Filters	Size/Stride	Output
0	Convolutional	16	3 × 3/1	416 × 416
1	Maxpool		2 × 2/2	208 × 208
2	Convolutional	32	3 × 3/1	208 × 208
3	Maxpool		2 × 2/2	104 × 104
4	Convolutional	64	3 × 3/1	104 × 104
5	Maxpool		2 × 2/2	52 × 52
6	Convolutional	128	3 × 3/1	52 × 52
7	Maxpool		2 × 2/2	26 × 26
8	Convolutional	256	3 × 3/1	26 × 26
9	Maxpool		2 × 2/2	13 × 13
10	Convolutional	512	3 × 3/1	13 × 13
11	Maxpool		2 × 2/1	13 × 13
12	Convolutional	1024	3 × 3/1	13 × 13

13	Convolutional	256	1 × 1/1	13 × 13
14	Convolutional	512	3 × 3/1	13 × 13
15	Convolutional	18	1× 1/1	13 × 13
16	YOLO			
17	Route 13			
18	Convolutional	128	1× 1/1	13 × 13
19	Upsampling		2 × 2/1	26 × 26
20	Route 19, 8			
21	Convolutional	256	3 × 3/1	26 × 26
22	Convolutional	18	1× 1/1	26 × 26
23	YOLO			

3.2 Step 2: On-line point detection for each frame

Point detection is an essential step in our approach. We propose LightPointNet, a lightweight CNN architecture using integral regression for real-time point detection on each video frame. LightPointNet consists of an encoder for hierarchical feature extraction and a decoder for heatmap prediction. Inspired by [28,36,37], the key insights behind LightPointNet are the lightweight backbone, the straightforward encoder-decoder structure and integral loss.

The architecture of LightPointNet is shown in Figure 4 and Table 2. The first 12 blocks build the encoder, while the last four blocks build the decoder. The whole network is built by stacking three building blocks. The first block "Conv" refers to a convolutional block, consisting of a convolutional layer, a batch normalization layer and the hard-swish function (HS) as the activation function for nonlinearity (NL). In this block, 16 convolutional filters parameterized by the weights $W \in \mathcal{R}^{3\times3}$ are performed on the input image $I \in \mathcal{R}^{256\times256\times3}$ to generate the feature map $F \in \mathcal{R}^{128\times128\times16}$. Then, mini-batch normalization [38] and hard-swish [36] are performed on the feature map F to reduce internal corvariate shift and add nonlinearity. The swish function aims to solve the dead neuron problem of ReLu, which is the most common activation function for CNN. The hard version of the swish function reduces the computational complexity of the original one, defined as:

$$Hswish = x \cdot (ReLu6(x+3)) \,/\, 6 \tag{1}$$

$$ReLu6 = min\,(\max\,(0, x), 6) \tag{2}$$

The convolutional block is followed by 11 blocks of inverted residual and linear bottleneck (Bneck) [36]. Bneck is a modified version of the original residual operation [39], which enables the skip connection between the input and output feature maps by following a wide-narrow-wide bottleneck structure in terms of the channel number. Bneck uses an inverted

bottleneck with a narrow-wide-narrow structure. It is implemented by stacking three convolutional layers. The first one is 1×1 pointwise convolution to expand the input channel dimension c by a factor e, followed by an activation function. The expanded size for each Bneck block is given in the column "Exp size" of Table 2. The second one is 3×3 depthwise convolution with an activation function, keeping the channel dimension unchanged. Replacing regular convolution with depthwise convolution is an effective lightweight measure. This will be further described in Section 4.6. The third one is 1×1 pointwise convolution to recover the output channel dimension to c, allowing the identity skip connection between the block inputs and outputs. The third convolution layer does not involve an activation function and thus remains linear. Bneck can be combined with Squeeze-and-Excite (SE) [40], which improves channel interdependencies of feature maps. The column "SE" indicates the presence of the SE module. The structure of SE can be found in [40]. We merely replace the original sigmoid activation with the hard sigmoid function, which is defined as:

$$HSig = ReLu6(x + 3) \tag{3}$$

The Bneck blocks extract hierarchical features and downsize the feature maps to 8×8. Afterward, 3 blocks of "ConvTranspose" are stacked to upsample the feature maps to 64×64. ConvTranspose consists of a transposed convolutional layer, a batch normalization layer, and an activation function. The final Conv block aims to output the final heatmaps $H \in \mathcal{R}^{64\times64\times3}$ for the defined three virtual points P_w, P_{r1} and P_{r2}, respectively.

Table 2. LightPointNet architecture

Block	Type	NL	SE	Exp size	Filters	Size/Stride	Output
0	Conv	HS	false	-	16	3×3/2	128×128×16
1	Bneck	RE	false	16	16	3×3/2	64×64×16
2	Bneck	RE	true	72	24	3×3/2	32×32×24
3	Bneck	RE	false	88	24	3×3/1	32×32×24
4	Bneck	HS	false	96	40	5×5/2	16×16×40
5	Bneck	HS	true	240	40	5×5/1	16×16×40
6	Bneck	HS	true	240	40	5×5/1	16×16×40
7	Bneck	HS	true	120	48	5×5/1	16×16×48
8	Bneck	HS	true	144	48	5×5/1	16×16×48
9	Bneck	HS	true	192	64	5×5/2	8×8×64
10	Bneck	HS	true	384	64	5×5/1	8×8×64
11	Bneck	HS	true	384	64	5×5/1	8×8×64

12	ConvTranspose	RE	false	-	256	4×4/2	16×16×256
13	ConvTranspose	RE	false	-	256	4×4/2	32×32×256
14	ConvTranspose	RE	false	-	256	4×4/2	64×64×256
15	Conv	-	false	-	3	1×1/1	64×64×3

The common regression-based HPE methods use mean square errors (MSE) between the predicted heatmaps and the ground-truth ones as the optimization objective for CNN training. The point coordinates are obtained using arguments of the maxima (argmax). This induces inevitable quantization errors. In our case, the resolution of the heatmaps is one-fourth of the input image. That means the quantization errors and the prediction errors at the heatmap level are enlarged four times. To avoid this, we use the discrete softmax function instead of argmax, which is differentiable and thus can be included in the training process [37]. The predicted point coordinates are the integration of the heatmap weights in all locations along the x and y-axis, respectively. In this way, the predicted point coordinates are continuous and thus on a subpixel level. Figure 5 exemplarily shows the prediction process of a wheel reference point. Finally, the loss function as the training objective, termed integral loss, is defined as the absolute differences between the ground-truth coordinates and the predicted ones using integral regression.

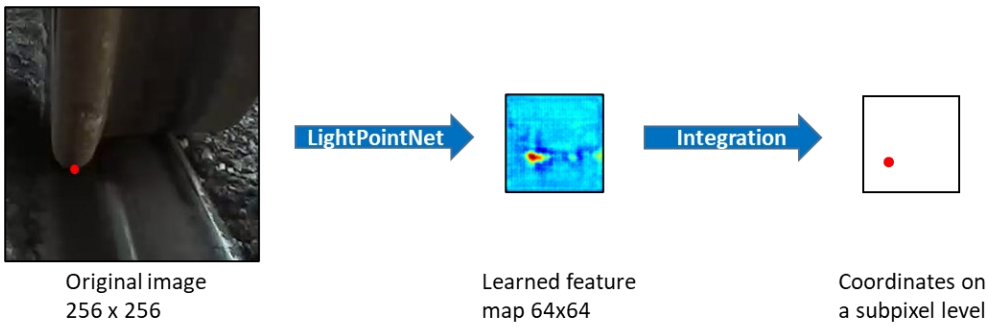


Figure 5. Prediction process of a wheel reference point

3.3 Step 3: On-line point tracking by a rule engine

LightPointNet may output false detections during regular railway operation, especially in corner cases. For instance, as shown in the third block in Figure 4, the grass occludes the points P_w and P_{r1}. A correct point detection is impossible on this single frame. To correct false detections, we propose a rule engine as the point tracker. The rule machine independent from CNN has two advantages in our application. Unlike the problems of object tracking and human pose tracking, our railway application has similar scenes, i.e. the wheels running on the rails. The virtual points have spatial correlations with each other under specific geometric constraints. This allows defining the rules based on railway domain knowledge. On the other hand, we have specific challenges in terms of data availability, which is a common issue for any domain-specific application. As in a recent work of real-time human pose tracking indicated [31], 85k annotated images were used to train a pose tracking network. In industrial practice, data collection and annotation are laborious and costly. Much fewer data obtained in

field tests are available to train CNN. Therefore, we combine a DL-based point detector with a domain-knowledge-based tracker to achieve real-time point tracking, which requires much less training data. Furthermore, the rule machine can automatically identify the corner cases, once the CV system is deployed for a long-term trial. The corresponding video frames can be collected to update the model for performance improvement.

Table 3. Defined rules, indicators and thresholds in the rule engine

Index	Rules	Indicators	Thresholds
1	Y coordinate of the detected points should remain constant in comparison to the reference one (which can be manually defined in the calibration process or using the detection result on the first frame).	RMSE $\sigma_y = \frac{1}{3}\sqrt{\sum_{i=1}^{3}(y_i - y_{ref})^2}$	$\sigma_{yTH_1} = 5\ mm$ $\sigma_{yTH_0} = 10\ mm$
2	Y coordinate of the detected points should remain constant in the adjacent frames.	RMSE $\sigma_{yt} = \frac{1}{3}\sqrt{\sum_{i=1}^{3}(y_{i,t} - y_{i,t-1})^2}$	$\sigma_{ytTH_1} = 5\ mm$ $\sigma_{ytTH_0} = 10\ mm$
3	The width of the rail head calculated by the x coordinates of the two rail reference points (P_{r1} and P_{r2}) should remain constant.	Difference $d_x = \lvert x_{r1} - x_{r2}\rvert$	$d_{xTH_1} = 5\ mm$ $d_{xTH_0} = 10\ mm$
4	The two rail reference points should move in the same lateral direction or remain unchanged in the adjacent frames.	Boolean $(x_{r1,t} - x_{r1,t-1}) \cdot (x_{r2,t} - x_{r2,t-1}) \geq 0$	$B_{TH_0} = True$
5	The wheel lateral displacement between two adjacent frames should be smaller than that calculated by the maximal wheel lateral acceleration.	Lateral displacement $\Delta D = \lvert x_{w,t} - x_{w,t-1}\rvert$	$\Delta D_{TH_0} = 0.5 \cdot a_{ymax} \cdot \Delta t^2$

The flow chart of the rule engine is shown in Figure 4 Step 3. We define the following rules as well as the corresponding indicators and thresholds in Table 3. Each rule is independently examined. Rule 1 and 2 constrain the y-coordinates of the virtual points, which represent the projection of the relative vertical and longitudinal motion between the camera and the wheel in the horizontal plane. Three virtual points are defined at the same horizontal level, i.e. y_{ref}.

The relative movement of rail reference points P_{r1} and P_{r2} does not exist. The only reasonable disparity of y-coordinates between the wheel and rail reference point is linked with wheel bounce due to a high excitation of rail irregularities. However, this is a rare event and can be compensated by wheel acceleration measurement. Therefore, we consider that the y-coordinates of three points should vary by a small margin. Root mean squared error (RMSE) is used as the indicator for Rule 1 and 2. When σ_y in Rule 1 exceeds the threshold σ_{yTH_0}, the detection results are regarded as unreliable. The detection results for the current frame are thus inherited from the previous frame. When σ_y lies between σ_{yTH_0} and σ_{yTH_1}, a correction mechanism is applied to the detection results. We take the averaged coordinates of the previous and current frames as the corrected values. The values of the threshold σ_{yTH_0} and σ_{yTH_1} indicate the error tolerance of the virtual point detection results. As the detection errors are expected to be at the level of 1 mm, we empirically define $\sigma_{yTH_0} = 5$ and $\sigma_{yTH_1} = 10$. The thresholds do not have to be changed when the monitoring system is installed on a different vehicle. Similarly, Rule 3 and 4 constrain the difference of x-coordinates between P_{r1} and P_{r2}, as it represents the railhead's width. In practice, the rail head width may vary at a small margin due to wear. Rule 5 constrains the difference of x-coordinates between P_w and P_{r1}, which indicates the possible maximum lateral movement of the wheel in relative to the rail. It can be estimated by the maximum instantaneous lateral acceleration a_{ymax} of the wheel in the sample period of the camera. For simplification, a_{ymax} is statistically estimated as a constant value derived from the field measurement data.

3.4 Image corruption for data augmentation

As the CV system is exposed to a harsh railway environment, a solid housing and an air cleaning system are tailored to protect and clean the camera lenses. Apart from this, we propose a data augmentation procedure during DL model training to enhance the model robustness against possible image corruptions. Taking advantage of previous studies on image corruption [41,42], the relevant corruption types in Figure 6 are modeled. For a given image $I \in \mathcal{R}^{N\times N}$, $I(x,y)$ in the range $(0,255)$ denotes the original pixel intensity at the position (x,y). Gaussian noise may arise during optical sensing. The intensity function of the corrupted image $I_{gn}(x,y)$ injected with Gaussian noise is given by

$$I_{gn}(\mathrm{x},\mathrm{y}) = I(x,y)/255 + c \cdot p \tag{4}$$

$$I_{gn}(\mathrm{x},\mathrm{y}) = \begin{cases} 0 & if\ I_{gn}(\mathrm{x},\mathrm{y}) < 0 \\ I_{gn}(\mathrm{x},\mathrm{y}) \cdot 255 & if\ 0 \le I_{gn}(\mathrm{x},\mathrm{y}) \le 255 \\ 255 & if\ I_{gn}(\mathrm{x},\mathrm{y}) > 255 \end{cases} \tag{5}$$

where c is a settable scale representing the severe level and p is the Gaussian distribution.

Shot noise could occur during photon counting in optical systems. The intensity function of the corrupted image $I_{gn}(x,y)$ injected with shot noise is given by

$$I_{sn}(\mathrm{x},\mathrm{y}) = \mathrm{f}[c \cdot I(x,y)/255]/c \tag{6}$$

$$I_{sn}(\mathrm{x},\mathrm{y}) = \begin{cases} 0 & if\ I_{sn}(\mathrm{x},\mathrm{y}) < 0 \\ I_{sn}(\mathrm{x},\mathrm{y}) \cdot 255 & if\ 0 \le I_{sn}(\mathrm{x},\mathrm{y}) \le 255 \\ 255 & if\ I_{sn}(\mathrm{x},\mathrm{y}) \end{cases} \tag{7}$$

where f is subject to the Poisson distribution.

The modeled impulsive noise refers to salt-and-pepper noise which could originate from sharp and sudden disturbances in the imaging process. The intensity function of the corrupted image $I_{in}(x, y)$ injected with impulsive noise is given by

$$I_{in}(\mathrm{x},\mathrm{y}) = \begin{cases} 0 & if\ c \cdot \alpha/2 \\ I(x,y) & if\ 1 - c \cdot \alpha \\ 255 & if\ c \cdot \alpha/2 \end{cases} \tag{8}$$

where α is the probability that a pixel is altered.

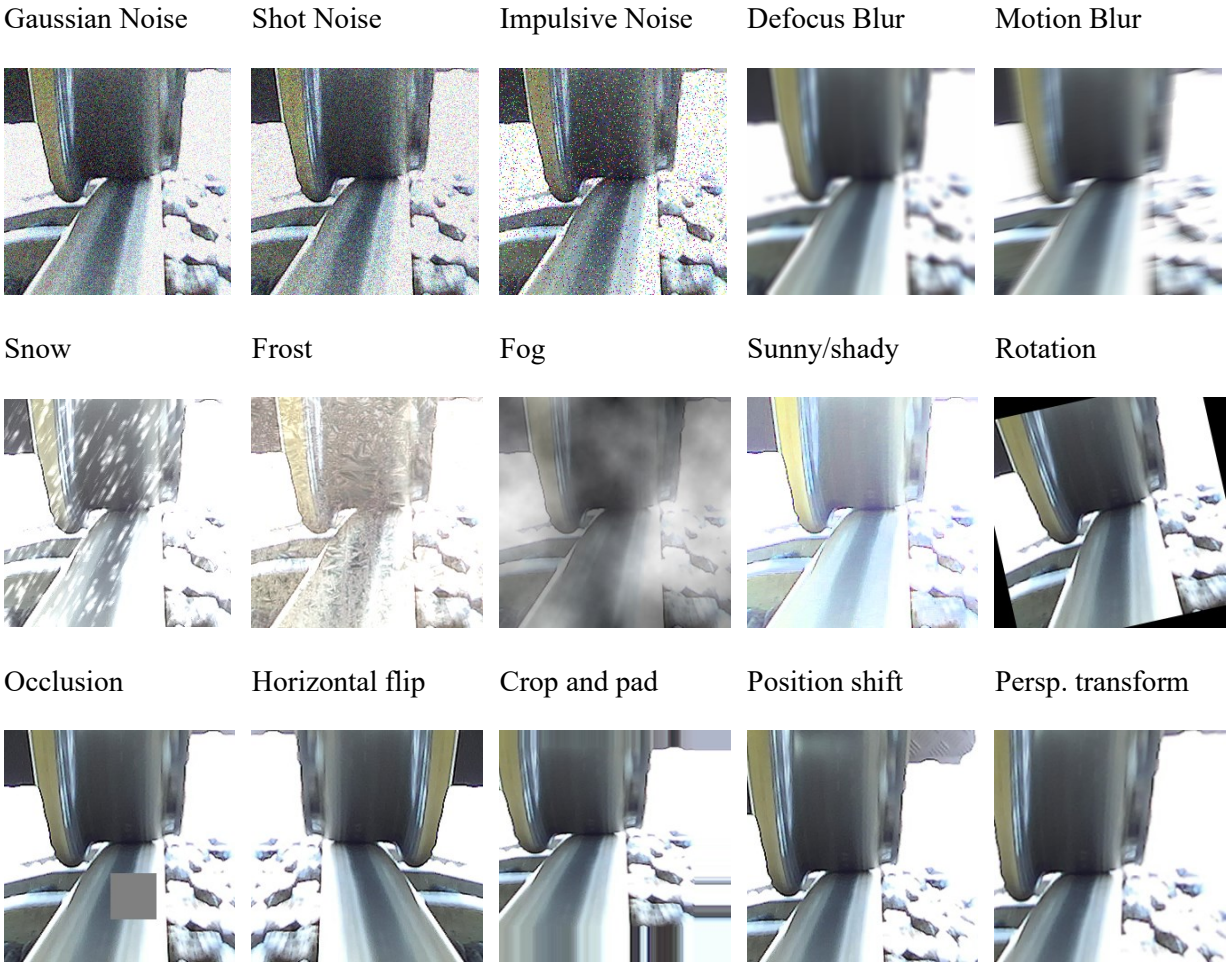


Figure 6. Synthetic images for data augmentation

Defocus blur is when the image is out of focus, which is caused by the fact that the camera integrates the light over areas during sensing. Blur is commonly modeled by convolution of the original image with a uniform point spread function (PSF). The defocus-blurred image I_{db} is given by

$$I_{db} = I * K \tag{9}$$

$$K(x,y) = \begin{cases} 0 & if\ \sqrt{x^2 + y^2} < r \\ 1/\pi r^2 & if\ \sqrt{x^2 + y^2} \geq r \end{cases} \tag{10}$$

where K is the parametric PSF for defocus blur and r is the radius parameter of K and linearly correlated to the severe level c.

