

CRANFIELD UNIVERSITY

BERNADIN NAMOANO

FAULT DIAGNOSIS IN TIME SERIES DATA WITH APPLICATION
TO RAILWAY ASSETS

SCHOOL OF AEROSPACE, TRANSPORT AND
MANUFACTURING
Engineering And Management Of Manufacturing Systems

PhD Thesis
Academic Year: 2017 - 2020

Supervisor: Prof. Andrew Starr
Associate Supervisor: Dr. Christos Emmanouilidis
12 2020

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND
MANUFACTURING
Engineering And Management Of Manufacturing Systems

PhD Thesis

Academic Year 2017 - 2020

BERNADIN NAMOANO

FAULT DIAGNOSIS IN TIME SERIES DATA WITH APPLICATION
TO RAILWAY ASSETS

Supervisor: Prof. Andrew Starr
Associate Supervisor: Dr. Christos Emmanouilidis
12 2020

This thesis is submitted in partial fulfilment of the requirements for
the degree of Doctor of Philosophy

© Cranfield University 2020. All rights reserved. No part of this
publication may be reproduced without the written permission of the
copyright owner.

ABSTRACT

This thesis focuses on diagnosing incipient faults in railway train assets through large data sets. It investigates data mining methods to detect changes in big time series data to enable service by employing three case studies that separately investigate (1) the doors, (2) the engines, and (3) the wheels. Each case study comprises data with different characteristics: the first case study examining engines presents datasets with variable speed and load; the second case study examining door data highlights start-stop characteristics, with discontinuities in the data; and the third case study examining wheel data contains fast transitions from normal to slip behaviours in acceleration and deceleration.

A time series architecture is proposed. It describes key steps for multivariate time series analysis and enables iterative improvement reusing results for the next iteration. A fault diagnosis is made for each case scenario, and procedures for synchronisation and alignment; pre-processing; and methods for feature extraction, classification or clustering are thus presented.

For engine fault diagnosis, the proposed graphical method has the best performance. In the case of the door fault diagnosis, the K-nearest neighbours method has the best performance. In the wheel slip diagnosis case study, the combined wavelet and LSTM methods present the best accuracy. Limitations include data quality issues, key input data uncertainties, and applied classification deficiencies.

The main challenges include big datasets that are desynchronised, wrongly time-indexed, noisy, redundant, and unlabelled; infrequent faults; the lack of monitoring in several subsystems; and incomplete or missing maintenance records.

Keywords:

Machine learning, predictive maintenance, time series, fault, detection, diagnosis, railway, door, engines, wheels, artificial intelligence, data fusion architectures, wavelets, neural network, graphical model.

ACKNOWLEDGEMENTS

The achievement of this study has not been an easy task, especially when moving to a new country and external issues are involved.

First of all, I would like to express my sincere gratitude to Professor Andrew Starr and Dr Christos Emmanouilidis for their assistance and advice throughout this thesis. I am also grateful to them for creating an excellent working environment with many graduate students and postdoctoral researchers. These have positively influenced and enhanced my work. Thanks to Dr Cristobal Ruiz Carcel, Isidro Durazo Cardenas, and Leigh Kirkwood for their support.

I would like to thank Unipart Rail, Instrumentel, and the Engineering and Physical Sciences Research Council (EPSRC), for funding the project and for providing a vast amount of datasets, as well as contextual knowledge. Thanks to Instrumentel's great team members David McGorman, Neil Tinworth, Dave Smith, Chris Fishlock, and Bradley Ayres for sharing their experiences.

I would also like to express my deep gratitude to my family for their help, patience, and support during the more difficult moments of this journey. Many thanks to Veronique, Caroline, Touwend-sida, Marcel, Odile, Bernadette, Christine, and Eric.

I am also extremely appreciative to have received such incredible support from all of my friends and colleagues. Special thanks to Laura Haviez, Joanne Holden, Ancele Yannis, Yousef Haddad, Katherine Gobey, Alexandre Lissy, Hannah Skilton, and Jean Parpaillon for all of their encouragement.

TABLE OF CONTENTS

ABSTRACT	i
ACKNOWLEDGEMENTS.....	iii
TABLE OF CONTENTS	v
LIST OF FIGURES.....	ix
LIST OF TABLES	xi
LIST OF EQUATIONS.....	xii
LIST OF ABBREVIATIONS	xiv
1 INTRODUCTION	1
1.1 Context	1
1.2 Research Aim and Objectives.....	3
1.3 General Methodology.....	4
1.3.1 Global Literature Review	5
1.3.2 Case Studies Domain Review	5
1.3.3 Data collection.....	6
1.3.4 Model Building.....	6
1.3.5 Model Evaluation.....	6
1.4 Published Papers.....	6
1.4.1 Journal Papers	7
1.4.2 Conference Papers	7
1.5 Thesis Structure.....	7
2 LITERATURE REVIEW	11
2.1 Introduction	11
2.2 Existing Maintenance Policies and Strategies	11
2.2.1 Corrective Maintenance	12
2.2.2 Preventive Maintenance.....	12
2.2.3 Summary.....	14
2.3 Big Data Analytics Tools and Comparison.....	14
2.3.1 Definition	15
2.3.2 Big Data Analytics	16
2.3.3 Summary.....	21
2.4 Review of Current Fault Diagnosis Methods in Time series.....	23
2.4.1 Quantitative Methods	26
2.4.2 Qualitative Methods.....	38
2.5 Conclusion and Gap in Knowledge	40
3 PREDICTIVE MAINTENANCE FOR RAILWAY ASSETS	43
3.1 Introduction	43
3.2 Condition Monitoring in the Railway Industry	44
3.3 Doors Basic Operations	44
3.3.1 Classical Motions of Automatic Doors	46

3.3.2 Doors Failures	47
3.3.3 Data collected	48
3.4 Basic Operations of Diesel Engines.....	49
3.4.1 Diesel Engines Operation.....	49
3.4.2 Cooling System	50
3.4.3 GWR and WMT Trains	52
3.5 Wheels-rail Contact.....	54
3.5.1 Wheel-rail adhesion.....	54
3.5.2 Data Collected.....	55
3.6 Data Issues and Challenges	55
3.6.1 Poor Knowledge of the System Behaviour	56
3.6.2 Lack of Maintenance Records.....	56
3.6.3 Lack of Input Parameters and Essential Attributes.....	56
3.6.4 Telemetry Issues and Desynchronisation.....	57
3.6.5 Issues with Noise and Sensor Readings	58
3.6.6 Lack of System Fault in the Datasets	59
3.7 Conclusion	59
4 BUILDING TIME SERIES ARCHITECTURE.....	61
4.1 Introduction	61
4.2 Review of Data Fusion Architectures.....	61
4.2.1 Single Sensor Fusion Architecture	62
4.2.2 Multiple Sensors Fusion Architecture	63
4.3 Proposed Architecture	69
4.4 Conclusion	73
5 DMU DOOR FAULTS DIAGNOSIS	75
5.1 Introduction	75
5.2 Background.....	75
5.3 Proposed Methodology	76
5.3.1 Data and Time Alignment.....	77
5.3.2 Cleaning, Segmentation and Local Features Extraction.....	79
5.3.3 Condition Indicators Extraction.....	81
5.3.4 Clustering	81
5.3.5 Classification Methods	82
5.3.6 Evaluation	82
5.4 Experiments.....	83
5.5 Results and Discussion.....	85
5.6 Conclusion	88
6 DMU TRAIN ENGINES FAULT DIAGNOSIS	91
6.1 Introduction	91
6.2 Related Work	92
6.2.1 Noise and Vibrations Analysis.....	94

6.2.2 Fuel and Oil Analysis.....	94
6.2.3 Exhaust Analysis	94
6.2.4 Speed and Torque	95
6.3 Challenges and Open Research Issues.....	95
6.4 Proposed Methodology	95
6.4.1 Data.....	96
6.4.2 Time Alignment and Cleaning	98
6.4.3 Segmentation	98
6.4.4 Feature Extraction	98
6.4.5 Proposed Annotation Method.....	100
6.5 Experiments.....	107
6.5.1 Algorithmic Parameter Settings.....	107
6.5.2 Compared Methods and Evaluation Metric	107
6.6 Results and Discussion.....	109
6.7 Conclusion	111
7 WHEEL SLIP ACTIVITIES DIAGNOSIS	113
7.1 Introduction	113
7.2 Background.....	115
7.3 Proposed Methodology	116
7.3.1 Data Cleaning and Time Alignment.....	117
7.3.2 Data Segmentation, Feature Extraction and Labelling	117
7.3.3 Pattern Analysis	118
7.3.4 Ensemble Learning Methods.....	120
7.3.5 Evaluation Methods.....	120
7.4 Experiments.....	120
7.5 Results and Discussion.....	121
7.6 Sanding system comparison	124
7.7 Conclusion	125
8 DISCUSSION	127
8.1 Introduction	127
8.2 Tool Selection	127
8.3 Time Series Architecture.....	128
8.4 Case Studies	130
8.4.1 Engine Diagnosis	130
8.4.2 Door Diagnosis.....	132
8.4.3 Wheel Slip Activity Diagnosis	132
8.5 Limitation of the pre-processing analysis: Assumptions, Limitations and Uncertainties.....	133
8.5.1 Data.....	133
8.5.2 Data Synchronisation	134
8.5.3 Pre-processing	134

8.5.4 Segmentation	135
8.5.5 Features	135
8.6 Conclusion	135
9 CONCLUSION	137
9.1 Research Aim and Objectives.....	137
9.2 Contribution to Knowledge.....	140
9.3 Future Work	142
9.3.1 Short-term	142
9.3.2 Medium-term	143
9.3.3 Long-term.....	144
REFERENCES.....	145

LIST OF FIGURES

Figure 1-1 Project scope.	4
Figure 1-2 Layout of the research methodology.	5
Figure 2-1 Overview of basic maintenance strategies.	11
Figure 2-2 Dimensions of big data.	15
Figure 2-3 Big data analytics comparison.	21
Figure 2-4 Designed platform architecture.	22
Figure 2-5 Description of fault diagnosis main objectives.	24
Figure 2-6 Quantitative methods.	26
Figure 2-7 Example of a neural network structure.	30
Figure 2-8 Qualitative methods.	38
Figure 2-9 Example of the structure of an expert system.	39
Figure 3-1 Example of doors on GWR train.	45
Figure 3-2 Sliding Door Motion.	46
Figure 3-3 Sliding Plug Door Motion.	47
Figure 3-4 Swing Plug Door Motion.	47
Figure 3-5 Diesel engine combustion stages (Woodfort, 2014).	50
Figure 3-6 Air cooling system (Hajam, 2014).	51
Figure 3-7 Example of engine cooling system (Hajam, 2014).	51
Figure 3-8 Example of a DMU train schematic design.	52
Figure 3-9 Low power event of a DMU train with three engines.	53
Figure 3-10 Wheel-rail contact with a sanding system (Ricardo, 2019).	54
Figure 3-11 Missing data length over two days during running hours.	57
Figure 3-12 Train DMU engines frequency over time.	58
Figure 4-1 Single sensor fusion architecture.	62
Figure 4-2 Centralised architecture: Feature level fusion.	65
Figure 4-3 Centralised fusion: Pattern level or analysis level architecture.	65
Figure 4-4 Centralised fusion: multiple fusion levels.	66

Figure 4-5 JDL data fusion architecture.	67
Figure 4-6 Architecture of the Boyd model.	68
Figure 4-7 Omnibus data fusion architecture.....	69
Figure 4-8 Novel time series architecture.	70
Figure 5-1 Methodology workflow.....	77
Figure 5-2 Example of a raw data profile of a door opening and closing.	78
Figure 5-3 Example of a cleaned and aligned of a door opening and closing. .	79
Figure 5-4 PCA principal components selection.....	83
Figure 5-5 PCA condition indicators.	84
Figure 5-6 K-means clustering results with the clusters C1, C2, C3, and C4. ..	84
Figure 5-7 Cluster C1 and intermittent fault associated in cluster C4.....	85
Figure 5-8 Cluster C2 and intermittent faults associated in cluster C3.	86
Figure 6-1 Methodology workflow.....	96
Figure 6-2 DMUs engines in carriage C1, C2 and C3.	96
Figure 6-3 Engine frequency and associated charge air pressure over time....	97
Figure 6-4 Clustering graph obtained.	105
Figure 7-1 Wheel slip activity in axle speed profile.....	113
Figure 7-2 Urban DMU train powered axles investigated.	114
Figure 7-3 Sanding system (Olofsson and Lewis, 2006).	114
Figure 7-4 methodology workflow.....	116
Figure 7-5 Angle value computed from the axles speed.	122
Figure 7-6 Scalogram of the wavelet.	122
Figure 7-7 Wheel slip activity during an acceleration phase.....	123
Figure 7-8 Old and new sand system slip number and length.....	125
Figure 8-1 platform architecture.	128
Figure 8-2 Proposed time series architecture.....	129
Figure 8-3 Engine frequency and associated charge air pressure over time..	131

LIST OF TABLES

Table 1-1 Research objectives index.	9
Table 2-1 Data management tools comparison.	19
Table 2-2 Advantages and drawbacks of fault diagnosis methods.	41
Table 4-1 Advantages and drawbacks of data fusion architectures.	73
Table 5-1 Features extracted from door profiles.	80
Table 5-2 Parameter setting for the classification methods.	87
Table 5-3 Validation results with and without split mode before training.	88
Table 6-1 Diesel engines fault and associated signals monitored.	93
Table 6-2 Features extracted from Engine datasets.	99
Table 6-3 Minimal path cost algorithm used.	106
Table 6-4 Parameters setting used for the model.	107
Table 6-5 Parameters settings used in the compared methods.	108
Table 6-6 Results of overall employed datasets.	109
Table 7-1 Methods parameter settings.	121
Table 7-2 Validation results of the classifications methods.	124

LIST OF EQUATIONS

(2-1).....	34
(5-1).....	81
(5-2).....	81
(5-3).....	81
(5-4).....	81
(5-5).....	82
(5-6).....	82
(5-7).....	82
(6-1).....	100
(6-2).....	100
(6-3).....	100
(6-4).....	101
(6-5).....	101
(6-6).....	101
(6-7).....	101
(6-8).....	102
(6-9).....	103
(6-10).....	104
(6-11).....	104
(6-12).....	104
(6-13).....	104
(7-1).....	118
(7-2).....	118
(7-3).....	119
(7-4).....	119
(7-5).....	120
(7-6).....	120
(7-7).....	120

LIST OF ABBREVIATIONS

ADMM	Alternative Direction Methods of Multipliers
API	Application Programming Interface
AR	Auto Auto-Regressive
ARIMA	Auto Auto-Regressive Integrated Moving Average
AUC	Area Under the Curve
BOCD	Bayesian Online Change Detection
CBM	Condition Based Monitoring
CF	Crest Factor
CNN	Convolutional Neural Network
CPM	Continuous Profiles Model
CUSUM	Cumulative Sum
CWT	Continuous Wavelets Transform
DMU	Diesel Multiple Units
DTW	Dynamic Time Wrapping
ES	Expert System
FFT	Fast Fourier Transform
FT	Fourier Transform
GMM	Gaussian Mixture Model
GPIO	General-Purpose Input-Output
GWR	Great Western Rail
HDFS	Hadoop Distributed File System
HMM	Hidden Markov Model
HPC	High-Performance Computer
IID	Independent and identically distributed
IT	Information Technology
JDL	Joint Directors of the Laboratories
K	Kurtosis
k-NN	K-Nearest Neighbour
LSTM	Long Short-Term Memory
MDT	Mean Delay Time
MLP	Multi-Layer Perceptron