Motion blur occurs when the vehicle is excited by large track/rail irregularities. The linear-motion-blurred image I_{mb} is given by

$$I_{mb} = I * K \tag{11}$$

$$K(x,y) = \begin{cases} 1/r & if\ 0 \le x \le r \\ 0 & otherwise \end{cases} \tag{12}$$

where K is the parametric PSF for linear motion blur and r denotes the extent of the motion blur, relying on the severe level c.

In addition, several weather conditions are modeled. Snowy scenes are generated by randomly adding white motion-blurred particles and whitening the entire image. The image with frost is an overlay of the original image and several template images of frosted glass. Fog is modeled by plasma fractal using the diamond-square algorithm. Sunny/shady effect is simulated by increasing/decreasing the brightness of the original image, where the pixel intensity of the first channel in the HLS color space of the image is altered. Furthermore, several common augmentation techniques are applied, such as horizontal flip, rotation and occlusion. In addition, we mimic the images taken at different camera positions and orientations. For each original 1920×1080 image, we randomly crop the 256×256 RoI at different positions. Afterward, point perspective transformation is applied to simulate the variations of the camera's orientation.

4. Experiments and results

4.1 Field tests and datasets

We conducted three field tests under different operational conditions in Italy and Lithuania. The first two tests were performed for data collection and algorithm development at low running speeds. Afterward, the last test was carried out as the validation test at regular operating speeds (up to 100 km/h). In Italy, the prototype of the CV system has been installed on the bogie frame of Aldebaran 2.0, which is the newest track recording coach of Rete Ferroviaria Italiana (RFI, i.e. Italian infrastructure manager) equipped with a commercial TG measurement system, as shown in Figure 7. (a). The first test consisted of several runs within the workshop plant in Catanzaro Lido on both straight and curved track sections. The curved track sections correspond to two switches with a curve radius of 170 m and a tangent of 0.12. During the field test, the Aldebaran 2.0 coach was driven by a locomotive at low speeds (between 2 and 10 km/h). We test different conditions, i.e. two lateral positions of the camera with respect to the wheel and four camera configurations for different resolutions and sample rates. The video data from 3 test runs are used for model training, while 3 test runs are used for testing.

In Lithuania, the second test was performed on the mainline in the vicinity of Vilnius. Two CV systems were installed on the bogie frame of a track recording coach operated by Lithuanian Railways, see Figure 7. (b). The videos for both wheels were recorded simultaneously. Two forward runs at speeds of around 20 km/h and one backwards run at

lower speeds were conducted. The camera setting remains unchanged during the test runs. One forward run is used for training, while the other data is used for testing.

The last validation test was performed in Italy on the same vehicle type of Aldebaran 2.0. The vehicle has been operated in regular operating conditions up to 100 km/h between Rome and Pisa for two days.

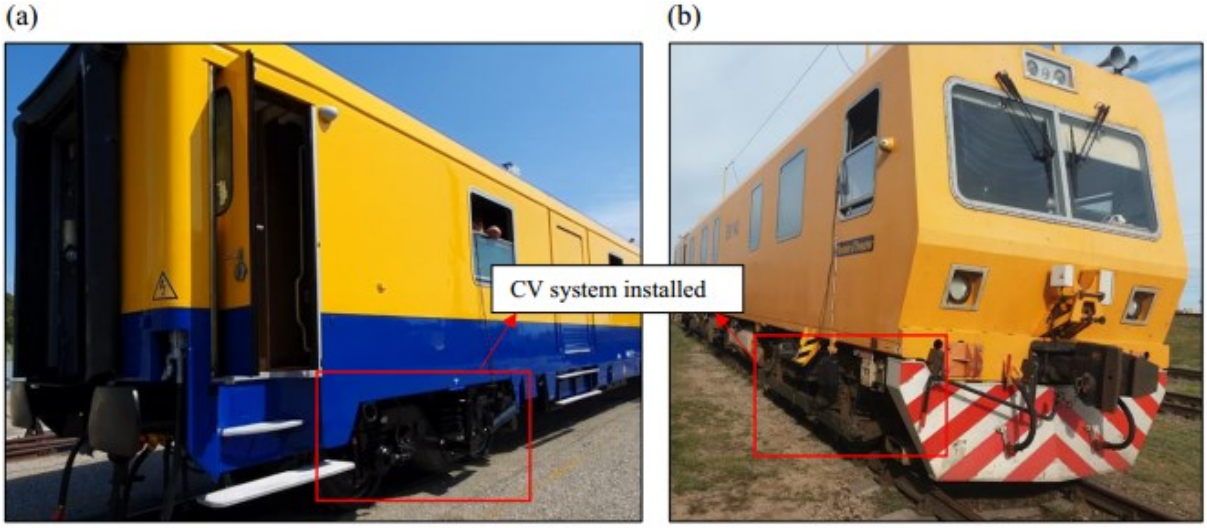


Figure 7. (a) CV system installed on a track recording coach in Italy; (b) CV system installed on a track recording coach in Lithuania

The algorithm development is based on the data obtained in the first two tests. As the video data was sampled at 30 fps and the vehicle ran at low speeds, there are a large number of duplicate frames in the video. To build the dataset, we select one image per 30 frames from the video data collected in Lithuania, while one image per 60 frames from the video data collected in Italy. Other images originate from static tests at other locations and a relevant Youtube video [43]. In static tests, the same ZED2 stereo camera was used for image capture. The images of different bogies standing on the track were obtained, examples of which are shown in Figure 14 in Annex I. The Youtube video was filmed by a GoPro camera during a regular railway operation. The video frames were extracted as shown in Figure 15 in Annex I. The defined virtual points were manually annotated on the original images of 1920×1080 pixels. The coordinates of the labeled points are the ground truth for CNN training. We have 767 annotated images in total. In order to increase the amount of the annotated data, we generate five 256×256 images of RoI cropped at different positions on each original image. In this way, we have 3835 labeled images. They are split into a training dataset, a validation dataset and a test dataset with the ratio of 6/2/2, namely 2301 images for training and 767 images for validation and testing respectively. We conduct extensive experiments to validate the proposed approach as follows.

4.2 Training and evaluation of YOLOv3-tiny for calibration

In YOLOv3-tiny, we merely modify the YOLO layers for RoI detection, while the first 13 layers, i.e. Darknet, have not been changed. This allows us to transfer the pre-trained weights of Darknet to the modified YOLOv3-tiny. In this way, the model for RoI detection can be trained with fewer annotated images. Figure 8 presents the pipeline for training and evaluation

of YOLOv3-tiny on our datasets. YOLOv3-tiny is first pre-trained on the COCO dataset [44], which contains 123287 annotated images in total, incl. 3745 images related to the railway. The learned parameters of Darknet are transferable, while the learned parameters of the YOLO layers are discarded. Our training dataset consists of 800 images from static tests and the Youtube video. The raw 1920×1080 images are resized to the 416×416 ones, fed into YOLOv3-tiny. The pre-trained YOLOv3-tiny is trained with adaptive moment estimation (Adam) for 30 epochs, which is a gradient-descent-based optimization algorithm. Afterward, the trained model of YOLOv3-tiny is evaluated on 767 annotated images for keypoint detection. The evaluation metric is whether the labeled keypoints are inside the predicted bounding box within an image. YOLOv3-tiny has achieved a detection accuracy of 100%.

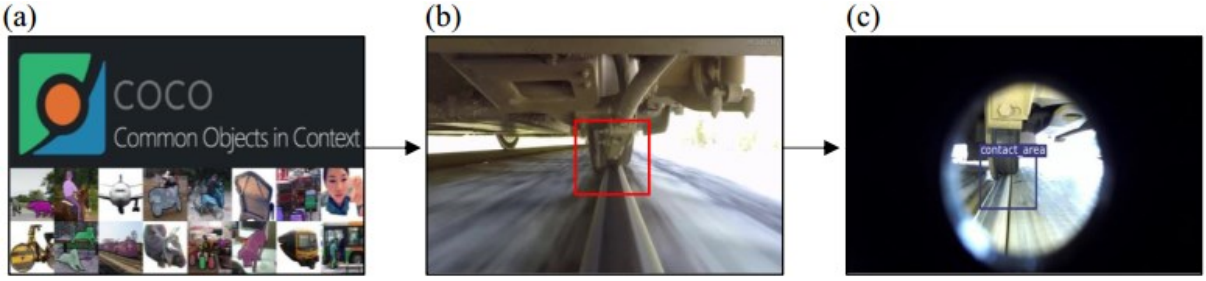


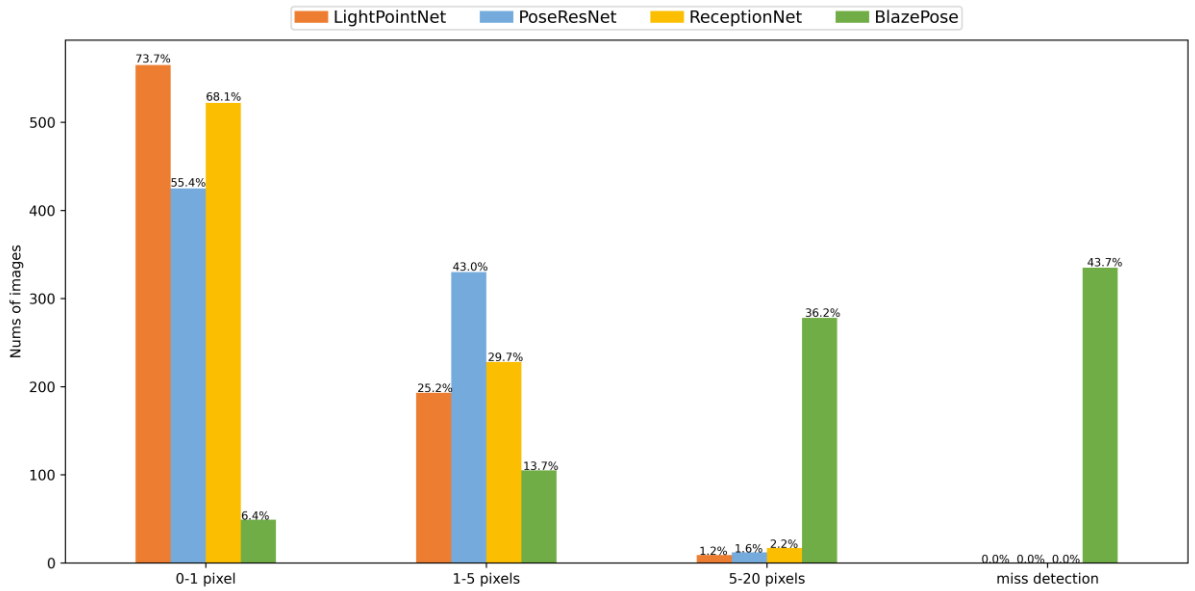
Figure 8. Pipeline for training and evaluation of YOLOv3-tiny: (a) Pre-training on COCO dataset; (b) Fine-tuning on our training dataset; (c) Evaluation on our test dataset

4.3 Training and evaluation of LightPointNet

The evaluation of LightPointNet is conducted in a two-fold way. Firstly, LightPointNet is trained and evaluated on the individual images randomly selected and cropped from the video frames. The evaluation metric is the deviation in terms of pixels between the ground-truth and the predicted x-coordinate of P_w and P_{r1}. We compare the evaluation results of LightPointNet with those of the baselines. Secondly, the trained LightPointNet is applied on the video sequences. The evaluation metric is defined as the count that detects the predictions exceeding the thresholds TH_0 in the rule engine.

For comparison, we implement three DL-based baselines, i.e. PoseResNet [28], ReceptionNet [30] and BlazePose [31], as well as a method using conventional CV techniques developed in our previous work [32]. PoseResNet is the representative of the detection-based HPE network. ReceptionNet is similar to our solution, using softmax for coordinate regression. The main difference lies in the encoder and decoder for feature extraction. BlazePose is a hybrid solution, requiring two-step training. In the first step, the network is trained on the heatmap branch as the common detection-based HPE networks do. In the second step, the weights in the heatmap branch are frozen, while the regression branch is activated for training. In the inference stage, the regression branch is applied to directly output the point coordinates, which avoids the quantization errors and reduces the computation complexity. Our previous method mainly uses template matching to track the wheel flange and line tracking to track the rail line. More details can be found in [32]. However, this method is only applicable to track the points in the video sequences by manually selecting the reference template and line on the first frame. It is not able to automatically detect the wheel and rail.

(a)



(b)

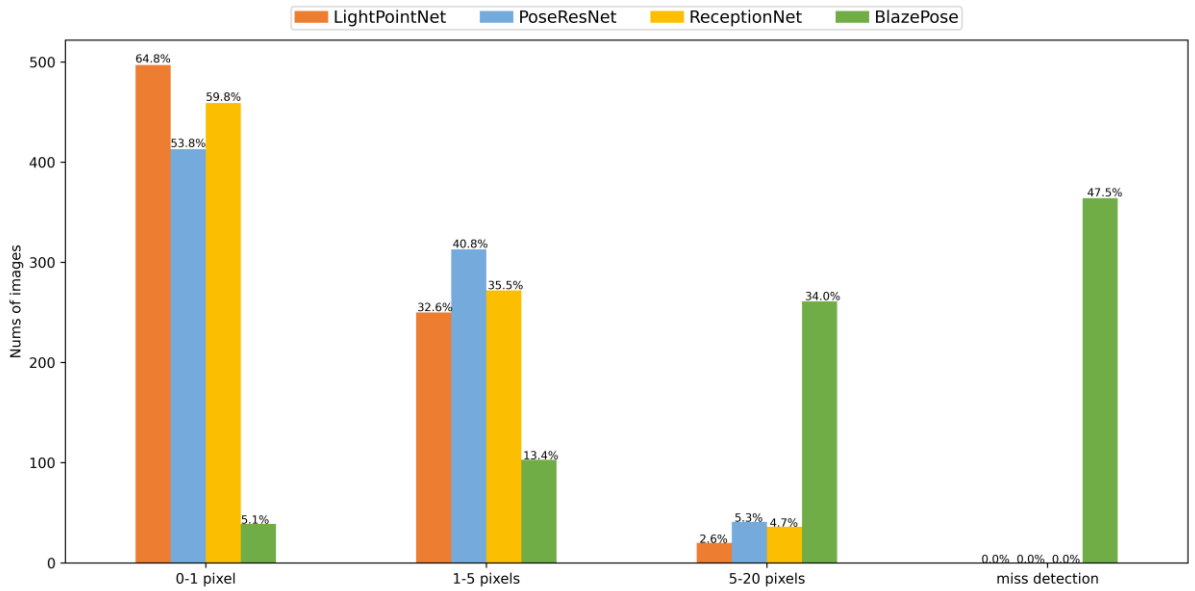


Figure 9. (a) detection errors (in pixels) for the wheel reference point; (b) detection errors (in pixels) for the rail reference point

In the first evaluation experiment, LightPointNet is compared with PoseResNet, ReceptionNet and BlazePose. The DL networks are trained from scratch on our training dataset (incl. 2301 256×256 images) and evaluated on the testing dataset (incl. 767 images). The validation dataset is used to prevent overfitting by evaluating the temporary model trained in each epoch during the training process. Adam with the multistage learning rate is applied to minimize the integral loss over 100 epochs. We repeat the training process using different random seeds five times and select the best models for the comparison. The main reason is that CNN

learnable weights are randomized at initialization and learning samples are randomized. We get models that perform differently with the same training conditions. The details of implementation and experiment settings can be found in our code repository.

The evaluation metric is the deviation in terms of pixels between the ground-truth and the predicted x-coordinate of P_w and P_{r1}. We divide detection errors into four groups. "0-1 pixel" means either no error or an error of 1 pixel. In our case, 1 pixel means 0.78 mm (depending on the image resolution as well as the distance between the camera and the wheel). A small error of 1-5 pixels is tolerable. A large error of 5-20 pixels is unacceptable. An error with more than 20 pixels is defined as "miss detection". Figure 9 shows the detection errors for the wheel and rail reference points. Overall, LightPointNet achieves the best performance among the baselines. The error rate for the wheel reference point is lower than that for the rail reference point. Comparing LightPointNet with the baselines, we observe at first glance that BlazePose fails to accurately detect the wheel and rail reference point using the regression branch. According to our experiment results, the heatmap branch of BlazePose can achieve comparable performance as PoseResNet does. Switching the heatmap branch to the regression branch induces significant performance loss. Second, LightPointNet and ReceptionNet, which involve integral regression from heatmaps, deliver fewer detection errors than PoseResNet, which directly outputs heatmaps and thus induces the quantization errors. Third, LightPointNet overperforms ReceptionNet. The reason may lie in the different architectures of the encoder and decoder.

Furthermore, we evaluate LightPointNet and our previous method [32] on video sequences. As the ground-truth points have not been manually labeled on video sequences, a rigorous validation comparing the prediction with the ground truth cannot be performed. In this case, the rule engine is used for the evaluation. The evaluation metric is defined as the count that detects the predictions exceeding the thresholds in the rule engine. We select several video sequences representing different track layouts. Table 4 compares the results of two methods, where the baseline refers to our previous method. The baseline and LightPointNet can achieve comparable results on the videos obtained in Lithuania. However, LightPointNet significantly overperforms the baseline in the first test in Italy. In these videos, brightness is much lower and the backgrounds are more complicated than those in Lithuania. The baseline almost fails to track the rail lines in the videos and thus fails to deliver reliable results, although much effort has been paid to tune the relevant parameters for pre-processing. In the last validation test, a part of the video sequences, named "Italy2_highspeed", has been randomly selected, where the vehicle has reached the maximum speed of 100 km/h. The illumination has been improved during this test. LightPointNet delivers 3.51% miss detection rate, while the baseline has 18.49%, which is much better than that in the first test. A high speed results in motion blur of the complex backgrounds, which makes it easier to track the wheel temple and rail line. It is worth noting that the current generalization ability of the LightPointNet model trained on the aforementioned training dataset was not sufficient to achieve a high detection accuracy on a new test dataset. We randomly labeled 351 images in the validation test for model fine-tuning. To prevent data leakage, we avoided using any images from the test video sequence "Italy2_highspeed". The number of the labeled images is merely 0.1% of the total frames obtained in this validation test.