NN	Neural Network
OCPD	Online Change Points Detection
OODA	Observe, Orient, Decide and Act
PCA	Principal Component Analysis
PHM	Prognosis and Health Management
PV	Peak Value
RBF	Radial Basis Function
RMS	Root Mean Square
RO	Research Objective
SDG	Signed Digraph
SOM	Self-Organising Map
SPC	Statistical Process Control
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
SWAB	Sliding Windows and Bottom-Up
T ²	Hotelling Test
TOC	Train Operating Company
WMT	West Midlands Trains
WT	Wavelet Transform

1 INTRODUCTION

1.1 Context

Predictive maintenance in engineering industries has been difficult to implement over the past decades. The lack of possibilities to continuously or regularly monitor and assess equipment has continued to be a serious issue (Khairy A.H.Kobbacy, 2008). Traditionally, preventive maintenance has been employed. This has involved scheduling a task to be carried out at specific times using human application. Such methods are costly and less reliable as components can fail between two scheduled dates and unnecessary maintenance may increase the cost and can cause serious deterioration (Mobley, 2002). However, assessing components in service is now possible due to the advancement of sensor technologies now measuring temperature, vibration, and sound. Additionally, sensors have become cheaper and therefore more accessible. However, it also generates significant amounts of data, creating new challenges in terms of data transmission and storage; data processing and insights retrieval; and data presentation, also known as data visualisation (Gandomi and Haider, 2015).

Transmission systems require new physical equipment as well as new norms and protocols to efficiently transport the generated data. Likewise, data storage and management systems have to present new architectures that allow fast data storage and retrieval whilst keeping some traditional properties such as fault tolerance, low latency, and resource consumption.

The main challenges of data processing are to design and implement models that enable multiple source fusion, integrate domain knowledge, and handle uncertainty. Moreover, developed models have to evolve with context and their environment in order to enable the data to be processed instantly, efficiently, and reliably. In order to fulfill these requirements, analytic tools are moving towards distributed architectures and allow real time processing (Rajagopalan and Krovi, 2015).

Data visualisation is also another significant challenge because it needs to integrate factors such as human experience, psychology, and memory; as well as the fact that processing capacities can cause issues with reliability. Further challenges with data visualisation can include visualisation charts that include interpretation bias, as well as underlying algorithms suffering from lack of efficiency in a multivariate data context.

Although broad investigations have been made to tackle the aforementioned challenges, time series analysis still requires attention. It is a growing area with various application domains, including engineering maintenance, cybersecurity, marketing, and wellbeing (P. and Ahmed, 2016). Data generators such as sensors, mobile devices, and cloud-integrated software have grown exponentially, consequently raising the importance of designing new tools, models, concepts, and architectures to efficiently handle and process the temporal data generated either whilst streaming or offline. Traditional time series methods, although widely researched within the finance field and applied mostly to statistics, currently have processing and data management deficiencies within big data analysis.

Focusing on the transportation sector, and specifically on railways, it is essential to use predictive maintenance in order to respond to social and economic demand whilst also improving public image. For example, in order to predict potential failures in maintenance operations, numerous sensors can be placed onto relevant train assets which then gather time series data such as equipment vibrations, temperature, tilt, track data, doors, and signalling and send it remotely to a data warehouse through wireless networks. In order to realise the value of this data, it is critical to transform the data into information which can then be used to make informative and evidence-based decisions for the business. In this study, there is a particular focus on fault detection and diagnosis. This is because the ability to recognise developing faults in a railway vehicle can avoid numerous disruptions to the network and reduce costs by scheduling early repairs. In this scenario, it would therefore be crucial to apply time series diagnosis techniques

to look for unusual patterns in specific events in order to understand the system's current rate of degradation.

1.2 Research Aim and Objectives

The aim of the present work can be explicitly stated as follows:

“Design and investigate the applicability of methods to detect unusual changes in the train-based large time series data in order to enable maintenance operation service support.”

As a result, the following three research objectives (RO) have been set:

RO1: Investigate the literature to select appropriate big time series data tools to store and manage the vast amount of data generated from the train assets. It aims to review the literature and select appropriate tools in order to build platform architecture.

RO2: Develop an architecture for multivariate time series data analysis. In this research objective, the goal is to provide an architectural framework for temporal data analysis enabling efficient decision making.

RO3: Design, implement, and validate methodology using developed architecture in three train asset case studies that examine fault diagnosis of train doors, train engines, and train wheels. It aims to provide suitable methodologies and methods for potential fault diagnosis on multiple unit diesel train engines, doors, and wheels. The architecture developed in **RO2** will serve as a basis for the methodologies used for each case study.

This project overlaps three main research areas, as shown in Figure 1-1. The first area relates to maintenance strategies and policies within engineering systems and how they are applied in transportation systems, especially in the railway industry. The second area is related to maintenance operations used in transportation systems such as the techniques, methods, and processes. The final area examines big data concepts, tools, and techniques for knowledge discovery, commonly known as data mining or machine learning. For a full

understanding of the present fault diagnosis research and context, it is essential to consider their overlapping problems. For this reason, most of the reviews in this thesis are dedicated to these interwoven areas, as it will help to build a comprehensive context of the project and to identify the needs for innovative solutions.

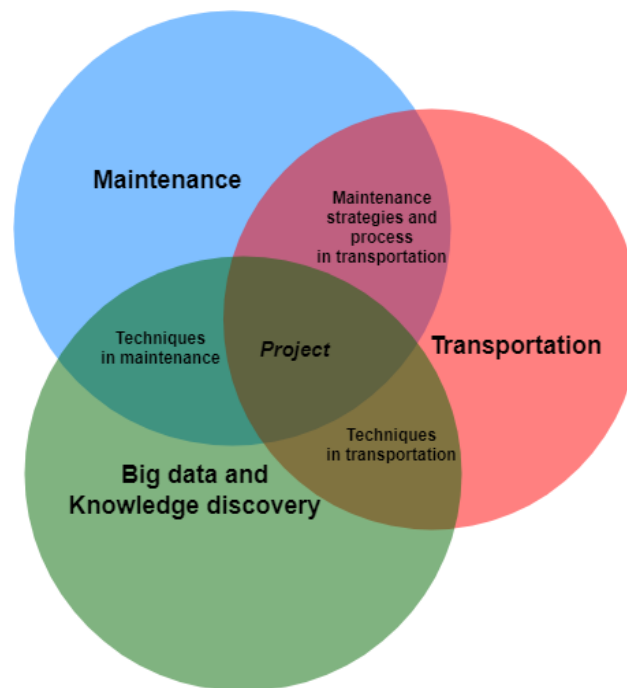


Figure 1-1 Project scope.

1.3 General Methodology

The choice of suitable methodology is leveraged by the nature of the research undertaken, the size of the team, the evolving environment and the time allocated (Kumar, 2019). To align with the research aim and objectives, an adapted condition-based monitoring methodology is used, as shown in Figure 1-2. The motivation for this choice lies in the fact that the project is quantitative in nature analysing numerical data in complex maintenance systems. The study evolved over time and therefore several stages were repeated were required to complete the research objectives. There were five main stages: a general literature review, the case study review, data collection, model building, and model evaluation to fulfill each research objective.

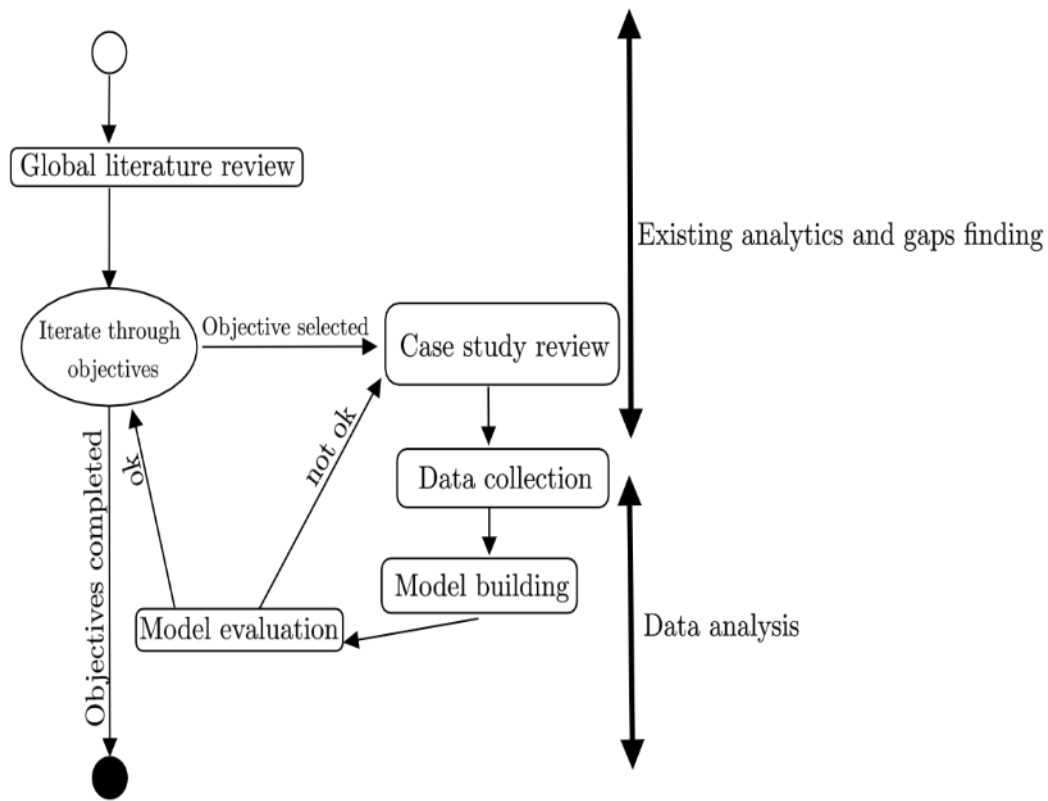


Figure 1-2 Layout of the research methodology.

1.3.1 Global Literature Review

This stage consisted of a broad literature review in order to build an initial understanding of current practices used for time series analysis for big data, such as concepts, assumptions, methods, tools, and infrastructures. Common practices and strategies used in railway maintenance were also researched in order to understand current challenges faced in this domain and past studies exploring railway maintenance were reviewed in order to identify research gaps. The databases used in the literature review were mainly Scopus, Web of Science, IEEE Xplore Digital Library, and Google Scholar.

1.3.2 Case Studies Domain Review

At this stage, the research aims and objectives were set, and a relevant research gap was identified. Moreover, a research objective was selected. As each case

study reviewed different parts of the train, a review for each objective was done for each case study. Similar problems in other disciplines were investigated and an in-depth understanding of the methods used and how they can be applied to this study was achieved. As a result, an additional aim to design novel concepts and techniques to tackle these problems was created.

1.3.3 Data collection

Instrumentel, one of the main sponsors of this work, was the data provider. The data in the three case studies are time series data with limited contextual knowledge such as a lack of data registration and maintenance records. All data was collected following standard data regulation and research ethics and appropriate care was taken to ensure data security.

1.3.4 Model Building

If required, data preparation and pre-processing methods were applied to select, clean, impute, and align the data. These were followed by feature extractions that represent the monitored assets' physical dynamic behaviour. Finally, the fault diagnosis techniques that were designed or selected during the literature reviews were applied to diagnose faults.

1.3.5 Model Evaluation

Key performance indicators were used for the evaluation of the methods. Methods were also evaluated and validated by respective Instrumentel experts.

1.4 Published Papers

During the general literature review stage, it was noticed that there was a lack of recent reviews of streaming change detection. In order to fill this gap, this researcher published a paper providing an overview in time series in a Prognostic and Health Management (PHM) conference (Namoano et al., 2019). After that, during each research objective iteration, due to the complexity of the signals, there was a need to segment the data. Two multivariate segmentation methods were compared using data with different characteristics and the results

were also published in a conference paper (Namoano et al., 2019). Following this, a multichannel annotation method was developed for multichannel time series annotations, which is used to detect usual changes within the channels. The model and the experimentation results were published in an Institute of Electrical and Electronics Engineers (IEEE) Transactions journals.

1.4.1 Journal Papers

Namoano, B., Starr, A., Emmanouilidis, C., & Cristobal, R. C. (2020) 'Graphical model based multiple time series fault detection through annotation', IEEE Transactions on Instrumentation and Measurement, (accepted), p. 1-14.

1.4.2 Conference Papers

Namoano, B., Starr, A., Emmanouilidis, C., & Cristobal, R. C. (2019). Online change detection techniques in time series: An overview. 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), 1–10. <https://doi.org/10.1109/ICPHM.2019.8819394>.

Bernadin Namoano, Christos Emmanouilidis, Cristobal Ruiz-Carcel, and Andrew G Starr, (2020), Change detection in streaming data analytics: A comparison of Bayesian online and martingale approaches, IFAC AMEST 2020, 4th international workshop on Advances in Maintenance, Engineering, Services, and Technology: Maintenance in the Digital Era, Cambridge 10-11 September 2020. <https://doi.org/10.1016/j.ifacol.2020.11.054>

1.5 Thesis Structure

The organisation of the thesis is as follows:

Chapter 1 introduces the context of the work, as well as the main contributions. In addition, it highlights the research aim and objectives and presents the general research methodology undertaken.

Chapter 2 reviews current maintenance strategies and policies, big data and machine learning tools and concepts, and fault diagnosis methods in time series. Moreover, it emphasises challenges and gaps in the current time series fault

diagnosis methods. Section 2.1 introduces the chapter; section 2.2 presents existing maintenance policies; section 2.3 highlights key big data tools and their comparisons; section 2.4 reviews fault diagnosis methods used in the transportation domain with a focus on the railway sector; and section 2.5 is the conclusion.

Chapter 3 describes the context and focuses on the challenges arising in the three types of collected data. In this chapter, section 3.1 is the introduction; sections 3.2 to 3.6 present the basic operations of the chosen assets: the engines, the doors, and the wheels; and section 3.7 is the conclusion.

Chapter 4 reviews traditional engineering data fusion architectures, their main advantages and drawbacks, and proposes a novel architecture that is more flexible for time series analysis. The chapter structure comprises the introduction in section 4.1, review of traditional data fusion architecture in section 4.2, the proposed architecture in section 4.3, and the conclusion in section 4.4.

Chapter 5 presents the analysis carried out on doors which present start-stop characteristics with comparable channels with discontinuities in the data. The chapter structure includes the introduction in section 5.1, the related work in section 5.2, and the proposed methodology in section 5.3. The experiments carried out are presented in section 5.4, the results and discussion are in section 5.5, and the conclusion is in section 5.6.

Chapter 6 describes the analysis carried out on multiple-unit diesel train engines which present a multichannel dataset with variable speed and load. The chapter structure is as follows: the introduction is in section 6.1; the related work is in section 6.2; the open research issues in section 6.3; the proposed methodology is in section 6.4 the description of the experiments carried out is in section 6.5; the results and discussion are in section 6.6; and the conclusion is in section 6.7.