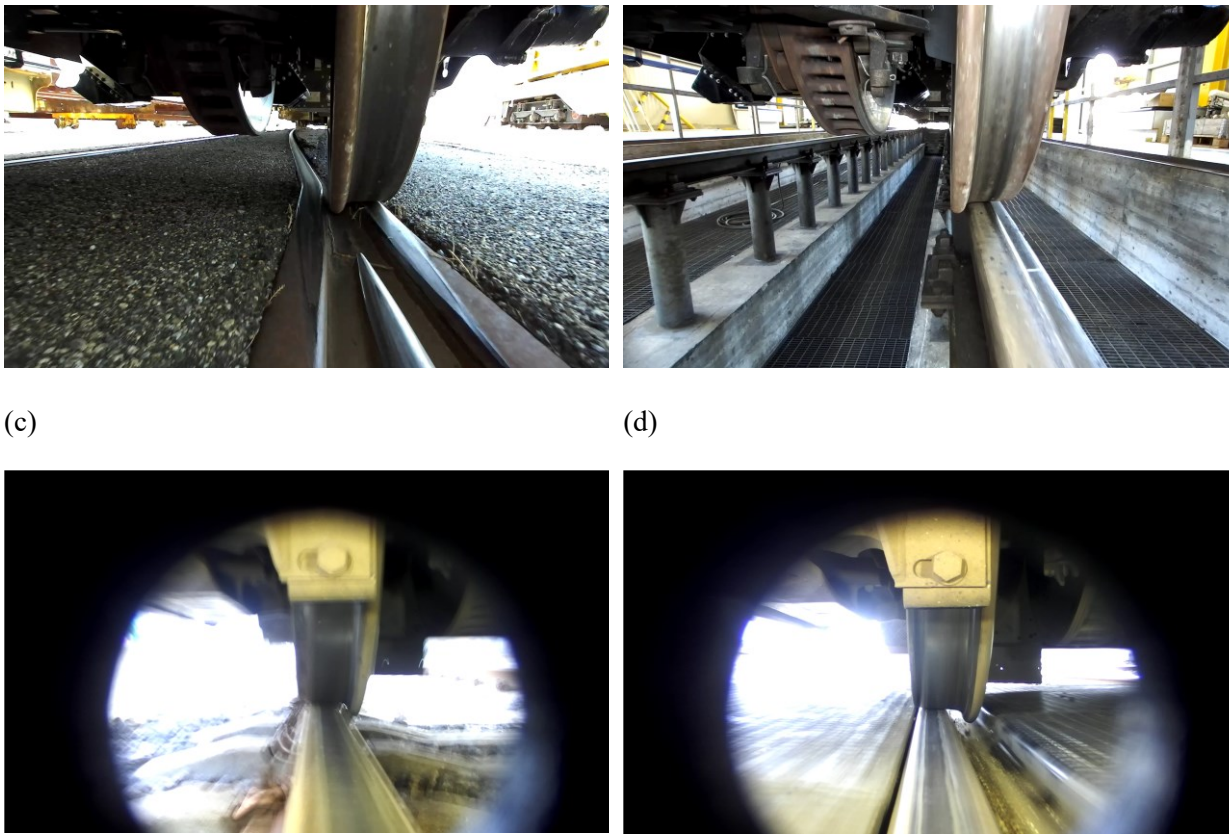
Table 4. Test results of LightPointNet and the baseline on the representative video sequences

Video sequence	Frame number	Model	Implausible detection exceeding TH_0	Percentage of Implausible detection
Lithuania_forward_straight	4997	LightPointNet	207	4.45%
		Baseline	245	4.90%
Lithuania_backward_straight	7996	LightPointNet	84	1.05%
		Baseline	78	0.98%
Italy_curve	2586	LightPointNet	200	7.73%
		Baseline	2263	87.5%
Italy_swtich	492	LightPointNet	50	10.16%
		Baseline	344	69.92%
Italy_straight	4024	LightPointNet	233	5.79%
		Baseline	3144	78.13%
Italy2_highspeed	14082	LightPointNet	495	3.51%
		Baseline	2604	18.49%

4.4 Evaluation of rule engine

In order to evaluate the effectiveness of the proposed rule engine, we evaluate our approach with and without the rule engine on the video sequences. In the approach without the rule engine, LightPointNet is applied for each video frame and directly outputs the predicted coordinates as the final results. The percentage of implausible detection has been shown in Table 4. In the approach with the rule engine, LightPointNet's outputs are fed into the rule engine. The corner cases are detected and the corresponding implausible results are discarded or corrected. Figure 10 illustrates several typical corner cases where LightPointNet fails to deliver a reliable detection result. The corner cases may have specific scenes in backgrounds like workshops and platforms. The switch and crossing zones have a unique track layout that may mislead the detector. A wheel bounce results in a sudden change of the y-coordinate of P_w and trigger the rule engine. It may also cause miss detection of LightPointNet due to the strong motion blur. Such corner cases will be added to the training dataset for further model training.

(a) (b)



(c) (d)

Figure 10. Corner cases detected by the rule engine: (a) switch & crossing zone; (b) workshop zone; (c) wheel bounce; (d) platform zone

For evaluation of the correction mechanism, the trajectory of the LDWR over the frames is displayed. Figure 11 shows the trajectory with and without the rule engine calculated on the video sequence “Italy_straight” containing 4024 frames. The correction mechanism based on the rule engine uses the information of two adjacent frames to remove the coordinates' unreliable sudden changes, as shown by red impulses in Figure 11. Nevertheless, tracking the actual lateral movement of the wheel has not been affected. For instance, a sizeable lateral wheel movement occurs between the 2750th and 2950th frame is visible by the blue line in Figure 11. However, we observe that the predicted coordinates' small-scale turbulence cannot be smoothed by the rule engine. The data fusion with the corresponding wheel accelerations may cover this gap.

4.5 Evaluation of the entire approach

The entire algorithm is executed on the Nvidia Jetson TX2 platform in real-time. The tracking results on two video sequences are recorded as the demonstration videos. As the points are hardly visible on the raw 1920×1080 images, the demo videos merely display the 256×256 RoI, which is automatically detected by YOLOv3-tiny at the first step of the proposed approach. Figure 12 shows the tracked points P_w and P_{r1} on the wheel flange and the rail edge in the RoI for the three field tests. These two points are used for the calculation of the lateral

wheel displacement. P_{r2} is on the other side of the rail edge and provides the geometric information for the rule engine. It is not displayed on the demo videos.

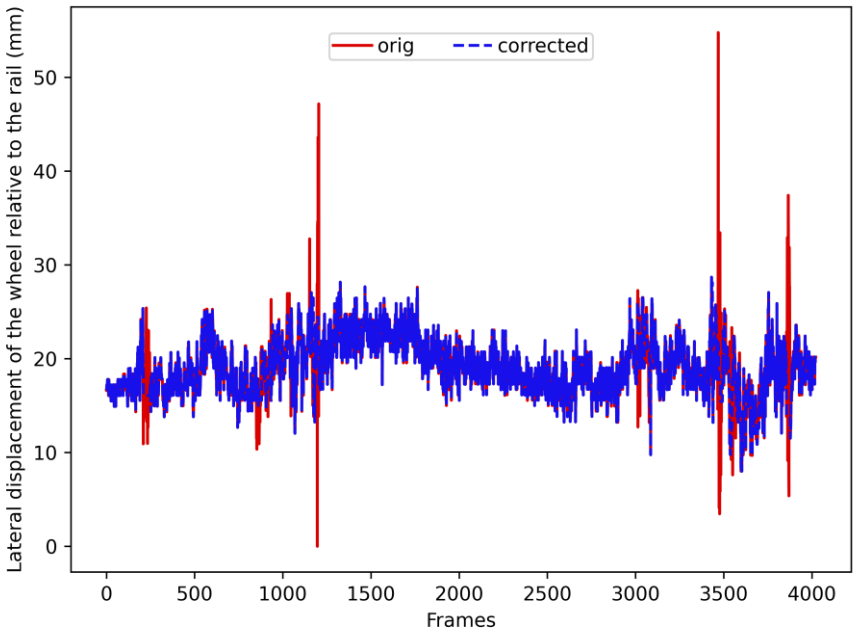


Figure 11. Comparison of the measured lateral displacement of the wheel relative to the rail with (blue) and without (red) the rule engine based correction mechanism

Figure 12. Real-time lateral displacement measurement of the wheel on the rail to support track geometry monitoring (screenshot from the demo videos *https://youtu.be/it21cE87LCM*, *https://youtu.be/Nc1bkQdkkSM* and https://youtu.be/K5HnUixyGxo

Wheel's lateral motion has been successfully tracked by tracking the virtual points. However, we observe slight shifts of the detected virtual points in the lateral direction, although the wheel's actual position does not change. It results in sudden changes of lateral displacement in several millimeters. This indicates a measurement uncertainty up to 2 mm based on our observation, which stems from point tracking and displacement calculation. In our previous study [32], the displacement calculation method based on two reference points have been tested in a laboratory, where the stereo camera was placed at different distances and view angles with respect to a standard gauge block. Two reference points were manually selected on the gauge block's left and right edge to calculate its width. The measurement uncertainty (i.e. in the form of standard deviation) has been determined as 0.4 mm. Therefore, we

conclude that the point detection of LightPointNet induces the main uncertainty. On the one hand, this is due to the model's performance limitation trained on the currently collected training data. On the other hand, the uncertainty could originate from label noise, which occurs when we manually annotate the virtual points as the ground truth. Due to the complex background, variable illumination conditions, and labeling tool restrictions, the accurate point position on the wheel flange and the railhead edge can hardly be determined. An annotation deviation of several pixels on a similar video frame is quite common. For the 1920×1080 resolution and the distance between the camera and the wheel, one pixel refers to 0.78 mm. Therefore, a measurement uncertainty of 2 mm due to manual annotation is understandable and can be hardly avoided. A possible solution is to increase the image resolution of RoI. In future work, we consider replacing the current camera with the one having a narrower field of view and closer focusing distance.

4.6 Computational complexity and real-time capability

CNN's computational complexity can be theoretically estimated by the number of parameters and floating-point operations (FLOPs). A regular convolution layer consists of N convolutional filters, each of which is parameterized by the weights $W \in \mathcal{R}^{K\times K}$, where K denotes the width. When it takes a feature map $F_{in} \in \mathcal{R}^{D\times D\times C}$ as the input and outputs a feature map $F_{out} \in \mathcal{R}^{D\times D\times N}$, the total parameters P_c and FLOPs are given by eq. (14) and (15), where the parameter number of bias and the accumulated operation are neglected.

$$P_c = K \times K \times C \times N \tag{13}$$

$$FLOPs_c = 2MAC = 2 \times (K \times K \times C \times N \times D \times D) \tag{14}$$

In LightPointNet, regular $K \times K$ convolution is replaced with the combination of 1×1 pointwise convolution and $K \times K$ depthwise convolution, which is named as the depthwise separable convolution (DSC). Its parameter numbers P_{DSC} and FLOP_{DSC} are significantly reduced, given by:

$$P_{DSC} = K \times K \times C \tag{15}$$

$$FLOPs_{DSC} = 2 \times (K \times K \times C \times D \times D + N \times C \times D \times D) \tag{16}$$

The reduction ratio in parameter r_P and in operation r_{FLOPs} are given by:

$$r_P = \frac{1}{N} \tag{17}$$

$$r_{FLOPs} = \frac{1}{N} + \frac{1}{K^2} \tag{18}$$

The computational complexity of a conventional image processing algorithm can be hardly accurately measured. In the baseline, we mainly use a template matching algorithm for wheel tracking and a line tracking algorithm provided in the Visual Servoing Platform library for rail tracking. Its theoretical computational complexity can hardly be calculated. For a more accurate comparison, the actual time consumption, i.e. latency, is measured for each algorithm. The latency relies on the hardware and software platform. In our application, we implement the DL models in PyTorch 1.9 (which is an open-source DL framework) and deploy the models on the edge computer Nvidia Jetson TX2 for inference. The baseline is implemented with OpenCV libraries. We measure the time consumption on this platform and

calculate frame per second (FPS) averaged over the testing video sequences as the evaluation metric. This allows the comparison between DL models and the baseline.

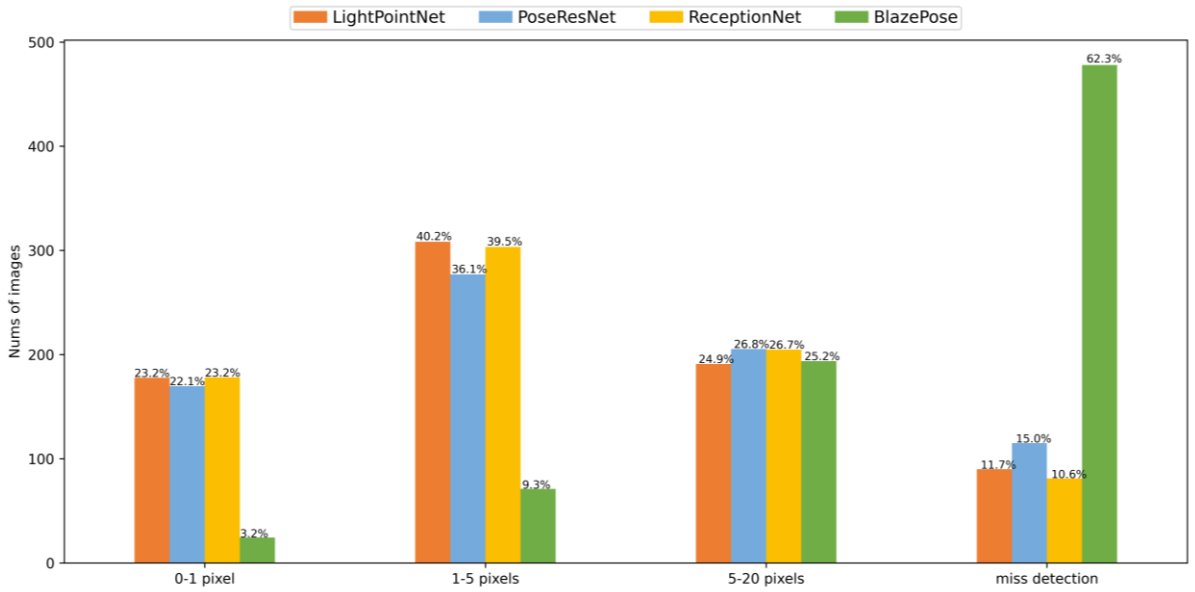
LightPointNet uses several lightweight measures to reduce the number of parameters and FLOPs while maintaining network performance, incl. using filters of small sizes, using DSC, using SE modules and linear bottleneck structure. More details of the lightweight measures can be found in our previous study [46]. We compare our LightPointNet with the baselines to show the effectiveness of lightweight. Table 5 shows the computational complexity of different models with a batch size of 16. Parameters and FLOPs of DL models are measured by a third-party tool. The third row indicates the latency in fps of the original DL models implemented in Pytorch. The fourth row indicates the latency of the DL models in the format of Nvidia TensorRT, which will be explained later. At first glance, we find that the baseline has a larger latency than our LightPointNet in terms of FPS. However, both are slower than the real-time requirement (i.e. 30 FPS). Comparing LightPointNet with PoseResnet and BlazePose, the latency of LightPointNet is slightly less than that of PoseResnet and BlazePose, although FLOPs of LightPointNet are much lower. It indicates that the platform-dependent latency is also much affected by other factors apart from FLOPs. In terms of parameters, LightPointNet has almost 12-times fewer parameters than PoseResnet, which means much less memory usage. A Pytorch model of LightPointNet occupies 11 MB, whereas a PyTorch model of PoseResnet occupies 130 MB.

Table 5. Computational complexity of LightPointNet and the baselines (M for million, G for Giga)

Results	LightPointNet	PoseResnet	ReceptionNet	BlazePose	Baseline
Parameters	2.86 M	33.99 M	5.97 M	34.00 M	-
FLOPs	5.51 G	12.90 G	6.67 G	17.39 G	-
FPS	20	16	19	15	18
FPS (TensorRT)	39	26	35	23	-

We observe that none of the PyTorch models has a real-time ability on the target platform. To further reduce the latency, we transform the PyTorch models into the format of TensorRT, which speeds up the inference of a DL model on Nvidia's GPUs. TensorRT forces the models for low precision inference. The learned parameters of weights and biases within a NN are typically represented in the format of float32, occupying 32 bits. TensorRT transforms these parameters into the 8-bit representation. This dramatically accelerates the inference process by sacrificing little accuracy. Furthermore, TensorRT optimizes the computation graph of a NN to accelerate the computation. More details can be found in [47]. The last row of Table 5 shows the latency of the DL models in the TensorRT format. LightPointNet and ReceptionNet can satisfy the real-time requirement.