Chapter 7 presents the analysis carried out on wheel slip datasets. It contains fast transitions from normal to slip condition during acceleration and deceleration. The chapter sections are as follows: the introduction is in section 7.1; the related

work is in section 7.2; the proposed methodology is in section 7.3; experiments carried out are described in section 7.4; the results and discussion are in section 7.5; the new and old sanding system are compared in section 7.6; and the conclusion is in section 7.7.

Chapter 8 provides a general discussion of the proposed model and architecture and how to apply it to the engineering time series analysis. It also highlights the limitations and uncertainties of the investigated approaches. Section 8.1 is the introduction, sections 8.2, 8.3, and 8.4 presents the findings and discussion of each research objective. Section 8.5 discusses data analysis assumptions, limitations, and uncertainties. The conclusion is in section 8.6.

Chapter 9 summarises the work presented and reviews the success of the research aim and objectives. Additionally, it presents contributions to knowledge and novelty to the field, and provides recommendations for future short-term, medium-term, and long-term research. The main objectives and achievements of the research are presented in section 9.1, this study's contribution to literature is acknowledged in section 9.2, and suggestions for potential future research are made in section 9.3. Table 1-1 presents the research objectives index.

Table 1-1 Research objectives index.

Research Objectives	Chapters
RO1: Tools selection	Chapter 2 .
RO2: Time series architecture	Chapter 4 , the applications are in chapters 5 , 6 and 7 .
RO3: Case studies fault diagnosis	Chapters 5 , 6 , and 7 .

2 LITERATURE REVIEW

2.1 Introduction

The previous section has described the present research aim and objectives as well as the general methodology undertaken. This chapter reviews main maintenance strategies, big data tools and concepts, and time series fault diagnosis methods. Additionally, the relevant research gap is identified. As this thesis is a collection of different case studies, this chapter focuses on time series analysis in general, rather than focussing on a specific train asset. Each case scenario chapter also highlights a short literature review.

2.2 Existing Maintenance Policies and Strategies

All maintenance is driven by cost minimisation, safety, and reliability (Triantaphyllou et al., 1997). To achieve these goals, maintenance activities themselves should be carried out in a safe working environment and enable low costs, less pollution and waste, and assets to be reliable and permanently available. Hence, there has arisen over recent decades a need for building efficient strategies and policies for maintenance. For this reason, many maintenance strategies and policies have been built and adopted and are all derived from two main basic categories (Lee and Scott, 2009): corrective and preventive maintenance (British Standards Institution, 2010).

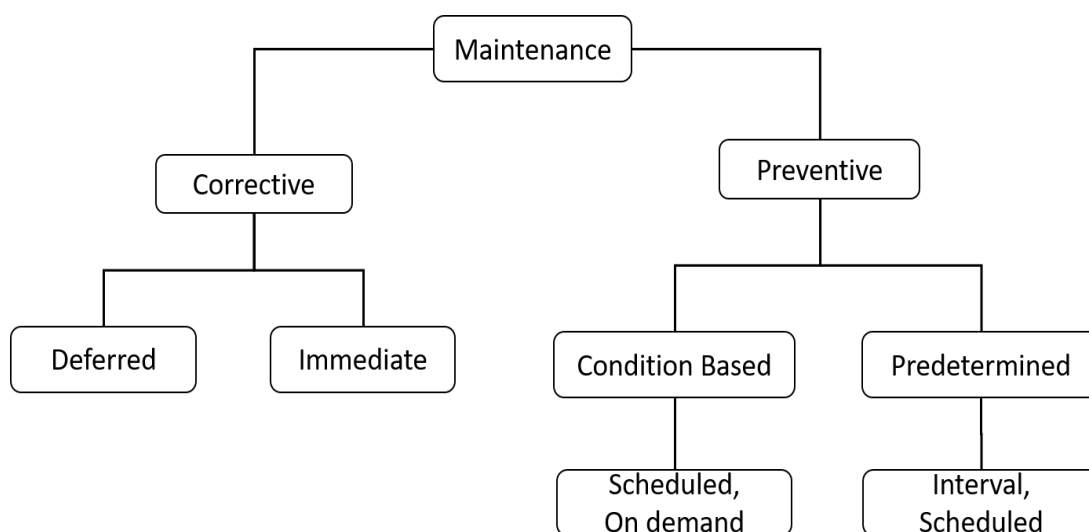


Figure 2-1 Overview of basic maintenance strategies.

2.2.1 Corrective Maintenance

Also known as “run to failure maintenance”, “no maintenance”, or “breakdown maintenance”, it is a straightforward maintenance strategy, where physical assets are repaired or replaced after they have broken down. The reparation can be immediate or delayed, according to the given rules. This strategy is useful for single assets with rarely any faults that have no impact on the system operation. However, such a strategy is not adequate in complex systems, as it may result in severe consequences. For instance, leaving a train engine until failure without any overhauls can result in severe damage and cost many lives.

2.2.2 Preventive Maintenance

Preventive maintenance is carried out either at specific intervals or at a prescribed time, in order to reduce the likelihood of faults or degradation. It comprises predetermined and condition-based maintenance.

2.2.2.1 Predetermined Maintenance

Often called scheduled maintenance, maintenance activities are carried out at given intervals. This approach is generally better than reactive maintenance since it enables some earlier failures to be detected and maintenance actions to be taken. It leads to the reduction of equipment downtime and therefore increases productivity. However, preventive maintenance can include needless maintenance activities; for example, a lot of planned maintenance can result in no detection of failure. As a result, it increases maintenance costs. An additional downfall of the preventive maintenance strategy is fault misdetection. A fault can occur within two consecutive scheduled maintenance activities. Typical examples are fatigue cracks, where the cracks growth distributions are exponential.

2.2.2.2 Condition-based Maintenance

Condition-based maintenance (CBM), often associated with predictive maintenance, reduces most of the preventive maintenance drawbacks. Asset conditions are continuously monitored and evaluated in order to detect incipient failures and hence launch maintenance activities. CBM is a process widely used

in sectors such as manufacturing, aerospace, maritime, and transportation. The main steps of the process for detecting faults are described as follows (Khairy A.H.Kobbacy, 2008).

- **Data acquisition:** in this step, data is collected from various sources. A physical signal such as temperature, visual inspection records, acoustic vibration, and image is converted into digital data.
- **Feature extraction:** in this step, data is processed for useful information extraction.
- **State diagnosis:** this involves a search for any abnormal behaviour or any deviation from the normal or expected behaviour of the system.
- **Health assessment:** this step evaluates the current health level, performs a fault diagnosis, and generates and provides recommendations and explanations.
- **Prognostic:** prognostics essentially estimates the future health state or the remaining life of the component and then predicts incoming faults. Prognostics are often difficult to implement in complex systems because it requires in-depth knowledge of the system's subcomponents life cycle and significant historical fault records for accurate prediction estimations.
- **Decision support:** this step aims to generate recommendations based on the health state diagnosed or predicted of the system.

Drawbacks of predictive maintenance are often the implementation difficulties. For data acquisition, not all of the components can be monitored. The reasons include the location of the components, which do not allow sensor fixtures, and the lack of appropriate sensors. Additionally, in a specific context such as aerospace, global weight limitations of the systems reduce full component monitoring, and poor data transmissions are often observed during flight operations. In general, predictive maintenance is data dependent and based on historical records. Hence, insufficient knowledge or historical data can lead to implementation issues.