(a)



(b)

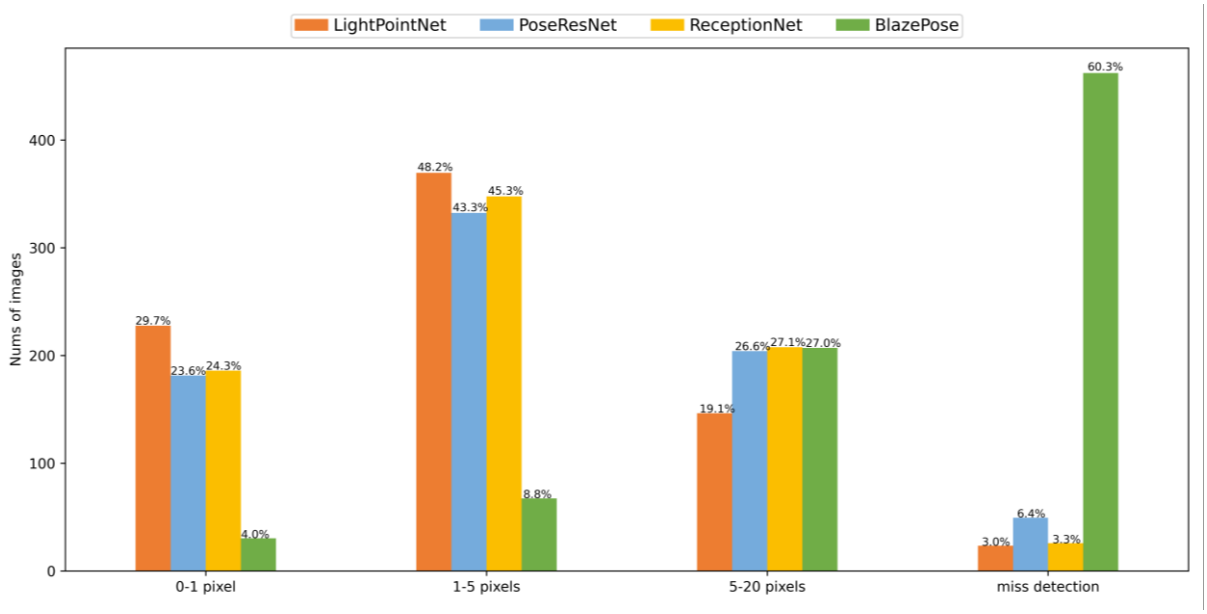


Figure 13. detection errors (in pixels) averaged over three reference points (a) when the models are trained on the cleaning training dataset and tested on the corrupted test dataset; (b) when the models are trained on the augmented training dataset and tested on the corrupted test dataset

4.7 Data augmentation for robustness enhancement

The model robustness plays an essential role in harsh outdoor conditions. The degradation or interference of sensors may result in image noise. Large vibrations induced by severe track/rail irregularities may result in image blur. Dirt and dust on camera lenses may result in occlusions in images. Varying weather conditions may result in variations of intensity distributions within images. Based on these types of image corruption, we build a corrupted testing dataset. Each image from the original test dataset containing 767 images is augmented

with a corruption method randomly selected from the ones shown in Figure 6. Each corruption method contains a severity scale c, which controls the severity of the corruption. The scale c is randomly set in the range from 1 to 3. First, we investigate the model robustness against image corruption. LightPointNet and the baselines are trained on the clean training dataset without corrupted images and tested on the corrupted test dataset. Comparing Figure 13 (a) with Figure 9, we observe that the model performance of LightPointNet dramatically drops from around 70% to 23.2% of "0-1 pixel" detection errors. ReceptionNet achieves similar performance as LightPointNet. This indicates the insufficient robustness of all the DL models against the potential image corruptions in harsh outdoor conditions. Afterward, we investigate the effect of data augmentation by repeating the corruption process on the training dataset as data augmentation. The DL models are trained on the augmented training dataset and tested on the corrupted test dataset. Figure 13 (b) shows data augmentation largely reduces the miss detection rate of LightPointNet, PoseResNet and ReceptionNet. In particular for LightPointNet, the rate of the "0-1 pixel" errors is improved by 6.5%. On the one hand, this result proves the effect of data augmentation and superiority of LightPointNet. On the other hand, data augmentation alone is not sufficient to ensure robustness, since the improvement is merely a drop in the bucket in comparison to performance loss induced by image corruption. Other generalization measures such as domain generalization should be combined.

5. Conclusions

The virtual point tracking approach was proposed to tackle the issue of dynamic displacement measurement with varying and noisy backgrounds. The entire approach has been validated and demonstrated for lateral displacement measurement of the wheelsets on the rail tracks, in order to support track geometry monitoring on in-service rail vehicles. The feasibility of the proposed solution has been demonstrated in the field tests under regular railway operating conditions. It can satisfy the real-time processing requirement and achieve a measurement uncertainty of up to 2 mm. The core component of our approach is LightPointNet for point detection, which is a lightweight CNN architecture. It outperforms several baselines using either conventional image processing techniques or other deep learning networks. One unsolved issue in this work is the robustness against harsh outdoor conditions and the generalization ability, which are the common issue for arbitrary machine learning methods. They will be addressed in our future work.

Declaration of Competing Interest

On behalf of all authors, the corresponding authors state that there is no conflict of interest.

Acknowledgments

The research is funded by the EU Shift2Rail project Assets4Rail (Grand number: 826250) under Horizon 2020 Framework Programme.

References

[1] Y. Xu, J.M.W. Brownjohn, Review of machine-vision based methodologies for displacement measurement in civil structures, Journal of Civil Structural Health Monitoring. 8 (2017) 91–110. doi:10.1007/s13349-017-0261-4.

[2] D. Zhan, D. Jing, M. Wu, D. Zhang, L. Yu, T. Chen, An Accurate and Efficient Vision Measurement Approach for Railway Catenary Geometry Parameters, IEEE Transactions on Instrumentation and Measurement. 67 (2018) 2841–2853. doi:10.1109/tim.2018.2830862.

[3] Z.-W. Li, Y.-L. He, X.-Z. Liu, Y.-L. Zhou, Long-Term Monitoring for Track Slab in High-Speed Rail via Vision Sensing, IEEE Access. 8 (2020) 156043–156052. doi:10.1109/access.2020.3017125.

[4] J. Baqersad, P. Poozesh, C. Niezrecki, P. Avitabile, Photogrammetry and optical methods in structural dynamics – A review, Mechanical Systems and Signal Processing. 86 (2017) 17–34. doi:10.1016/j.ymssp.2016.02.011.

[5] C.-Z. Dong, F.N. Catbas, A review of computer vision–based structural health monitoring at local and global levels, Structural Health Monitoring 20 (2021) 692–743. https://doi.org/10.1177/1475921720935585.

[6] S. Wang, H. Wang, Y. Zhou, J. Liu, P. Dai, X. Du, M. Abdel Wahab, Automatic laser profile recognition and fast tracking for structured light measurement using deep learning and template matching, Measurement 169 (2021) 108362. https://doi.org/10.1016/j.measurement.2020.108362.

[7] T. Jiang, G.T. Frøseth, A. Rønnquist, E. Fagerholt, A robust line-tracking photogrammetry method for uplift measurements of railway catenary systems in noisy backgrounds, Mechanical Systems and Signal Processing. 144 (2020) 106888. doi:10.1016/j.ymssp.2020.106888.

[8] European Committee for standardization. EN 13848-1: Railway applications - Track - Track geometry quality - Part 1: Characterization of track geometry. (2019)

[9] V. Skrickij, D. Shi, M. Palinko, L. Rizzetto, G. Bureika, Wheel-rail transversal position monitoring technologies. Technical Report Deliverable D8.1, EU Horizon 2020 project Assets4Rail. (2019). http://www.assets4rail.eu/results-publications/. (accessed December 7, 2020).

[10] B. Ripke et al., Report on track/switch parameters and problem zones, technical report Deliverable D4.1 of the IN2SMART project, https://projects.shift2rail.org/s2r_ip3_n.aspx?p=IN2SMART (accessed December 3, 2020).

[11] I. Soleimanmeigouni, A. Ahmadi, H. Khajehei, A. Nissen, Investigation of the effect of the inspection intervals on the track geometry condition. Structure and Infrastructure Engineering. 16 (2020) 1138–1146. doi: 10.1080/15732479.2019.1687528.

[12] P. Weston, C. Roberts, G. Yeo, E. Stewart, Perspectives on railway track geometry condition monitoring from in-service railway vehicles. Vehicle System Dynamics. 53(2015) 1063-1091. doi:10.1080/00423114.2015.1034730

[13] H. True, L.E. Christiansen, Why is it so difficult to determine the lateral Position of the Rails by a Measurement of the Motion of an Axle on a moving Vehicle? Proceedings of First International Conference on Rail Transportation. (2017)

[14] A.D. Rosa, S. Alfi, S. Bruni, Estimation of lateral and cross alignment in a railway track based on vehicle dynamics measurements, Mechanical Systems and Signal Processing. 116 (2019) 606–623. doi:10.1016/j.ymssp.2018.06.041.

[15] A.D. Rosa, R. Kulkarni, A. Qazizadeh, M. Berg, E.D. Gialleonardo, A. Facchinetti, et al., monitoring of lateral and cross level track geometry irregularities through onboard vehicle dynamics measurements using machine learning classification algorithms, Proceedings of the

Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit. (2020) 095440972090664. doi:10.1177/0954409720906649.

[16] J. Sun, E. Meli, W. Cai, H. Gao, M. Chi, A. Rindi, S, Liang, A signal analysis based hunting instability detection methodology for high-speed railway vehicles, Vehicle System Dynamics. (2020). doi: 10.1080/00423114.2020.1763407

[17] M. Kim, Measurement of the Wheel-rail Relative Displacement using the Image Processing Algorithm for the Active Steering Wheelsets, International Journal of Systems Applications, Engineering & Development 6 (2012)

[18] SET Limited, Laser triangulation sensors measure lateral position of rail bogie wheels, Laser Triangulation Sensors Measure Lateral Position of Rail Bogie Wheels, Engineer Live. https://www.engineerlive.com/content/laser-triangulation-sensors-measure-lateral-position-rail-bogie-wheels (accessed December 5, 2020).

[19] D. Yamamoto, Improvement of method for locating position of wheel/rail contact by means of thermal imaging, Quarterly Report of RTRI (2019)

[20] J. Guo, C. Zhu, Dynamic displacement measurement of large-scale structures based on the Lucas–Kanade template tracking algorithm, Mechanical Systems and Signal Processing. 66-67 (2016) 425–436. doi:10.1016/j.ymssp.2015.06.004.

[21] Y.J. Cha, J.G. Chen, O. Büyüköztürk, Output-only computer vision based damage detection using phase-based optical flow and unscented Kalman filters, Engineering Structures. 132(2017) 300-313. doi: https://doi.org/10.1016/j.engstruct.2016.11.038

[22] C.-Z. Dong, O. Celik, F.N. Catbas, E. OBrien, S. Taylor, A Robust Vision-Based Method for Displacement Measurement under Adverse Environmental Factors Using Spatio-Temporal Context Learning and Taylor Approximation, Sensors (Basel) 19 (2019). https://doi.org/10.3390/s19143197.

[23] R. Yang, S.K. Singh, M. Tavakkoli, N. Amiri, Y. Yang, M.A. Karami, et al., CNN-LSTM deep learning architecture for computer vision-based modal frequency detection, Mechanical Systems and Signal Processing. 144 (2020) 106885. doi:10.1016/j.ymssp.2020.106885.

[24] J. Liu, X. Yang, Learning to See the Vibration: A Neural Network for Vibration Frequency Prediction, Sensors. 18 (2018) 2530. doi:10.3390/s18082530.

[25] Y. Chen, Y. Tian, M. He, Monocular human pose estimation: A survey of deep learning-based methods, Computer Vision and Image Understanding. (2020). https://www.sciencedirect.com/science/article/abs/pii/S1077314219301778 (accessed December 7, 2020).

[26] Microsoft, Azure Kinect body tracking joints, Microsoft Docs. (2019). https://docs.microsoft.com/en-us/azure/kinect-dk/body-joints (accessed December 7, 2020).

[27] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, European conference on computer vision. (2016) 483–499.

[28] B. Xiao, H. Wu, Y. Wei, Simple Baselines for Human Pose Estimation and Tracking, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018) 472-487.

[29] K. Sun, B. Xiao, D. Liu, J. Wang, Deep High-Resolution Representation Learning for Human Pose Estimation, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019): 5686-5696.

[30] D. Luvizon, D. Picard, H. Tabia, 2D/3D Pose Estimation and Action Recognition using Multitask Deep Learning, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (2018) 5137-5146.

[31] V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, M. Grundmann, BlazePose: On-device Real-time Body Pose tracking. arXiv.org. (2020). https://arxiv.org/abs/2006.10204. (accessed December 7, 2020).

[32] V. Skrickij, E. Šabanovič, D. Shi, S. Ricci, L. Rizzetto, G. Bureika, Visual Measurement System for Wheel-Rail Lateral Position Evaluation, Sensors (Basel) 21 (2021). https://doi.org/10.3390/s21041297.

[33] Stereolabs, Datasheet ZED2 Nov 2019 rev6 - Stereolabs, (2019). https://www.stereolabs.com/assets/datasheets/zed2-camera-datasheet.pdf (accessed December 9, 2020).

[34] NVIDIA, NVIDIA Jetson TX2: High Performance AI at the Edge, NVIDIA. (n.d.). https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-tx2/ (accessed December 9, 2020).

[35] J. Redmon, A. Farhadi, YOLOv3: An Incremental Improvement, ArXiv.org. (2018). https://arxiv.org/abs/1804.02767v1 (accessed December 9, 2020).

[36] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, et al., Searching for MobileNetV3, 2019 IEEE/CVF International Conference on Computer Vision (ICCV). (2019). doi:10.1109/iccv.2019.00140.

[37] X. Sun, B. Xiao, F. Wei, S. Liang, Y. Wei, Integral Human Pose Regression, Proceedings of the European Conference on Computer Vision (ECCV), 529-545.(2017)

[38] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, (2015). https://dl.acm.org/doi/10.5555/3045118.3045167 (accessed December 13, 2020).

[39] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016). doi:10.1109/cvpr.2016.90.

[40] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-Excitation Networks, ArXiv.org. (2019). https://arxiv.org/abs/1709.01507 (accessed December 13, 2020).

[41] D. Hendrycks, T. Dietterich, Benchmarking Neural Network Robustness to Common Corruptions and Perturbations, ArXiv.org. (2019). https://arxiv.org/abs/1903.12261 (accessed August 30, 2020).

[42] A.B. Jung, K. Wada, S. Tanaka, C. Reinder, et al. Imgaug, (2020). https://github.com/aleju/imgaug (accessed December 13, 2020)

[43] diiselrong, Train wheelon a rail 2, (2018). https://youtu.be/6oEkVbhT_T8 (accessed December 13, 2020)

[44] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, et al., Microsoft COCO: Common Objects in Context, (2018). https://www.microsoft.com/en-us/research/publication/microsoft-coco-common-objects-in-context/ (accessed December 21, 2020).

[45] T. Asano, N. Katoh, Variants for the Hough transform for line detection, Computational Geometry. 06-04 (1996) 231-252. doi: 10.1016/0925-7721(95)00023-2.

[46] D. Shi, Y. Ye, M. Gillwald, M. Hecht, Designing a lightweight 1D convolutional neural network with Bayesian optimization for wheel flat detection using carbody accelerations, International Journal of Rail Transportation. (2020) 1–31. doi:10.1080/23248378.2020.1795942.

[47] H. Abbasian, J. Park, S. Sharma, S. Rella, Speeding Up Deep Learning Inference Using TensorRT. (2020). https://developer.nvidia.com/blog/speeding-up-deep-learning-inference-using-tensorrt/ (accessed December 13, 2020)

Annex I

Figure 14. Examples of static images taken on different bogies

Figure 15. Examples of images from a YouTube video

7. Discussions

This section discusses the raised research questions. For each research question, the results of the corresponding publication are discussed, the limitations of our work are explained, the future work for the potential improvement is proposed. In some cases, supplementary experiments are performed. Apart from this, the potential uses of our methods for other purposes are envisaged.

7.1. Can wheel flat be detected in real time on embedded systems using carbody accelerations?

Yes, the experimental results in the first publication have validated that the proposed 1D LCNN architecture, i.e. LightWFNet, is able to detect wheel flats by using carbody accelerations and can be executed in real time on embedded systems.

The experiment was based on the following facts. The experiment data was obtained by creating a synthetic wheel flat with a length of 20 mm across the entire wheel surface, as shown in Figure 9 of Pub.1. This excludes the possibility that a wheel flat may not appear in the wheel-rail contact area. In other words, the wheel-flat excitation always exists. Furthermore, the measurement equipment, incl. accelerometers and data acquisition devices, used in the field test was not real condition monitoring device. The quality of the acquired data in terms of the signal-to-noise ratio is much better than that of a typical monitoring device. In addition, the employed freight wagon had a revision before the field test and thus had a good technical condition. A consequence of all the above facts is that the wheel-flat patterns do not vanish in carbody accelerations. As presented in Figure 3 of Pub.1, the periodic pattern is recognizable at the carbody level, although it is much weaker than that at the bogie and axlebox levels. This should be the prerequisite for WFD using carbody accelerations. Figure 10 compares the carbody acceleration with and without a wheel flat at the vehicle speed of 50 km/h in the time domain and the wavelet scalogram. In this speed range, the proposed LightWFNet can achieve a detection accuracy of 95.17%. The weak periodic surges can be visible in the left diagram. In the right diagram, the vibration energy in the wavelet scalogram is even higher due to the severe track irregularities. However, it does not show a periodic pattern. In comparison, detection at high speeds (over 85 km/h) is much harder. As shown in Figure 11, the carbody acceleration with and without wheel flats can hardly be distinguished. This indicates that the changes in experimental conditions may result in the vanishment of wheel flat patterns, making the detection infeasible. One limitation of our Pub.1 is that we were not able to test the proposed LightWFNet under different experimental conditions due to a lack of data. The decisive factor of detection feasibility was unclear.

A suspicion may be raised why complicated deep learning approaches have to be used when the wheel flat pattern is recognizable after signal processing. As reviewed in Section 2.2, most of the relevant studies in industry and academia for WFD concern signal processing techniques for denoising and the identification of the features. This approach works on the typical healthy and wheel-flat signals. However, unexpected interferences in practice may distort the typical signal patterns. This problem has been presented in Pub.2. In comparison, deep learning approaches are able to adaptively filter the input data and automatically learn

the features in an end-to-end manner. The features learned by a deep learning model can be more effective to distinguish the data samples of different health conditions. In Pub.1, we compared these two approaches in terms of the WFD accuracies. We used HT and CWT to process the raw carbody accelerations and extracted several statistical features such as skewness and kurtosis of the envelope spectrum and scale-averaged wavelet scalogram, respectively. Afterward, 16 features per data sample were selected based on correlation and importance analysis. A classifier was built and trained based on gradient boosting decision tree (GBDT). This classical machine learning approach has achieved the average detection accuracy of 75.36%, whereas the proposed LightWFNet achieved 93.58%. To compare the manually defined features and the ones learned by LightWFNet, the feature distribution map with and without wheel flat is illustrated by t-distributed Stochastic Neighbor Embedding (t-SNE), which converts the high-dimensional features into the 2D embeddings for visualization. Figure 12 and 13 show the distribution map produced from the manually defined features the ones learned by LightWFNet, respectively. The learned features conspicuously form two clusters, representing the vehicle conditions with and without wheel flat. Therefore, the subsequent classification layer can easily achieve a higher detection accuracy than that by GBDT.