2.2.3 Summary

In summary, maintenance strategies aim to reduce maintenance costs whilst extending the life of the assets. Traditional maintenance strategy encompasses reactive maintenance, preventive maintenance, and predictive maintenance. Each strategy has its advantages and disadvantages for each domain that uses it. It is worth noticing that emerging maintenance is still predictive maintenance-based, notably proactive maintenance and self-maintenance. Proactive maintenance, instead of analysing symptoms that lead to failure, focuses on the root causes. Self-maintenance is a new machine design strategy that allows machine self-repairing, diagnosis, and reparation. Both maintenance strategies involve embedded technologies and equipment within a network. This project evolves in CBM context, implemented by a railway supply chain partner: Unipart Rail and Instrumentel.

After defining the maintenance context, the next section defines key notions of the project and seeks an adequate tool for managing and analysing the large amount of data acquired from the sensors.

2.3 Big Data Analytics Tools and Comparison

The last decade has witnessed a phenomenal increase in the amount of data stored in the world, leading to the emergence of the term 'big data'. Growing from a few petabytes (10^{15}) in 2010, the amount of data generated annually has reached 40 zettabytes (10^{21}) per year today. This is linked to the improvement of equipment and storage conditions. Unfortunately, these data have no benefits unless they are transformed into meaningful values for the business. Therefore, the whole amount of data needs to be analysed for useful information retrieval. This section introduces key notions of the term 'big data' and presents common tools used. The tools are compared, emphasising their advantages and weaknesses.

2.3.1 Definition

The term “big data” arises when traditional methods are not sufficiently able to capture, manage, and process the significant amount of data generated. Literature usually defines big data with 5V concepts (Gandomi and Haider, 2015), as shown in Figure 2-2:

Volume: the amount of data captured is tremendous. Data such as videos, images, emails, sensory data, and documents are generated every second, in the magnitude of zettabytes for companies such as Amazon, Google, and Facebook. Around 120 exabytes (10^{19}) of data are generated per day in the world.

Variety: the datasets have various types and come from various sources. Data can be structured, semi-structured, and unstructured (Chawda and Thakur, 2016) such as clickstreams, videos, files, images, audio, documents, and transactions. Data sources are mostly from the web and social media and can be both machine-generated and human-generated; this includes companies’ data transactions, biometrics, and simulations.

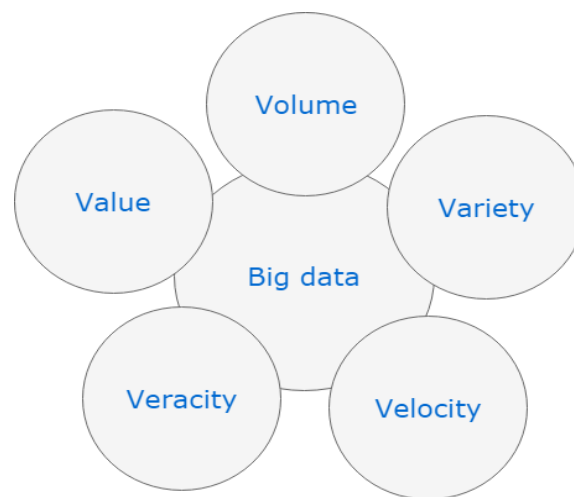


Figure 2-2 Dimensions of big data.

Velocity: it represents the growth rate and how often the data are gathered (P. and Ahmed, 2016). For instance, every second, a vast amount of data-related emails, photos, videos, music, and tweets are generated and analysed by social

media. As a result, the speed of transmitting, storing, accessing, and processing needs to be instantaneous to fit user demand.

Veracity: it refers to data uncertainty or trustworthiness. Unfortunately, a considerable amount contains noises, bursts, and missing values, which increases the level of uncertainty.

Value: it is the benefit that can be gained from the data stored, how this data can be appropriated, and how it can be turned into valuable information. Regarding the value, the most important aspect is to understand the costs and the benefits of gathering and analysing data and then ensure that benefits can be derived from it.

Beyond the concepts, a straightforward definition of big data is all the techniques, methods, and tools that allow a vast amount of data management and processing.

2.3.2 Big Data Analytics

Often defined as the application of advanced analytics tools and methods in a big data context, big data analytics are mostly composed of different software-embedding complex processes. Four groups of tools arose from the literature, namely data storage, data management, data analysis and visualisation, and decision making. The present review focuses only on the first three tools, as decision-making tools are beyond the scope of the study.

Data storage aims to store a considerable amount of various data efficiently and keep it available. The data storage system includes file system-based storage, not only SQL (NoSQL), and traditional transactional databases. The popular systems are presented below.

- **Hadoop Distributed File System (HDFS):** it is a distributed file system designed for large-scale data storage with streaming access, working on commodity hardware. It does not necessarily require a special hardware characteristic such as high-performance computing (HPC). It has a master-worker architecture (White, 2015). The master, also known as

namenode, manages the file system tree and the metadata of files and directories under the tree. The worker (datanode) stores and retrieves data and reports to the master information related to the data.

- **NoSQL databases:** it stores data from different structures with redundancy. Examples are key-value databases, document databases, row or column-oriented databases, and graph-based databases.

Relational databases: many traditional transactional databases are used in the big data context. Typical examples of relational databases are Oracle, MySQL, Microsoft SQL server, and MariaDB.

Data management: The data management tools described here are widely used in industries and the research sector.

- **MapReduce** (Foundation, 2016): a tool based on the map-reduce concept. The mapping phase consists of parsing and filtering the raw data and retrieving key-value pairs. The reduce phase takes as input the pairs generated in the mapping phase and then aggregates it with parallel processing to produce other key-value pairs. The MapReduce tool is a batch job-processing engine connected with the HDFS data store, allowing fast processing of massive data.
- **Spark** (Spark, 2017): this is an Apache open-source software tool initially developed at the University of California. It is a MapReduce based software that improves batch-processing speed, adding additional features such as real time processing, and is able to connect to data stores other than HDFS. Moreover, Spark has in-memory storage that allows fault-tolerance without replication.
- **H2O** (H2O, 2016a): an open-source framework designed for distributed, parallel, and streaming processing. It includes analytics for real time data mining. Like Spark, it has in-memory processing. However, H2O does not provide batch processing, but the H2O sparkling water project aims to add this feature.

- **Flink** (Flink, 2013): a real time processing engine, which essentially deals with unbound datasets. Unbound datasets are appended continuously. Unlike Spark or MapReduce, Flink is not a Hadoop-based system. It has its own runtime environment and is fault-tolerant. It also provides an application programming interface (API) for streams and SQL-like queries. It is widely used to manage, process, and analyse machine historical data in real time, such as physical sensors providing measurement, user interactions with mobile and web application.
- **Storm** (Manning, 2004): an Apache big data framework for real time processing. Focusing on very low latency implementation, it works on the concept of streams, spouts, and bolts. Spouts are the data sources, which are various. Examples include APIs, messaging queues, and datastores. The bolts typically represent a processing step that consumes the streams. A disadvantage of Storm is that there is no guarantee in the streams' processing order. In order to overcome the stateless problem, an abstract feature called trident is developed which aims to alter processes to add a respective state. It increases the latency and slightly changes the Storm processing dynamics.
- **Kafka** (Garg, 2013): Apache Kafka is a combination of messaging, storage, and stream processing framework. It combines distributed systems storage strategies and traditional messaging technologies, allowing fault tolerance by data replication. It also allows streaming processing through a system of event subscriptions. With a distributed and robust messaging queue, it can manage a high volume of data and enable the transfer of the data from one point to another. Kafka and Storm can be seen as complementary tools as Storm allows distributed queue handling and Kafka allows distributed and parallel data processing.

Table 2-1 provides a summary of the data management tools analysed with their supported programming languages, the support, their related latencies, and the engine models.

Table 2-1 Data management tools comparison.