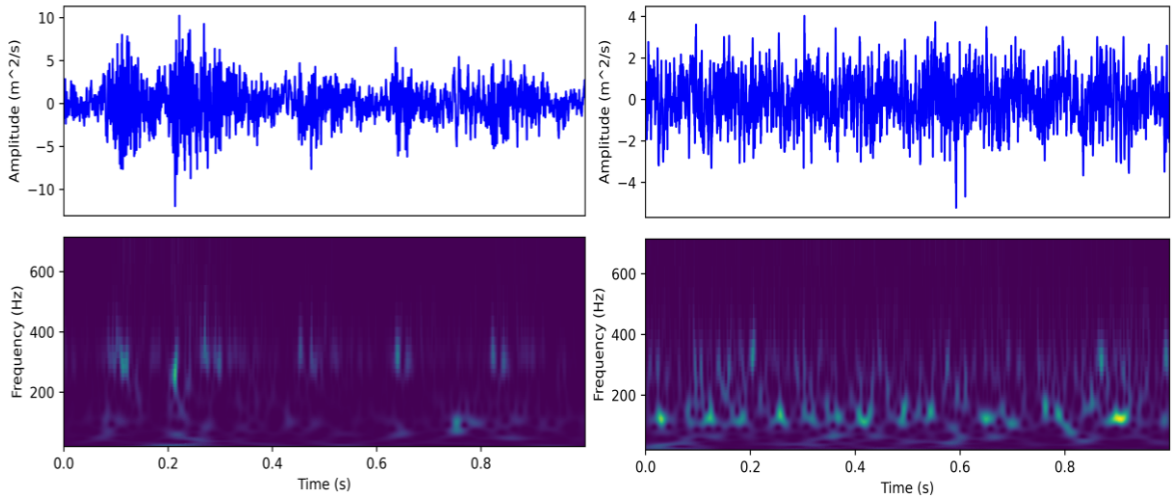


Figure 10 Left: detectable wheel flat signal, right: normal carbody accelerations in the time domain and the wavelet scalogram at the vehicle speed of 50 km/h

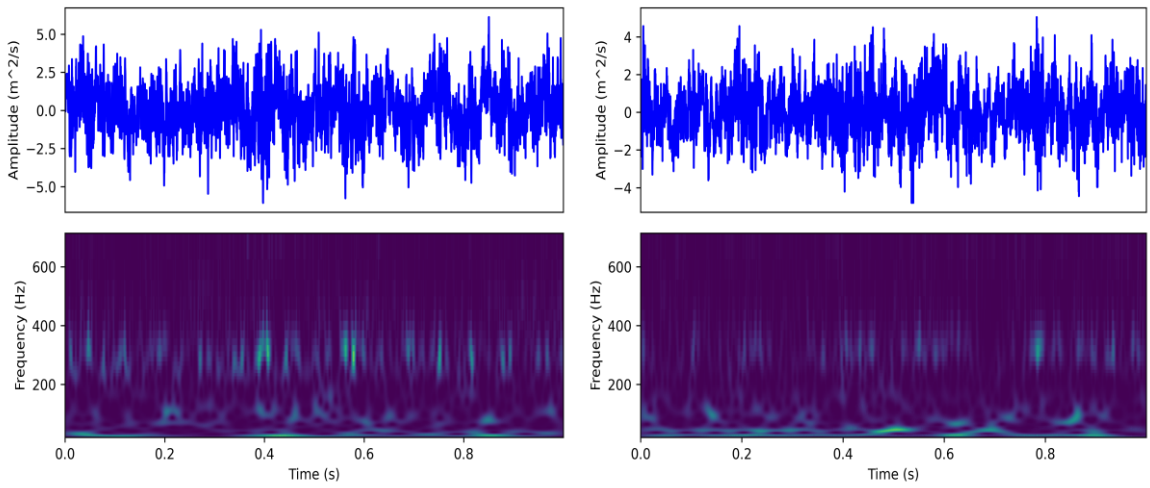


Figure 11 Left: miss detected wheel flat signal, right: normal carbody accelerations in the time domain and wavelet scalogram at the vehicle speed of 90 km/h

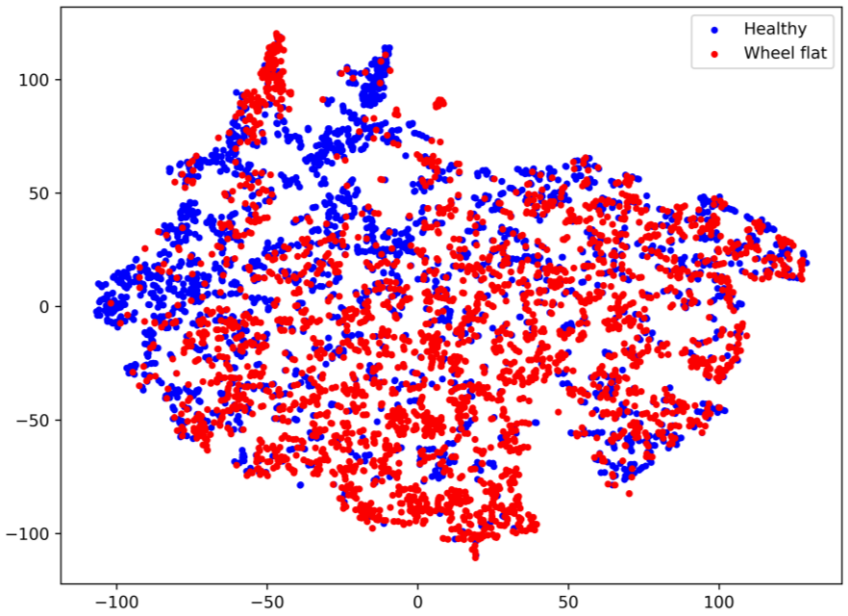


Figure 12 t-SNE feature distribution map derived from the manually defined features in the speed range 25 km/h – 45 km/h. Blue points indicate the healthy condition. Red points indicate wheel flat.

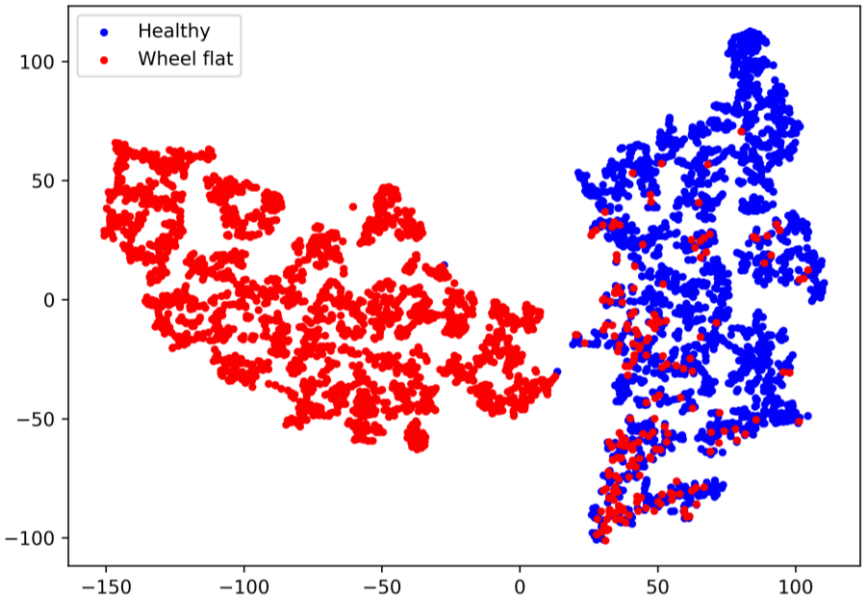


Figure 13 t-SNE feature distribution map derived from the features learned by LightWFNet in the speed range 25 km/h – 45 km/h. Blue points indicate the healthy condition. Red points indicate wheel flat.

In terms of real-time capability, FLOPs were used to roughly measure the computational complexity of a diagnosis model. A fair comparison between different deep learning models can be done by using one common FLOP counting function in the same software framework. As shown in Table 6 of Pub.1, the proposed LightWFNet costs over 10-times fewer FLOPs than other SOTA lightweight deep learning diagnosis models. The FLOPs of signaling processing methods can be theoretically estimated based on the relevant formulas in the

literature. We found that LightWFNet merely requires one-third of FFT's FLOPs, given that the length of the input data sample is 5000. This indicates that LightWFNet has higher computational efficiency than conventional diagnosis approaches that rely on signal processing and feature calculation. On the other hand, the execution time highly depends on the software implementation and the execution platform. For onboard condition monitoring, the diagnosis models were assumed to be executed on embedded systems. As a demonstration, we have successfully tested LightWFNet on Raspberry Pi 3 Model B for real-time inference, which is equipped with a four-core Cortex-A53 processor. However, the superiority of LightWFNet in computational complexity has not been revealed much. The inference latency of LightWFNet was not much smaller than another SOTA diagnosis model, although LightWFNet has 10-times fewer FLOPs. It ran slower than FFT, despite having one-third of FLOPs. FLOPs did not translate to the real latency. This was the main limitation of Pub.1. We used Bayesian optimization to search for the most accurate CNN architecture and manually conducted several lightweight measures to reduce its FLOPs and parameter numbers. The on-device latency has not been involved in the design process. In the future, the on-device latency of each CNN component should be measured prior to the network design. The network architecture search should include both diagnosis accuracy and network latency as the optimization objective. In this way, the best tradeoff between latency and performance can be achieved.

7.2. Are the algorithms for wheel flat detection robust to variable railway operating conditions?

The machine learning and deep learning based diagnostic approaches highly rely on the quality and quantity of the training data. In Pub.1, the diagnostic model of the proposed LightWFNet has merely trained on one dataset collected in a single field test. Although the trained model has achieved outstanding performance for feature extraction and WFD, it is unclear whether the model can work well on other datasets without further training. This concern stems from a basic assumption of machine learning that the training and test data belong to the same feature space and follow the same probability distribution. When the training and test data are obtained from one field test that is performed under very similar operating conditions, the assumption is fulfilled. However, the limited data from a single field test may significantly underrepresent or misrepresent the true underlying data distributions. For instance, when the test wagon is replaced. the wagon runs at different speed ranges, or the data are measured by different devices, the data distribution may be changed. This issue is well known as covariate shift [69]. Therefore, an empirical study of model robustness against condition variations commonly encountered in normal railway operations was conducted in Pub.2.

In the experiments, the axlebox accelerations measured under one condition were used for training, while those measured under a different condition were used for testing. The training dataset is termed the source domain. The test dataset is termed the target domain. The disparities of the feature distribution caused by speed variations, changing test wagons and different track irregularities were investigated by a distance measure. Also, their influences on diagnosis were tested by several WFD algorithms. We have found that the classical machine learning approach based on manually defined features was relatively robust to speed variations and changing test wagons, but vulnerable to severe track irregularities. As shown in

Figure 8 in Pub.2, severe longitudinal irregularities cause oscillatory interferences in axlebox accelerations. A discrete rail defect such as rail squats and broken rails or the turnout in a switch and crossing area could result in an impulsive interference. Several adjacent impulsive interferences form a periodic-like pattern. These interferences distort the frequency patterns of the acceleration signals and thus the extracted features. On the other hand, the deep learning approach was not much affected by severe track irregularities but was sensitive to speed. Given a fixed signal length, variations of vehicle speeds reshape the acceleration signal. The classifier considers that the high-speed signal is different from the low-speed one, which results in misclassification. The worst case is that the deep learning models are trained on a high-speed source domain and tested on a low-speed test domain. Changing test wagons refers to measuring axlebox accelerations on different wagons. As shown in Figure 7 in Pub.2, the wheel flat signals on different wagons look similar. Feeding the raw acceleration signals in the time domain directly into the deep learning model, its diagnostic performance is significantly impaired in the test domains. As shown in Table 5 in Pub.2, ResNet (which is a well-known CNN) achieved the detection accuracy of 99.01% on the source-domain dataset M1, 81.06% on the target-domain dataset M2 and 56.89% on the target-domain dataset M3. For further investigation, the feature maps learned from M2 and M3 datasets are visualized in Figure 14 and 15, noting that the target-domain datasets M2 and M3 are not involved in the training process of ResNet. In both figures, the wheel-flat signals and the healthy signals in the test domain can be well separated into two clusters. This indicates that ResNet is able to extract discriminative features from the unknown test domains. However, the cluster of M3 healthy points (shallow green points in Figure 15) is far away from the M1 healthy cluster (shallow red points). In other words, the disparities between the source and target domain learned by ResNet are larger than those between the healthy and faulty classes within the target domain. Therefore, the classifier trained on the source domain is not able to correctly separate the two classes in the target domain.

The robustness study conducted in Pub.2 had the worst-case assumption that the models were merely trained on one training domain. When the models are trained on the data with certain diversities, the robustness problem can be largely mitigated. Regarding speed variations, it is not hard in practice to collect the healthy and faulty data at different speeds. Even if data at extremely high or low speeds are not available, these speed ranges can be excluded from the diagnosis process to avoid misclassification. Regarding track irregularities, the training data certainly includes the information on diverse track and rail irregularities, as long as the training data is collected in normal rail operation. Consequently, the trained deep learning models are not much affected by track irregularities.

The most critical robustness problem corresponds to the deployment of the trained diagnosis model on different vehicles that may have different dynamic behaviors. As training data on the target vehicle is mostly unavailable, the diagnosis model has to face the unseen target domain. When the target domain is very different from the existing source domains collected on other vehicles, the pre-trained diagnosis model would probably fail. The existing commercial condition monitoring systems avoid this issue by handing over the diagnosis decision to the users. Their systems typically derive a healthy indicator from the defined features and visualize the healthy indicator over time for the individual assets. The users have to define a diagnosis threshold based on the experience and historical data for the individual assets. This approach applies to scenarios where real-time diagnosis is not mandatory. As

investigated above, deep learning models can extract the discriminative features from the unseen target domains. These features can replace statistical features, since they are more robust to signal interferences caused by track irregularities. In the case of real-time fault diagnosis, the pre-trained diagnosis model is expected to make a correct classification for unknown target domains. This brings the next research question, how the robustness of the diagnosis algorithms can be improved.

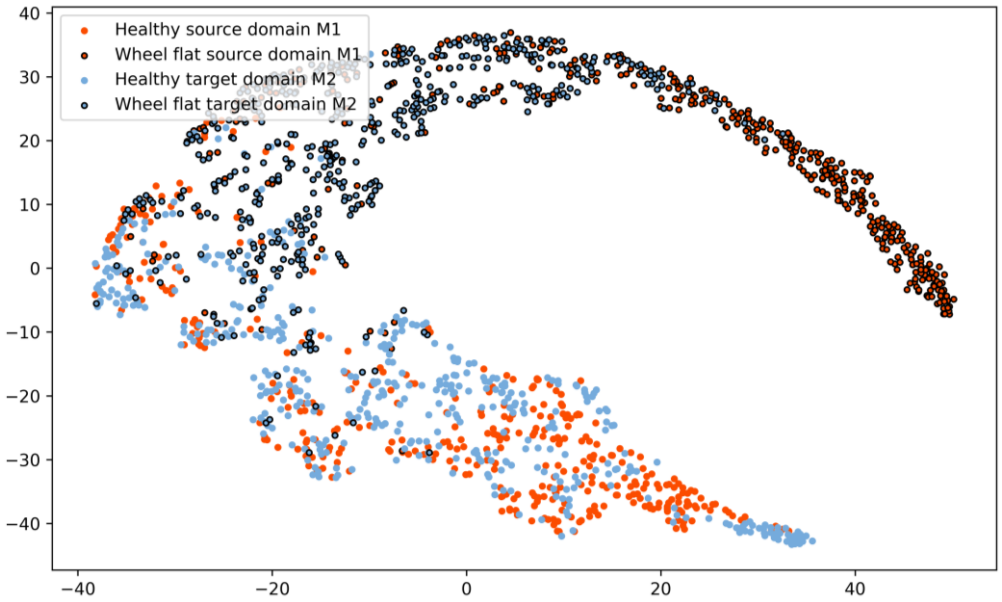


Figure 14 Features learned by ResNet on the source-domain dataset M1 and the target-domain dataset M2

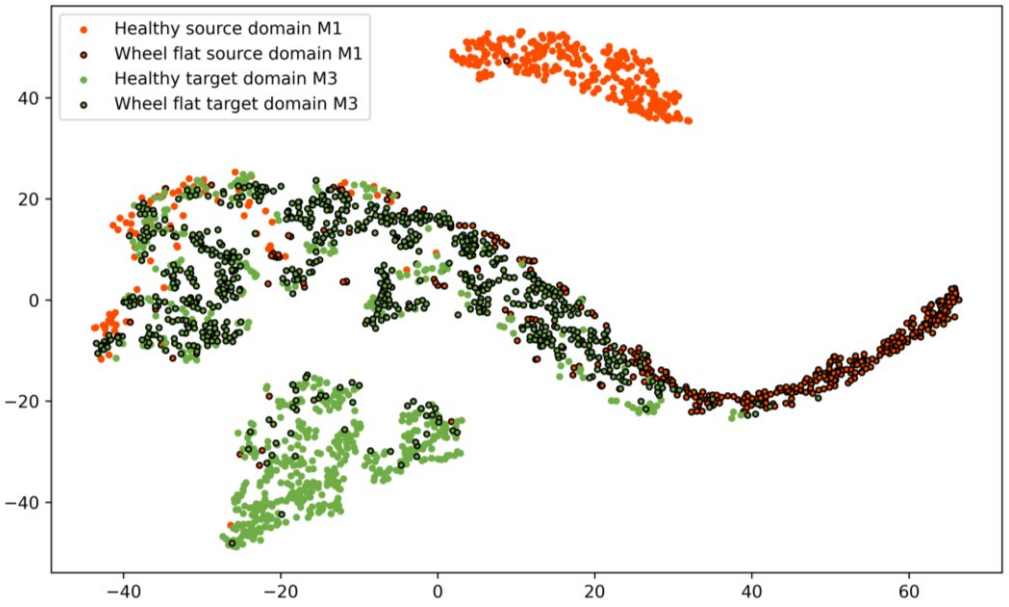


Figure 15 Features learned by ResNet on the source-domain dataset M1 and the target-domain dataset M3

7.3. How can the robustness of the diagnosis algorithms be improved?

In Pub.2, a data augmentation framework was proposed to generate synthetic data to extend the diversity of the source domain for model training and thus enhance the robustness of

diagnostic models. This data augmentation framework incorporates MBS to simulate vehicle failure behavior under arbitrary operating conditions and FWFSA as a new data augmentation technique to augment the simulated faulty data. The proposed MBS-FWFSA produces infinitive reality-augmented simulation data as the additional synthetic source domains. Its effectiveness in robustness enhancement has been experimentally validated in the cases of speed variations, changing test wagons and different track irregularities. Figure 16 and 17 show the feature maps after data augmentation. ResNet was trained on the hybrid source domain, consisting of M1 and the synthetic data samples generated by the MBS-FWFSA framework. The added synthetic training data reshape the feature map learned by ResNet. Comparing Figure 17 with Figure 15, the green points get closer to the red points. Accordingly, the intra-class distance (i.e. the distance between the same-class points) is less than the distance between the source and target domain. This results in a significant improvement of detection accuracy on the target domain M3 (from 56.89% to 79.6%).