Tools	Version	Input data sources	Supported languages	Enterprise Support	latency	Engine Model
MapReduce	3.0	HDFS	Java, python	No	high	Batch
Spark	2.2	Datastore / streams	Java, Python, R, Scala	Yes	low	Streaming, Batch
Flink	1.4	Datastore / streams	Java, Scala	No	low	Streaming, Batch
Storm	1.2	streams	Any	No	low	Streaming
H2O	3.18	Datastore	Java, Python, R, Scala	yes	low	Batch
H2O Sparkling Water	2.2	Datastore/streams	Java, Python, R, Scala	yes	low	Streaming, Batch
Kafka	1.1	Streams	R, Python, Scala	yes	low	Streaming

Data analysis: Most of the statistical analysis tools developed are embedded as libraries in the programming languages. One may notice that the tools and software keep evolving. Whilst new features are added over time, some become deprecated, which makes the comparison more difficult. The present comparison is based on the most up to-date version at the time of the thesis writing.

The analysis packages presented here are a group of tools containing machine-learning methods.

- **Mahout** (Manning, 2004):): Apache Mahout is a scalable machine learning algorithm library working with the Hadoop environment and implemented using the map-reduce paradigm. It mainly supports four categories of machine learning algorithms, which are collaborative filtering algorithms, clustering algorithms, classification algorithms, and frequent itemsets meaning. Typical data management tools that can be used with Mahout are MapReduce and Spark.
- **MLlib** (Spark, 2016): a data-mining library developed under Spark that provides more machine learning algorithms in comparison to Mahout. Along with algorithms, MLlib provides methods for basic statistics analysis such as correlation analysis and hypothesis testing; classification and

regression algorithms such as linear regression and logistic regression; support vector machine (SVM); collaborative filtering; clustering such as K-means; dimensionality reduction and feature extraction; and transformation.

- **SAMOA** (SAMOA, 2017): it is a framework for distributed data streams analysis that provides an abstraction for machine-learning algorithms. SAMOA is developed under Storm but can also be used with Apache SAMSA, S4, and Apache Kafka. Implemented algorithms are mostly classification and regression algorithms, clustering algorithms, and some meta-algorithms.
- **H2O ML** (H2O, 2016b): a machine-learning library developed under H2O that focuses on distributed analysis, mainly deep learning algorithms, Random Forest algorithms, and linear models. It supports languages such as Python, Java, R, and Scala and can be used with Spark or Storm.
- **TensorFlow** (TensorFlow, 2017): initially developed by Google Brain Team, TensorFlow is an open-source machine learning library focusing on high-performance numerical computing. It allows deployment of computation on various platforms such as servers and mobile devices in a standalone or clusters. It is compatible with Kafka or Spark for data analysis. Implemented algorithms include deep learning algorithms such as neural networks algorithms, clustering, and regression algorithms.

Figure 2-3 below describes the possible connection between the machine-learning libraries, the programming language, and the data management and processing engine. For instance, Spark can be used with the machine learning libraries 3, 4, 5, 6, and 8, which are H2O ML, MLlib, Mahout, Oryx, and TensorFlow. It can also be used with the programming languages J, P, R, S, and M, which are respectively Java, Python, R, Scala, and Matlab. For instance, taking a machine-learning library such as Mahout, implemented tasks are A, B, and C. These are respectively classification, clustering, and association analysis.

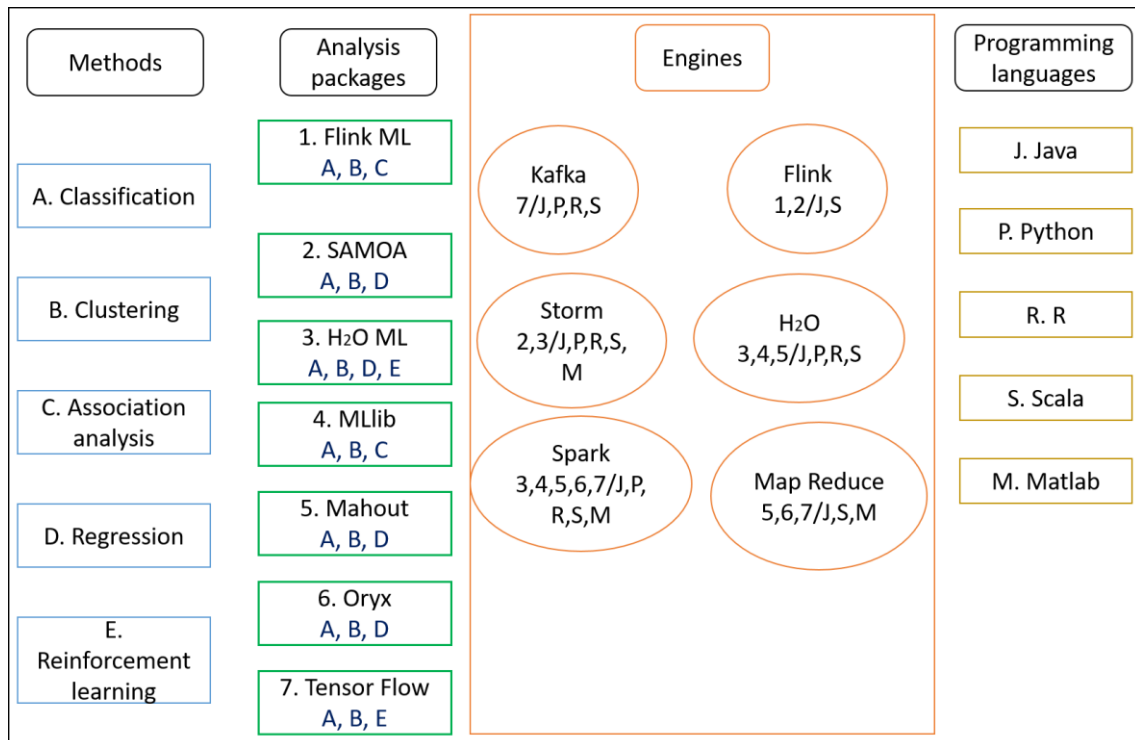


Figure 2-3 Big data analytics comparison.

2.3.3 Summary

Various tools for data storage, management, and analysis exist. The present review covers several widely-used tools. These tools are often combined to build a robust data system enabling, depending on the contextual requirement, fast acquisition, fault tolerance, different data types storage, fast processing, and data visualisation at low costs. In the present work, there was a need to build an open-source system that is able to continuously store, manage, and offer analysis of different characteristics of the temporal data generated. Data are mainly the monitoring of different train, track, signalisation, and pantograph assets as well as the ambient weather conditions, the vegetation, and possible disturbances near the tracks.

From the above-mentioned tools, a platform has been built using the designed architecture shown in Figure 2-4 below. Each record read from a train asset (also called producer) is published in a Kafka stream record called topic. In a Kafka topic, records represent time-indexed key-value pairs. The data in Kafka are

processed by spark-streaming instances acting as consumers. Part of the spark-streaming consumers include the online diagnosis module, where fault diagnosis procedures are implemented. Selected data and metadata generated by consumers are used by the spark-batch for the offline processing module and also for storage in databases (mainly Hadoop).

The selected tools are mainly Apache Spark, Kafka, and HDFS. The selection of these tools is mainly due to their maturity (less bugs), their wide use offering broad online support, the possibility of storing a wide range of data structures, the ability to process the streaming data offline as well as online, and finally, their low costs. Moreover, they offer more flexibility by enabling the service exchange with external tools, the application of various machine learning algorithms, and the possibility to jointly use programming languages. The remaining analysis work of the thesis focuses on the diagnosis module (offline and online) and the visualisation, where most of the analysis is made.