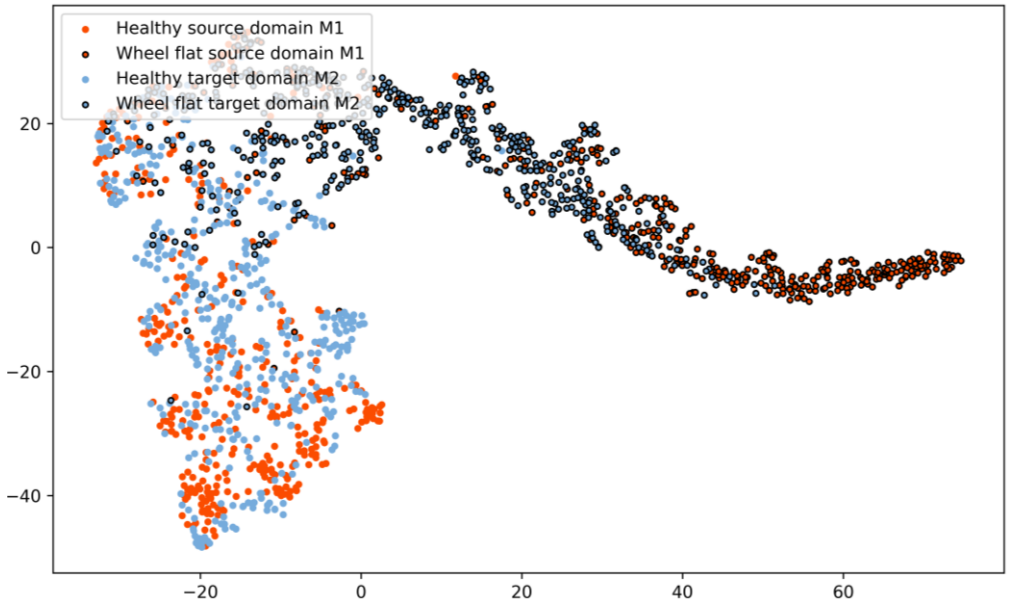


Figure 16 Features learned by ResNet after data augmentation on the source-domain dataset M1 and the target-domain dataset M2

Nevertheless, the detection performance on the unseen target domain is still much worse than that on the source domain. In the previous experiments, the hybrid source domain merely includes M1 and the synthetic dataset. We hypothesize that more data with greater diversity can further improve model robustness. For instance, we include the M1, M2 and synthetic datasets in the source domain for training. The trained model is tested on the M3 dataset. The test accuracy is expected to be better than the previously achieved 79.6%. Table 6 presents the test results of ResNet trained on different source domains and tested on different target domains. The WFD accuracy on M3/M2 has not been largely increased by adding M2/M3 into the source domain. Despite more training data, the batches of data samples were randomly picked and fed into the neural network without specific strategies during the training progress. In our case, one batch contains 32 data samples. Source domains' proportion within these 32 samples are randomly determined which may cause the unbalanced learning progress and ultimately result in a biased diagnostic model. A classical algorithm that addresses this issue is known as empirical risk minimization (ERM) [70]. It averages the

training loss for each source domain at each training step. In other words, the proportion of the source domains within a single training batch should be equal. This may force ResNet to learn more general features of wheel flat, rather than specific domain features.

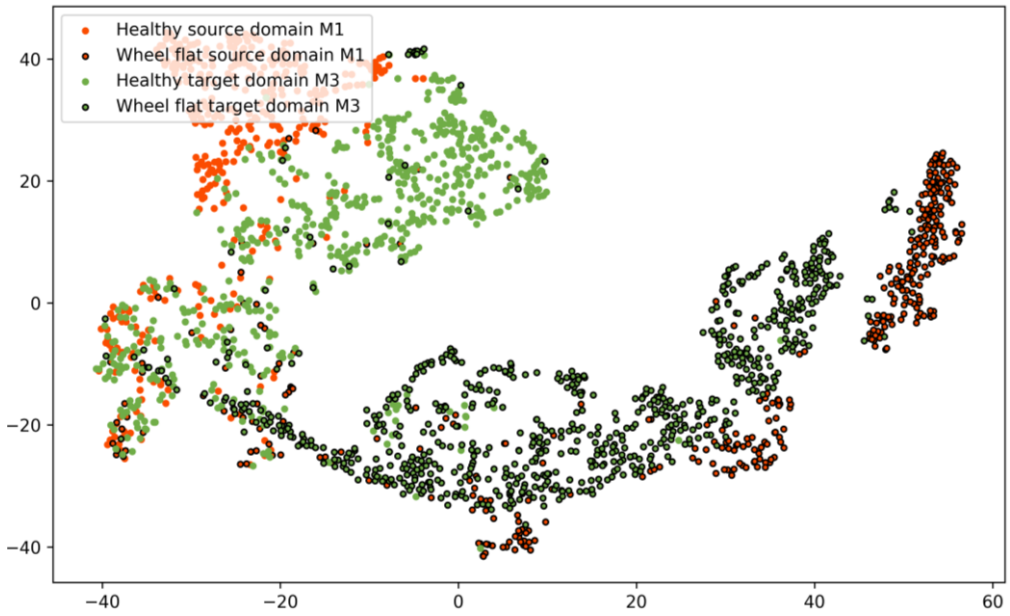


Figure 17 Features learned by ResNet after data augmentation on the source-domain dataset M1 and the target-domain dataset M3

Table 6 Wheel flat detection accuracies of ResNet trained on different source domains and tested on different target domains

Source Domain	M1	M2	M3
M1	99.01 ± 0.85%	81.06 ± 2.95%	56.89 ± 5.32%
M1 + Syn.	97.72 ± 1.93%	86.33 ± 1.99%	79.62 ± 4.34%
M1 + Syn. + M2	98.11 ± 1.11%	-	80.67 ± 3.46%
M1 + Syn. + M3	98.16 ± 1.44%	84.52 ± 1.74%	-

The robustness problem has been intensively studied as domain generalization during the last decade in the machine learning community. Apart from ERM, different algorithms have been proposed to incorporate the invariances across the source domains into a classifier, in hopes that such invariances can hold in unseen target domains for robust classification [71]. Several domain generalization solutions have been adopted for machine fault diagnosis [72-77]. They aimed at real-time and robust fault diagnosis in unseen target domains, being consistent with our goal for wheel flat detection. In the future, we will incorporate domain generalization methods into the model training procedure along with data augmentation towards robust fault diagnosis.

7.4. How can track alignment be monitored in real time by using inexpensive sensors?

The third publication stemmed from a project task, aiming to monitor the relative lateral displacement of the wheel on the rail (LDWR). The motivation was based on the hypothesis that track alignment was correlated to lateral acceleration and wheel displacement. To measure LDWR in real time, we have developed and validated the proposed computer vision solution based on the affordable off-the-shelf cameras in the third publication. However, the hypothesis cannot be verified due to the lack of acceleration measurement. The final validation test was performed under the supervision of the Italian railway infrastructure manager (i.e. Rete Ferroviaria Italiana). During the test preparation, the request for the accelerometer installation on the axlebox and bogie frame was rejected. Therefore, we cannot answer this research question based on the acquired data. It requires sensor fusion of wheel acceleration and the calculated wheel displacement. In this discussion, we mainly investigate the correlation between the wheel lateral displacement and track alignment.

The test vehicle was equipped with a commercial track geometry inspection system provided by Mermec, as shown in Figure 18. It is based on the optical-inertial measurement principle. Two optical boxes (containing a laser and a high-speed camera) are used for the measurements of the rail profile and the rail location, while the inertial unit (incorporating high accuracy accelerometers and solid-state rate gyroscopes) makes available the linear and angular accelerations. The combination of optical and inertial data allows the determination of the track geometrical and rail profile parameters. The track geometry parameters measured in the track section between Santa Severa and Civitavecchia were provided by Mermec for comparison with our measurement. The Mermec system merely outputs the final processed parameters, which do not contain the information of wheel displacement on the rail. A direct comparison for the validation purpose is impossible. Instead, the correlation between the synchronized track alignment and wheel displacement is investigated.

Figure 18 Mermec track geometry inspection system

The synchronization between two measurements relies on timestamps and the known locations of the fixed points such as turnouts and level crossings. The alignments in different wavelength ranges and track gauge deviations measured in the synchronized track section are

presented in Figure 19. There are severe irregularities at around 290s and 900s, especially in large wavelength ranges. Comparing the calculated wheel displacement with the measured track alignment in the wavelength range D3, we observe that the peaks of the red curve are always accompanied by the blue peaks in Figure 20. This means, severe alignments consistently result in large lateral displacements of the wheel. However, the wheel displacement can occur without a large alignment. The video recordings with the calculated wheel displacement for this track section can be accessed at Youtube. In the video, the red point on the wheel refers to the wheel reference point detected by the proposed algorithm, while the blue point on the rail refers to the rail reference point. The lateral wheel displacement is the horizontal distance between these two points in the image plane. Observing the video, the small variation of the calculated wheel displacement could be the detection errors or uncertainties. This has been discussed in Section 4.5 of Pub.3. In some cases, (for instance between 5s-12s in the video,) the wheel movement can be clearly observed and has been detected by the algorithm, despite small track alignments. On the other hand, the wheel movement between 275s (4'35) and 305s (5'05) in the video is caused by large track alignments. To be more precise, we calculate the rolling windowed Pearson correlation coefficients, which represent the dependence of the track alignment as the excitation and the LDWR as the vehicle response. Figure 21 presents the color map of the correlation coefficients, obtained from the normalized LDWR and the normalized track alignment. The dark red indicates a high linear correlation, which often appears at the moments of large track alignments. This is consistent with our observation in Figure 20.

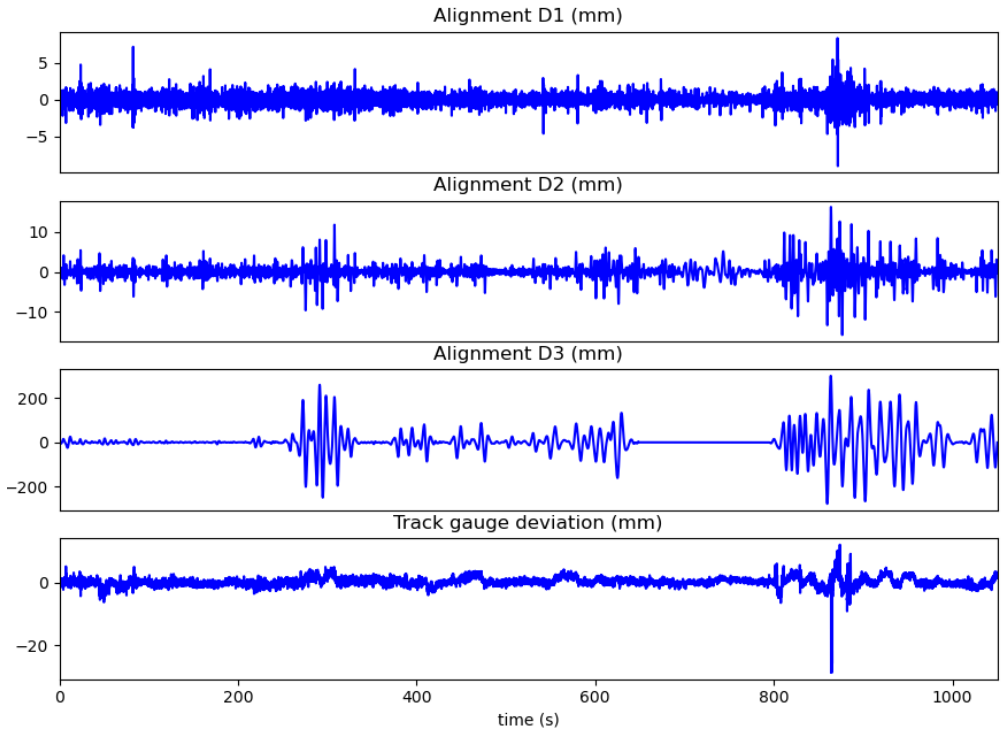


Figure 19 Track alignment and track gauge deviation outputed by the Mermec system in the synchronized track section, where $D1: 3m < \lambda \leq 25m$, $D2: 25m < \lambda \leq 70m$, $D3: 70m < \lambda \leq 150m$

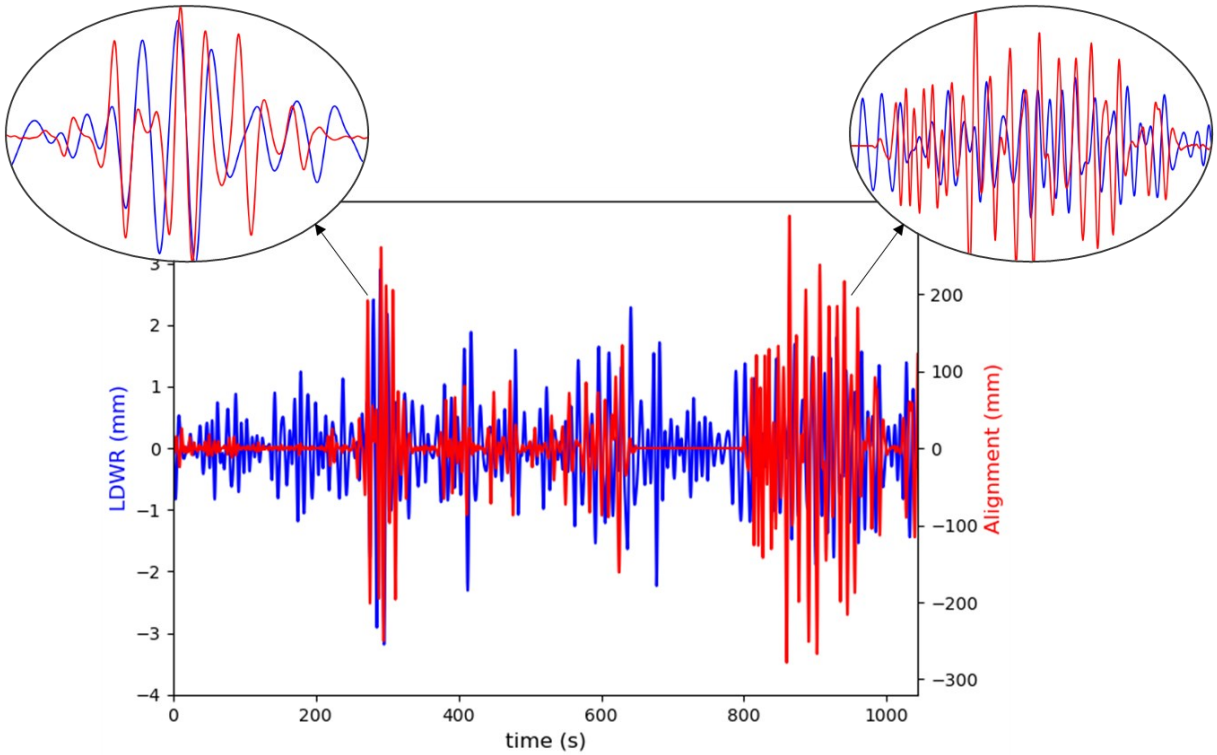


Figure 20 Comparison between the measured track alignment and the calculated wheel displacement in the wavelength range D3

To derive track alignment, lateral accelerations must be measured. The envisaged hypothesis can be proved, as long as lateral accelerations can compensate for wheel displacements not caused by track alignments. This will be our future work.

7.5. Is the proposed method based on optical sensing robust to variable railway operating conditions?

The robustness of the proposed virtual point tracking algorithm mainly concerns two aspects. First, the generalization ability of the trained deep learning model should allow easy adaptation when the computer vision system is installed on different in-service vehicles. To examine this, we have conducted three field tests for data collection and system validation. As described in Section 4.1 of Pub.3, Test A was conducted within a workshop plant in Italy at low speeds (up to ca. 10km/h). Test B was performed on a different vehicle in Lithuania at speeds of around 20 km/h. Test C was carried out on the same vehicle as in Test A, however, in regular operating conditions on the main lines at speeds of up to 100 km/h. In each test, several hours of video sequences were recorded at 30 fps. Hundreds of images were randomly selected and annotated with the true coordinates of the defined reference points for the numerical experiments. When training and test data stem from the same test(s), the proposed LightPointNet can achieve the optimal detection accuracies, as reported in Section 4.3 of the third publication. However, its performance significantly degrades, when the trained LightPointNet model is directly applied to a different test dataset. For instance, the model trained on the source-domain datasets from Test A and B cannot work well on the target-domain dataset from Test C. This generalization problem is consistent with that for WFD, as discussed in Section 7.3. To overcome this problem, we fine-tuned the model on a small part

(0.1%) of the target-domain dataset, which can be regarded as the “calibration” process. It requires manual annotation of the new training data. A promising way to improve the generalization ability and avoid the calibration process is to migrate the domain generalization measures into the model training process [71]. This requires the involvement of several source domains. In our case, only two source domains are available, which are not sufficient to force deep learning models to learn domain-agnostic features. More data with certain diversity should be collected by installing the computer vision system on different vehicles, changing the installation position of the cameras or using the cameras with different fields of view and resolutions.

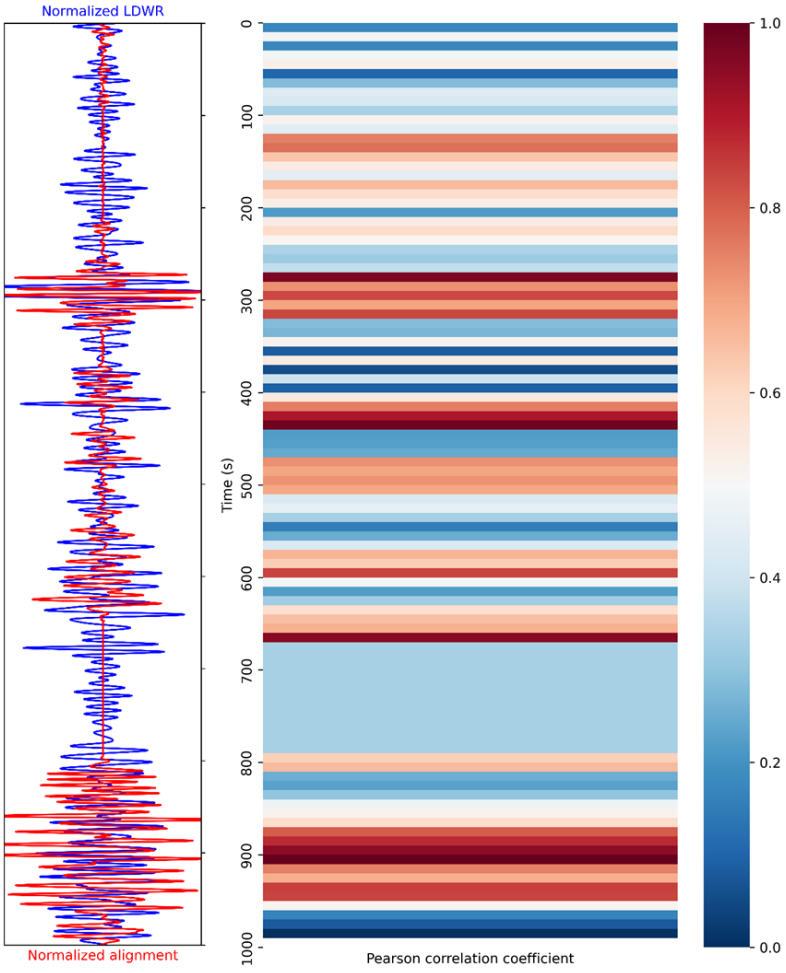


Figure 21 Rolling windowed Pearson correlation coefficients between wheel displacement and track alignment. The time window is set as 30 s.

Second, the model is desired to work in different lighting and weather conditions, although optical systems are inherently sensitive to the variations of these conditions. The conventional track inspection systems which involve any optical sensors cannot work properly in bad weather such as rain, snow, sleet, etc. Apart from this, the system requires frequent cleaning, since the optical sensors are also vulnerable to contaminants. From the software perspective, this sensitivity can trace back to the fact that the image processing techniques employed in conventional track inspection systems rely on manually designed denoising filters, Canny edge detection and Hough line transform [50,51], which do not have the adaptive ability to

the new scenes and thus are vulnerable to condition variations. We have implemented a baseline method to detect and track wheel displacements based on these techniques. As shown in Figure 22, the pipeline mainly involves template matching for wheel detection and line detection for rail detection. The wheel flange's template points and two points on the left railhead edge are manually selected in the calibration process. The two points on the rail are used to calculate the rail edge slope, while the points on the wheel are used to generate a wheel template. In pre-processing, the median filter is applied for noise reduction, and histogram equalization is applied for contrast enhancement. Canny edge detector is used with a Gaussian blur filter to extract the edges. Afterward, a template matching algorithm using correlation coefficients is used to detect the position of the wheel flange automatically. For line detection, several filters are stacked to emphasize the rail lines. Afterward, the probabilistic Hough transform is applied for line detections. A small range of the slopes is defined according to the pre-calculated rail line slope, allowing for selecting the desired rail line from the detected line candidates. Finally, the extended lines of the selected line sections are created, allowing to calculate the horizontal distance from the wheel reference point to the rail line. This baseline method works well on the video sequences from Test A and C but fails on the video sequences from Test B due to weak illumination.

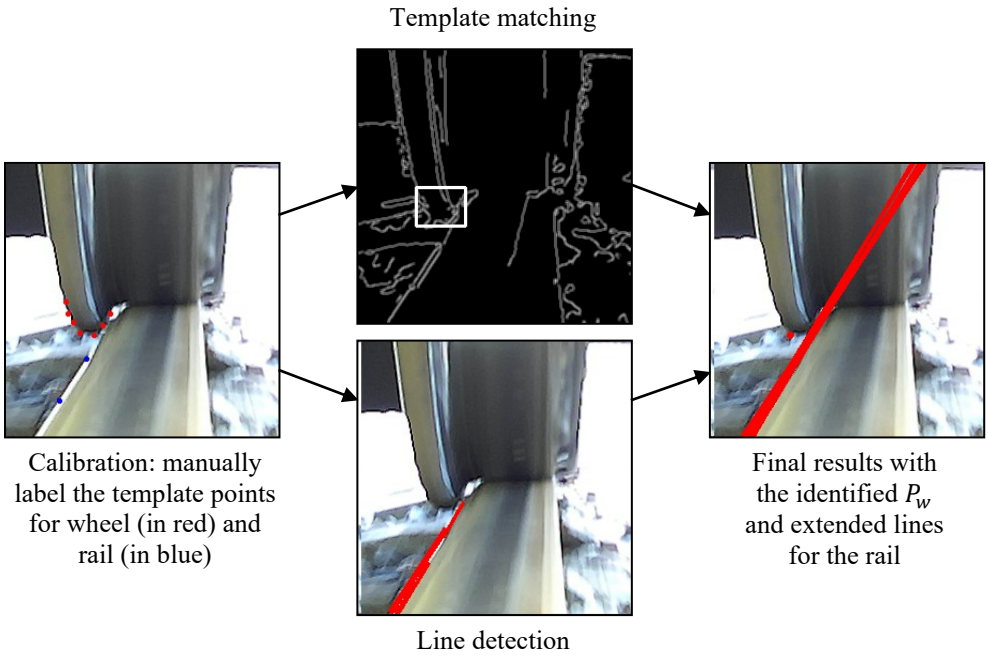


Figure 22 Pipeline of the baseline method based on conventional image processing techniques

To further test the robustness of our LightPointNet and the baseline method against contaminants and different weather conditions, we generate a synthetically corrupted dataset by manipulating the original images. The representatives of the synthetic images can be seen in Figure 6 of Pub.3. Within the total 767 corrupted test images, LightPointNet achieves a detection accuracy of 77.9% with data augmentation and 63.4% without data augmentation, where the errors of the detected reference points are below 5 pixels. In contrast, the baseline method cannot deliver any meaningful results. Despite significant performance impairment, LightPointNet as a deep learning based algorithm shows certain robustness against

contaminants and bad weather. It has the potential to further increase its robustness by domain generalization. In contrast, the conventional image processing techniques highly rely on parameter tuning for the used filters and algorithms such as Gaussian filter, Canny edge detector and Hough line transform. The relevant parameters have to be calibrated manually for the target scenes. This calibration process requires high expertise and cannot be avoided.

7.6. Other potential uses

The 1D CNN proposed in Pub.1 and 2 was demonstrated for WFD. A CNN consists of an encoder for feature extraction and a classifier for classification. As discussed in Section 7.1-7.3, the trained encoder can be used independently for robust feature extraction, instead of manually defined statistic features. By replacing the classifier with a decoder, it can be used for signal compression and reconstruction. This concept is well known as autoencoder [78].

Real-time condition monitoring, especially vibration monitoring with a high sampling frequency, produces a large amount of data. In practice, only the extracted features and the metadata are transmitted wirelessly to the cloud server. The raw data is discarded due to limited network bandwidth or economic aspects, although it is crucial for continuous improvement of the data processing algorithms. General data compression techniques struggle to achieve high compression rates while recovering the data without losing information that reveals the condition of the machine. Autoencoder has been widely used for signal compression and reconstruction of images [79], seismic vibration data [80], inertial measurement data [81], and electroencephalography signals [82]. For machine fault diagnosis, Russel et al. [83] proposed a deep convolutional autoencoder (CAE) with local structure and physics-informed loss terms to compress acceleration signals. These works show the feasibility of CAE for task-aware data compression and construction.

In the applications of onboard vibration monitoring, the encoder and the classifier can be integrated into the onboard embedded monitoring device for real-time fault diagnosis. The outputs of the encoder as the learned features and the diagnostic results can be transmitted to the cloud. The decoder is deployed on the cloud to reconstruct the raw acceleration signals from the received features. In this configuration, the monitoring device is able to conduct fault diagnosis and inform the relevant entities of the detected failures in real time. The learned features are used to derive a healthy indicator, revealing condition degradation over time. The reconstructed raw signals can be used to continuously train and update the diagnostic model, incl. the encoder, the decoder and the classifier.

The virtual point tracking algorithm proposed in Pub.3 served as a part of track alignment monitoring. The calculated LDWR can be used for vehicle hunting detection.

Hunting motion refers to the lateral swing of the railway vehicle (wheelset/bogie). The violent lateral swing above the vehicle critical speed damages tracks and wheels and may result in derailment. Bogie lateral accelerations are commonly adopted for hunting detection on in-service vehicles. The standard detection algorithm relies on two manually defined features, namely the filtered RMS values and the counts exceeding an amplitude threshold [84]. This algorithm can detect the large amplitude hunting, but not the small-amplitude hunting [84]. Different signal processing methods and specific features have been investigated [84-87]. However, vibration monitoring is an indirect measurement method for hunting detection. It

can be interfered with by severe track irregularities and different vehicle conditions, which is similar to the robustness problem of WFD. In addition, it is extremely hard to validate the detection algorithm in practice. Even in a controlled experimental environment, a vehicle is hardly forced into hunting motions for measurement.

Video recording of wheel movement is a direct measurement method. The proposed virtual point tracking algorithm is able to track the lateral movement of the wheel on the rail. The large periodic wheel displacement may indicate hunting instability. Once detected, the raw video data at that moment can be directly used as validation evidence.

8. Conclusions

More and more condition monitoring systems are involved in the maintenance process for railway assets, accelerating the paradigm shift from traditional corrective and preventive maintenance towards efficient condition based maintenance and predictive maintenance. Condition monitoring aims to determine the up-to-date health conditions of the monitored assets for fault diagnosis and degradation prognosis. This requires intelligent algorithms for data processing. Especially in practical railway operation, real-time capability and robustness against variation of operating conditions are decisive for reliable monitoring results. Unreliable results such as false alarms and wear predictions can only disrupt normal operations. However, existing research in railway academia has not paid much attention to this.

In the present cumulative dissertation, we conduct extensive research towards robust real-time condition monitoring and fault diagnosis for railway assets. Lightweight convolutional neural network (LCNN) is introduced for real-time fault diagnosis. Data augmentation is applied to generate synthetic data in arbitrary operating conditions, which may be difficult to obtain in practice. Adding the synthetic data in the training dataset can enhance the robustness of the trained LCNN (or other deep learning based) diagnostic models. The proposed methods are demonstrated in the applications of vibration monitoring for wheel flat detection (WFD) and video monitoring to support track geometry monitoring.

The first accomplishment of our research is the proposed design process of LCNN for the given monitoring data. In different monitoring tasks, the dimension of the monitoring data as the input for the diagnostic model can be different. There exist no universal model applicable for arbitrary monitoring tasks. The proposed design process aims to automatically search for a suitable architecture of LCNN by Bayesian Optimization so that the identified LCNN can achieve an optimal diagnostic result for the given input data. Furthermore, we demonstrate the designed LCNN for WFD overperforms the classical diagnostic approach, which relies on the manually defined features, and other state-of-the-art lightweight deep learning based diagnostic models in terms of detection accuracy and computational complexity. As diagnostic algorithms for condition monitoring are commonly executed on embedded systems with limited computation power, the complexity of the algorithm is crucial in practical use. One interesting finding is that the complexity of the designed LCNN is even less than that of fast Fourier transform (FFT) in terms of floating point operations (FLOPs), given that the length of the input signal is 5000. Also, the designed LCNN can directly take the raw vibration signal as the input for WFD and avoid using signal processing techniques such as FFT. Therefore, the overall computational cost of LCNN is minimal. However, we experimentally found that executing an LCNN model on an embedded system takes no less time than executing FFT. In other words, the execution latency of a diagnostic model is also affected by other factors besides FLOPs. In future work, the platform-aware latency should be included together with the diagnostic accuracy as the optimization objective for the neural network search.

Secondly, we empirically investigate the robustness of common diagnostic methods against the variation of vehicle speeds, monitored wagons and track conditions. The variation of operating conditions may cause the shift of data distribution (termed domain), which is not

“seen” by the diagnostic model during the training process. A fact is that the machine learning based models are vulnerable to the unseen domains. This has been proved in the conducted experiments. Adding a part of data samples from an unseen domain to the training dataset, the diagnostic performance for this domain will be largely increased. Therefore, the variation of vehicle speeds and track conditions is not a real problem in practice, as long as the training data can be collected at different running speeds and on different tracks. However, the variations of vehicle conditions may be infinite so we are not able to include all the possible domains in the training dataset. In practice, the diagnostic models are normally trained on the limited data collected in one or several field tests. Based on this fact, we propose to involve the synthetic data generated by the proposed data augmentation framework in the training dataset to enhance the robustness of the trained models. The data augmentation framework can take advantage of multibody dynamic simulation to mimic different vehicle dynamic behaviors under various operating conditions. It increases the diversity of the training dataset. This strategy of robustness enhancement has been validated in the experiments. However, the improved detection accuracy is not high as expected. Adding more data cannot result in further improvement, since the model may be saturated but still cannot be generalized to the unseen domain. By visualizing the distribution of the features learned by the trained diagnostic model, we find that the robustness problem mainly lies in the classifier rather than the featurizer. Although the model cannot achieve an optimal detection accuracy in the unseen domain, the featurizer within the model can still extract the discriminative features. In other words, the features extracted from the wheel-flat signals and the healthy signals in the unseen domains still form two clusters that can be easily separated. However, the distance between these two clusters is smaller than that between the source(/training) domain and the target(/test) domain. This is the reason why the classifier trained on the source domain fails to make a correct classification in the target domain. In future work, specific measures should be taken to align the domain disparities, which are studied in the research area of domain generalization(/adaptation).

Thirdly, we demonstrate that a similar diagnostic approach based on LCNN and data augmentation can be adopted to process the image data for track geometry monitoring. The designed 2D LCNN can detect the virtual points defined on the wheel flange and the rail edge in the video. The horizontal distance between the wheel reference point and the rail reference point indicates the relative lateral displacement of the wheel on the rail (LDWR). We hypothesize that LDWR may support the reconstruction of track alignment by fusing with the wheel acceleration. This approach is the first attempt for track alignment estimation using a cheap camera and an accelerometer. The calculated LDWR can also be used as an intuitive indicator for hunting detection. Due to the lack of acceleration data, we cannot prove the proposed hypothesis for track alignment. Nevertheless, we analyze the correlation between LDWR and track alignment and conclude that severe alignments consistently result in large lateral displacements of the wheel. However, the wheel displacement can occur without a large alignment. To investigate the model robustness against bad weather and contamination, image corruption techniques are employed to generate synthetic data in the corresponding conditions. We compare our model to a conventional image processing approach on the raw images and the corrupted ones. The comparison reveals that the proposed deep learning based model has certain robustness against contaminants and bad weather, whereas the conventional image processing approach does not. However, the deep learning model still suffers from the generalization problem, despite data augmentation. This finding is consistent with that in the

application of vibration monitoring for WFD. In future work, domain generalization measures should be applied for robustness enhancement. Field tests in typical severe weather should be performed. Vibration monitoring should be applied to measure the wheel acceleration and synchronized with the developed computer vision system. The algorithm for data fusion should be developed and tested to verify the proposed hypothesis for track alignment estimation.

Our research has been published in three papers. The codes of the developed methods have been published and accessible at GitHub. This will help other researchers who are interested in our work. The proposed methods for real-time data processing and robustness enhancement are not confined to the two exemplary applications. They can be adapted into similar scenarios. For instance, the algorithms for WFD can be directly applied for fault diagnosis of rotating machines. The algorithms for virtual point detection can be used in civil engineering, where the displacement of large infrastructure is measured by computer vision approaches.

References

(excluding bibliography of Pub.1-3)

[1] EN 13306 (2017), "Maintenance –Maintenance terminology; Trilingual version EN 13306:2017", European Standard

[2] GCU (2021), GENERAL CONTRACT OF USE FOR WAGONS, Edition dated. 1 January 2021

[3] Leiste, M. (2018), "Roadmap zur Digitalisierung der Wagentechnischen Untersuchung," project report Nr. 16/2018, Fachgebiet Schienenfahrzeuge, Technische Universität Berlin

[4] S. Strano and M. Terzo, "Review on model-based methods for on-board condition monitoring in railway vehicle dynamics," *Advances in Mechanical Engineering*, vol. 11, no. 2, 168781401982679, 2019, doi: 10.1177/1687814019826795.

[5] R. W. Ngigi, C. Pislaru, A. Ball, and F. Gu, "Modern techniques for condition monitoring of railway vehicle dynamics," J. Phys.: Conf. Ser., vol. 364, p. 12016, 2012, doi: 10.1088/1742-6596/364/1/012016.

[6] A. Khan, A. Sohail, U. Zohoora and A.S. Qureshi (2020), "A Survey of the Recent Architectures of Deep Convolutional Neural Networks", https://arxiv.org/abs/1901.06032

[7] Y. L. Cun et al. (1990), ''Handwritten digit recognition with a back-propagation network,'' in Proc. Adv. Neural Inf. Process. Syst., pp. 396–404.

[8] S. Ioffe and C. Szegedy, 2015, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", arXiv:1502.03167

[9] R. Lengu (2018), "Deliverable D 3.6 Wayside Condition Monitoring Impact Analysis", Technical report, [online], [accessed on 13.01.2022], https://projects.shift2rail.org/download.aspx?id=b24589df-9ad6-44fe-8658-d9b88fc86d2d

[10] DTEC (2021), „Produktübersicht", [online], [accessed on 13.01.2022], https://dtec-gruppe.com/produkte/?lang=de#eagle

[11] DB Cargo (2021): „Wagon Intelligence – mehr Transparenz im Schienengüterverkehr", [online], [accessed on 13.01.2022], https://www.dbcargo.com/rail-de-de/logistik-news/transparenz-im-schienengueterverkehr-mit-wagon-intelligence-6160622

[12] EKE (2022), „Train Control and Management System", [online], [accessed on 13.01.2022], https://www.eke-electronics.com/train-control-and-management-system-tcms

[13] TIS (2019), "White Paper – The Intelligent Freight Train", report from Technical Innovation Circle for Rail Freight Transport, [online], [accessed on 13.01.2022], https://tis.ag/en/download/tis-white-paper-intelligent-freight-train-2019-06/?wpdmdl=652&refresh=61e45d17e9d641642355991

[14] Knorr (2022), „Modular Brake Control System", Knorr-Bremse product manuel, [online], [accessed on 13.01.2022], https://www.knorr-bremse.com/remote/media/documents/railvehicles/product_broschures/brake_systems/Modular_Brake_System_P_1245_EN.pdf

[15] UIC (2017), "UIC Digital Awards - Traxens", UIC report, [online], [accessed on 13.01.2022], https://uic.org/events/IMG/pdf/traxens.pdf

[16] EN 15437-2 (2012), "Railway applications – Axlebox condition monitoring - Interface and design requirements – Part 2: Performance and design requirements of on-board systems for temperature monitoring," EU technical standard

[17] SKF (2022), "Railway technical handbook -- Volume 1", [online], [accessed on 13.01.2022], https://www.messestand-online.de/innotrans18/skf/pdfs/10987_2%20EN_PDF-file,_low_resolution,_72_DPI_locked_for_editing.pdf

[18] Corni, I., Symonds, N., Wood, RJK, Wasenczuk, A., Vincent, D. (2015), "Real-time on-board condition monitoring of train axle bearings", The Stephenson Conference - Research for Railways

[19] C. Becker (2018), "Flachstellenerkennung mit Telematik", Project report, [online], [accessed on 13.01.2022], https://www.aramis.admin.ch/?DocumentID=46109

[20] Z. Ren (2018), "An investigation on wheel/rail impact dynamics with a three-dimensional flat model," Vehicle System Dynamics, vol. 57, no. 3, pp. 369–388, 2019, doi: 10.1080/00423114.2018.1469774.

[21] E. Bernal, M. Spiryagin, and C. Cole (2019), "Wheel flat detectability for Y25 railway freight wagon using vehicle component acceleration signals," Vehicle System Dynamics, vol. 232, no. 4, pp. 1–21, doi: 10.1080/00423114.2019.1657155.

[22] N. Bosso, A. Gugliotta, and N. Zampieri (2018), "Wheel flat detection algorithm for onboard diagnostic," Measurement, vol. 123, pp. 193–202, 2018, doi: 10.1016/j.measurement.2018.03.072.

[23] Y. Ye, D. Shi, P. Krause, and M. Hecht (2019), "A data-driven method for estimating wheel flat length," Vehicle System Dynamics, vol. 213, no. 214, pp. 1–19, doi: 10.1080/00423114.2019.1620956.

[24] B. Baasch, J. Heusel, M. Roth, and T. Neumann (2021), "Train Wheel Condition Monitoring via Cepstral Analysis of Axle Box Accelerations," Applied Sciences, vol. 11, no. 4, p. 1432, 2021, doi: 10.3390/app11041432.

[25] S. Chen, K. Wang, C. Chang, B. Xie, and W. Zhai (2021), "A two-level adaptive chirp mode decomposition method for the railway wheel flat detection under variable-speed conditions," Journal of Sound and Vibration, vol. 498, p. 115963, 2021, doi: 10.1016/j.jsv.2021.115963.

[26] J. Shim, G. Kim, B. Cho, and J. Koo (2021), "Application of Vibration Signal Processing Methods to Detect and Diagnose Wheel Flats in Railway Vehicles," *Applied Sciences*, vol. 11, no. 5, p. 2151, doi: 10.3390/app11052151.

[27] H. Jiang, J. Lin (2018). Fault diagnosis of wheel flat using empirical mode decomposition-Hilbert envelope spectrum. Math Prob Eng; 2018:1 – 16.

[28] R. Zhao, H. Shi (2017), "Research on wheel-flat recognition algorithm for high-speed train based on high-order spectrum feature extraction", J Mech Eng. 2017;53:102.

[29] Y. Li, MJ. Zuo, J. Lin (2017), "Fault detection method for railway wheel flat using an adaptive multiscale morphological filter". Mech Syst Signal Process; 84:642–658.

[30] C. Gericke (2013), „Methoden zur on-board-Diagnose von Radlaufflächenschäden: ein Beitrag zur zustandsorientierten Instandhaltung von Schienenfahrzeugen. Dissertation", Dissertation, Technische Universität Berlin

[31] E. Kim, N. Jayaprakasam, Y. Cui, U. Martin (2020), "Defect Prediction of Railway Wheel Flats based on Hilbert Transform and Wavelet Packet Decomposition", preprint, arXiv.org, doi: arXiv:2008.12111.

[32] M. Entezami, C. Roberts, P. Weston, E. Stewart, A. Amini, and M. Papaelias (2020), “Perspectives on railway axle bearing condition monitoring,” Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, vol. 234, no. 1, pp. 17–31, 2020, doi: 10.1177/0954409719831822.

[33] G. Xu, D. Hou, H. Qi, and L. Bo (2021), “High-speed train wheel set bearing fault diagnosis and prognostics: A new prognostic model based on extendable useful life,” Mechanical Systems and Signal Processing, vol. 146, p. 107050, 2021, doi: 10.1016/j.ymssp.2020.107050.

[34] Z. Liu and L. Zhang (2020), “A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings,” Measurement, vol. 149, p. 107002, 2020, doi: 10.1016/j.measurement.2019.107002.

[35] T. Wang, Q. Han, F. Chu, and Z. Feng (2019), “Vibration based condition monitoring and fault diagnosis of wind turbine planetary gearbox: A review,” Mechanical Systems and Signal Processing, vol. 126, pp. 662–685, 2019, doi: 10.1016/j.ymssp.2019.02.051.

[36] A. L’Heureux, K. Grolinger, H. F. Elyamany and M. A. M. Capretz (2017), "Machine Learning With Big Data: Challenges and Approaches," in IEEE Access, vol. 5, pp. 7776-7797, 2017, doi: 10.1109/ACCESS.2017.2696365

[37] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, and A. K. Nandi, “Applications of machine learning to machine fault diagnosis: A review and roadmap,” Mechanical Systems and Signal Processing, vol. 138, p. 106587, 2020, doi: 10.1016/j.ymssp.2019.106587.

[38] R. Liu, B. Yang, E. Zio, and X. Chen (2018), “Artificial intelligence for fault diagnosis of rotating machinery: A review,” Mechanical Systems and Signal Processing, vol. 108, pp. 33–47, 2018, doi: 10.1016/j.ymssp.2018.02.016.

[39] S. Zhang, S. Zhang, B. Wang and T. G. Habetler (2020), "Deep Learning Algorithms for Bearing Fault Diagnostics—A Comprehensive Review," in IEEE Access, vol. 8, pp. 29857-29881, doi: 10.1109/ACCESS.2020.2972859.

[40] J. Jiao, M. Zhao, J. Lin, and K. Liang (2020), “A comprehensive review on convolutional neural network in machine fault diagnosis,” Neurocomputing, vol. 417, pp. 36–63, 2020, doi: 10.1016/j.neucom.2020.07.088.

[41] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman (2021), “1D convolutional neural networks and applications: A survey,” Mechanical Systems and Signal Processing, vol. 151, p. 107398, 2021, doi: 10.1016/j.ymssp.2020.107398.

[42] EN 13848-1 (2019): Railway applications - Track - Track geometry quality - Part 1: Characterization of track geometry.

[43] EN 13848-2 (2020): Railway applications - Track - Track geometry quality - Part 2: Measuring systems – Track recording vehicles.

[44] S. Grassie (1996), “Measurement of railhead longitudinal profiles: a comparison of different techniques,” Wear 191 (1996) 245-251.

[45] Mermec (2022), “Track geometry”, [online], [accessed on 18.01.2022], available: https://www.mermecgroup.com/diagnostictrack-geometry/185/track-geometry.php

[46] RIVAS (2013), “Deliverable D2.5 Overview of Methods for Measurement of Track Irregularities Important for Ground-Borne Vibration,” [online], [accessed on 18.01.2022], available: http://www.rivas-project.eu/fileadmin/documents/RIVAS_CHALMERS_WP2_D2_5_FINAL.pdf

[47] VGTU (2020), Deliverable D8.1 Wheel-rail transversal position monitoring technologies, project report, [online], [accessed on 18.01.2022], http://www.assets4rail.eu/wp-content/uploads/2021/12/A4R-D8.1.pdf

[48] A. Haigermoser, B. Luber, J. Rauh, and G. Gräfe (2015), "Road and track irregularities: measurement, assessment and simulation," Vehicle System Dynamics, vol. 53, no. 7, pp. 878 – 957, doi: 10.1080/00423114.2015.1037312.

[49] P. Weston, C. Roberts, G. Yeo, and E. Stewart (2015), "Perspectives on railway track geometry condition monitoring from in-service railway vehicles," Vehicle System Dynamics, vol. 53, no. 7, pp. 1063–1091, doi: 10.1080/00423114.2015.1034730.

[50] J. L. Escalona, P. Urda, and S. Munoz (2021), "A Track Geometry Measuring System Based on Multibody Kinematics, Inertial Sensors and Computer Vision," Sensors (Basel, Switzerland), vol. 21, no. 3, doi: 10.3390/s21030683.

[51] L. Peng, S. Zheng, P. Li, Y. Wang, and Q. Zhong (2020), "A Comprehensive Detection System for Track Geometry Using Fused Vision and Inertia," IEEE Trans. Instrum. Meas., p. 1, doi: 10.1109/TIM.2020.3039301.

[52] EN 13848-6 (2020), Railway applications - Track - Track geometry quality - Part 6: Characterisation of track geometry quality.

[53] X. Sun, F. Yang, J. Shi, Z. Ke, and Y. Zhou (2021), "On-Board Detection of Longitudinal Track Irregularity Via Axle Box Acceleration in HSR," IEEE Access, vol. 9, pp. 14025–14037, doi: 10.1109/ACCESS.2021.3052099.

[54] DB (2022), „Efficient maintenance: Monitoring the infrastructure using regularly scheduled trains", [online], [accessed on 13.01.2022], available: https://www.db-systemtechnik.de/resource/blob/1669102/4975181f198f9ca6c700e42e772c3ff9/Aktuell_E_flyer-continuous-track-monitoring-data.pdf

[55] C. Ágh (2019), "Comparative Analysis of Axlebox Accelerations in Correlation with Track Geometry Irregularities," Acta Tech Jaur, vol. 12, no. 2, pp. 161–177, 2019, doi: 10.14513/actatechjaur.v12.n2.501.

[56] B. Ripke et al. (2020), "Report on track/switch parameters and problem zones", technical report Deliverable D4.1 of the IN2SMART project, [online], [accessed on 03.12.2020], available: https://projects.shift2rail. org/s2rip3n.aspx?p=IN2SMART

[57] A. de Rosa, S. Alfi, and S. Bruni (2019), "Estimation of lateral and cross alignment in a railway track based on vehicle dynamics measurements," Mechanical Systems and Signal Processing, vol. 116, pp. 606–623, doi: 10.1016/j.ymssp.2018.06.041.

[58] S. Munoz, J. Ros, P. Urda, and J. L. Escalona (2021), "Estimation of Lateral Track Irregularity Through Kalman Filtering Techniques," IEEE Access, vol. 9, pp. 60010–60025, doi: 10.1109/ACCESS.2021.3073606.

[59] Z. Yuan, J. Luo, S. Zhu, and W. Zhai, "A Wasserstein generative adversarial network-based approach for real-time track irregularity estimation using vehicle dynamic responses," Vehicle System Dynamics, pp. 1–20, 2021, doi: 10.1080/00423114.2021.1999480.

[60] E. J. OBrien, P. Quirke, C. Bowe, and D. Cantero (2018), "Determination of railway track longitudinal profile using measured inertial response of an in-service railway vehicle," Structural Health Monitoring, vol. 17, no. 6, pp. 1425–1440, doi: 10.1177/1475921717744479.

[61] X. Xiao, Z. Sun, W. Shen, (2020), "A Kalman filter algorithm for identifying track irregularities of railway bridges using vehicle dynamic responses." Mechanical Systems and Signal Processing, 138, 106582. https://doi.org/10.1016/j.ymssp.2019.106582

[62] C. Li, Q. He, and P. Wang (2021), "Estimation of railway track longitudinal irregularity using vehicle response with information compression and Bayesian deep learning," Computer aided Civil Eng, doi: 10.1111/mice.12802.

[63] F. Balouchi, A. Bevan, and R. Formston (2020), "Development of railway track condition monitoring from multi-train in-service vehicles," Vehicle System Dynamics, pp. 1–21, doi: 10.1080/00423114.2020.1755045.

[64] H. Tsunashima and R. Hirose (2020), "Condition monitoring of railway track from car-body vibration using time–frequency analysis," Vehicle System Dynamics, pp. 1–18, doi: 10.1080/00423114.2020.1850808.

[65] H. Tsunashima (2019), "Condition Monitoring of Railway Tracks from Car-Body Vibration Using a Machine Learning Technique," Applied Sciences, vol. 9, no. 13, p. 2734, doi: 10.3390/app9132734.

[66] S. Kaewunruen (2018), "Monitoring of Rail Corrugation Growth on Sharp Curves For Track Maintenance Prioritisation," IJAV, vol. 23, no. 1, doi: 10.20855/ijav.2018.23.11078.

[67] A. Chudzikiewicz, R. Bogacz, M. Kostrzewski, and R. Konowrocki (2018), "Condition monitoring of railway track systems by using acceleration signals on wheelset axle-boxes," Transport, vol. 33, no. 2, pp. 555–566, doi: 10.3846/16484142.2017.1342101.

[68] A. de Rosa et al. (2020), "Monitoring of lateral and cross level track geometry irregularities through onboard vehicle dynamics measurements using machine learning classification algorithms," Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 095440972090664, doi: 10.1177/0954409720906649.

[69] G.D. Dharani, N.G. Nair, P. Satpathy, J. Christopher (2019), "Covariate Shift: A Review and Analysis on Classifiers," 2019 Global Conference for Advancement in Technology (GCAT) Bangalore

[70] Vapnik, V. (2000). "The Nature of Statistical Learning Theory. Information Science and Statistics". Springer-Verlag. ISBN 978-0-387-98780-4.

[71] I. Gulrajani and D. Lopez-Paz (2020), "In Search of Lost Domain Generalization," Jul. 2020. [Online]. Available: http://arxiv.org/pdf/2007.01434v1

[72] Chen, Q. Li, C. Shen, J. Zhu, D. Wang, and M. Xia, "Adversarial domain-invariant generalization: a generic domain-regressive framework for bearing fault diagnosis under unseen conditions," IEEE Trans. Ind. Inf., p. 1, 2021, doi: 10.1109/TII.2021.3078712.

[73] R. Huang, J. Li, Y. Liao, J. Chen, Z. Wang, and W. Li, "Deep Adversarial Capsule Network for Compound Fault Diagnosis of Machinery Toward Multidomain Generalization Task," IEEE Trans. Instrum. Meas., vol. 70, pp. 1–11, 2021, doi: 10.1109/TIM.2020.3042300.

[74] X. Li, W. Zhang, H. Ma, Z. Luo, and X. Li, "Domain generalization in rotating machinery fault diagnostics using deep neural networks," Neurocomputing, vol. 403, pp. 409–420, 2020, doi: 10.1016/j.neucom.2020.05.014.

[75] Y. Liao, R. Huang, J. Li, Z. Chen, and W. Li, "Deep Semi-supervised Domain Generalization Network for Rotary Machinery Fault Diagnosis under Variable Speed," IEEE Trans. Instrum. Meas., p. 1, 2020, doi: 10.1109/TIM.2020.2992829.

[76] Q. Zhang et al., “Conditional Adversarial Domain Generalization With a Single Discriminator for Bearing Fault Diagnosis,” IEEE Trans. Instrum. Meas., vol. 70, pp. 1–15, 2021, doi: 10.1109/TIM.2021.3071350.

[77] H. Zheng, Y. Yang, J. Yin, Y. Li, R. Wang, and M. Xu, “Deep Domain Generalization Combining A Priori Diagnosis Knowledge Toward Cross-Domain Fault Diagnosis of Rolling Bearing,” IEEE Trans. Instrum. Meas., vol. 70, pp. 1–11, 2021, doi: 10.1109/TIM.2020.3016068.

[78] Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron (2016). Deep Learning. MIT Press. ISBN 978-0262035613.

[79] F. Yang, L. Herranz, J. v. d. Weijer, J. A. I. Guitián, A. M. López and M. G. Mozerov (2020), "Variable Rate Deep Image Compression With Modulated Autoencoder," in IEEE Signal Processing Letters, vol. 27, pp. 331-335, doi: 10.1109/LSP.2020.2970539.

[80] E. B. Helal, O. M. Saad, A. G. Hafez, Y. Chen and G. M. Dousoky (2021), "Seismic Data Compression Using Deep Learning," in IEEE Access, vol. 9, pp. 58161-58169, doi: 10.1109/ACCESS.2021.3073090.

[81] M. Sepahvand and F. Abdali-Mohammadi (2019), "A Deep Learning-Based Compression Algorithm for 9-DOF Inertial Measurement Unit Signals Along With an Error Compensating Mechanism," in IEEE Sensors Journal, vol. 19, no. 2, pp. 632-640, 15 Jan.15, doi: 10.1109/JSEN.2018.2877360.

[82] H. -T. Chiang, Y. -Y. Hsieh, S. -W. Fu, K. -H. Hung, Y. Tsao and S. -Y. Chien (2019), "Noise Reduction in ECG Signals Using Fully Convolutional Denoising Autoencoders," in IEEE Access, vol. 7, pp. 60806-60813, doi: 10.1109/ACCESS.2019.2912036.

[83] M. Russel, P. Wang (2022), “Physics-informed deep learning for signal compression and reconstruction of big data in industrial condition monitoring”, Mechanical Systems and Signal Processing, vol. 168, doi: 10.1016/j.ymssp.2021.108709

[84] J. Sun et al. (2020), “A signal analysis based hunting instability detection methodology for high-speed railway vehicles,” Vehicle System Dynamics, pp. 1–23, 2020, doi: 10.1080/00423114.2020.1763407.

[85] R. Kulkarni, A. Qazizadeh, M. Berg, U. Carlsson, and S. Stichel (2021), “Vehicle running instability detection algorithm (VRIDA): A signal based onboard diagnostic method for detecting hunting instability of rail vehicles,” Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 095440972110205, doi: 10.1177/09544097211020578.

[86] J. Ning, M. Fang, W. Ran, C. Chen, and Y. Li (2020), “Rapid Multi-Sensor Feature Fusion Based on Non-Stationary Kernel JADE for the Small-Amplitude Hunting Monitoring of High-Speed Trains,” Sensors (Basel, Switzerland), vol. 20, no. 12, doi: 10.3390/s20123457.

[87] J. Ning, Q. Liu, H. Ouyang, C. Chen, and B. Zhang (2018), “A multi-sensor fusion framework for detecting small amplitude hunting of high-speed trains,” Journal of Vibration and Control, vol. 24, no. 17, pp. 3797–3808, doi: 10.1177/1077546318787945.