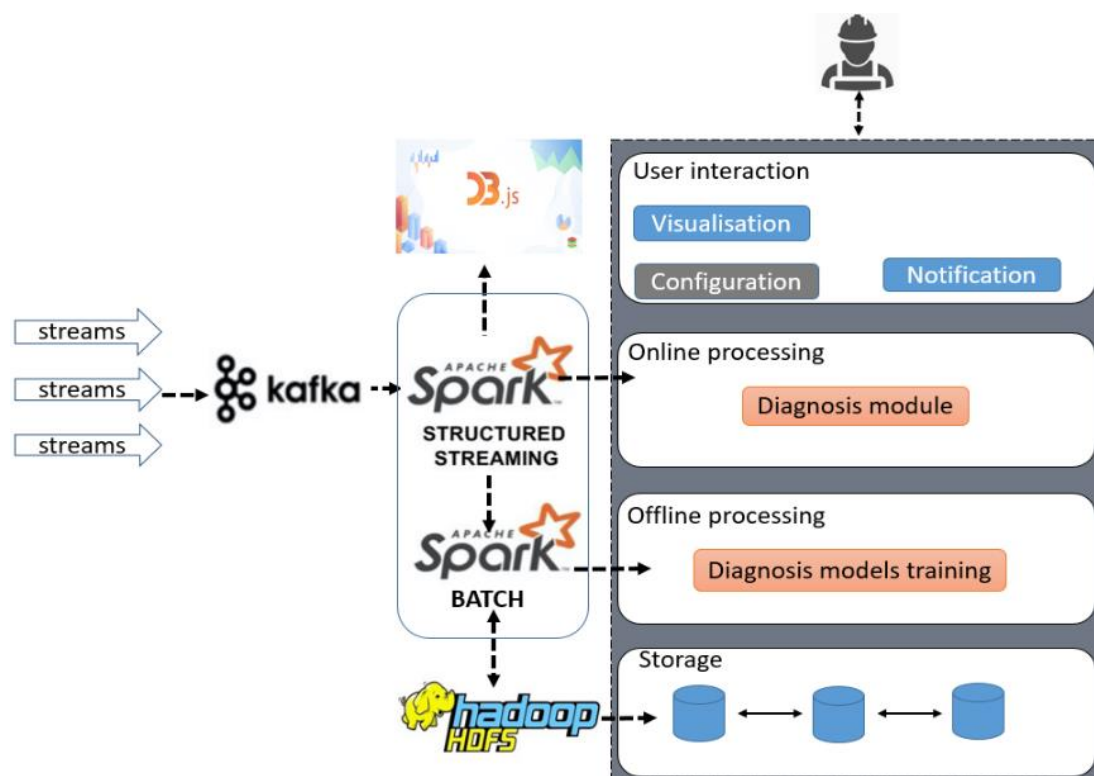


Figure 2-4 Designed platform architecture.

The next section reviews main fault diagnosis methods in time series that are applied in the transportation domain in general, as well as focussing on the railway domain.

2.4 Review of Current Fault Diagnosis Methods in Time series

In this section, key definitions of the notion of fault, fault diagnosis, and time series are first presented, followed by a review of current diagnosis methods. Next, challenges and relevant gaps in knowledge are highlighted. It is worth mentioning that the reviewed methods are for basic individual faults instead of multiple combined faults. Multiple faults diagnosis is out of the scope of this work, even if there is often a combination of the individual faults for multiple fault diagnosis. In addition, the review focuses on the railway domain with extension to transportation areas such as automotive, maritime, and aerospace. The reason is that they often share similar components such as bearings, engine types, and actuators.

Fault: many definitions of the notion of a fault have been suggested in the literature (Aggarwal, 2013). Depending on the application, a fault is often assimilated to anomalies, abnormalities, errors, surprising observations, unexpected behaviour, outliers, extreme values, or anomalous change. Although these terms may have a different meaning, in an analysis point of view, they all refer to deviations from a normal or acceptable condition arousing suspicions that there is a modification in the underlying mechanism. In the engineering domain, a fault usually represents a system or component degradation, which may lead to its inability to achieve its goal in service. In general, three types of faults can be observed. The first type of fault is transient faults. Also known as temporary faults, they occur once and disappear. The second type of fault is intermittent faults. It keeps appearing and disappearing over time. The third type of faults is permanent faults, where a fault appears and will not disappear until it is fixed.

Fault diagnosis: it is a discipline with four main concerns: fault detection, fault isolation, fault classification, and root causes, as described in Figure 2-5. Assuming for instance, that a mistake in a sentence is a fault and considering this sentence:

“The weathers is great today”.

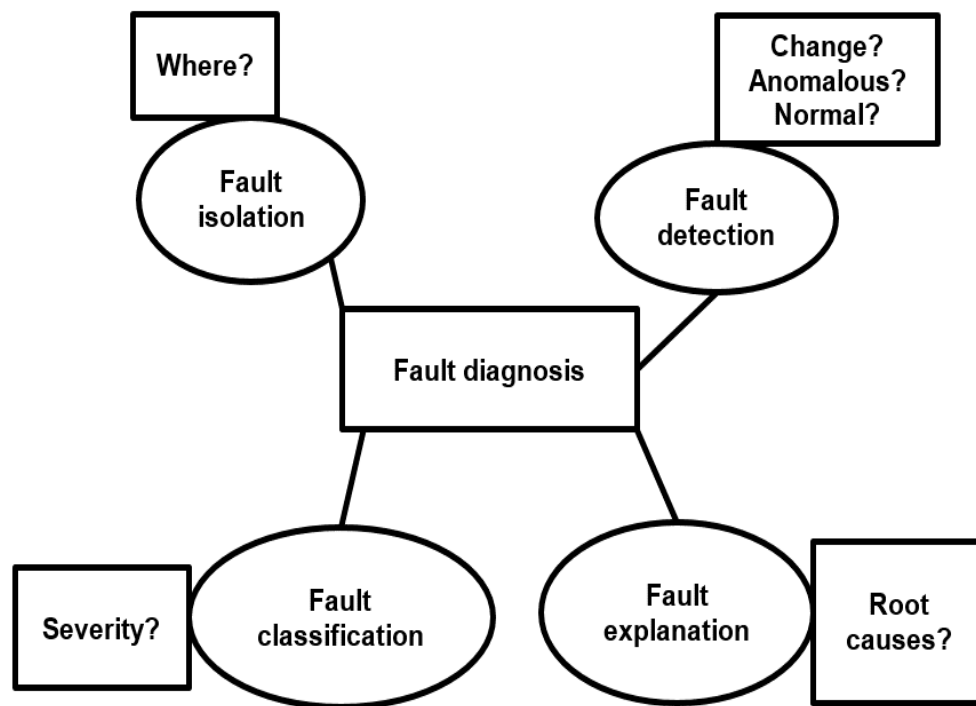


Figure 2-5 Description of fault diagnosis main objectives.

- **Fault detection:** it aims to devise methods and processes to decide whether a fault is present or not in the given observations. In the given sentence, a fault is present.
- **Fault isolation:** its main goal is to find the location of the detected fault. In the example, the fault is located in the word “weathers”, where mistakenly an “s” is added.
- **Fault explanation:** it seeks for root causes of the detected fault. In the example, a root cause is “typographical error”.

- **Fault classification:** After a fault is detected, an estimation of its severity is often required. Severity estimation is contextual and may depend on how it affects the operation of the system. In the given sentence, the fault may not be critical, as it does not modify the meaning of the sentence.

Time series: also known as temporal or time-varying observations, these can be defined as a time-ordered sequence of observations often taken at regular intervals. Important properties arise from such data leading to specific methods development for fault diagnosis. There are:

Stationarity: it refers to time series where the statistical properties such as mean, variance, and autocorrelation do not change over time.

Seasonality: seasonal time series is a time-indexed sequence of observations containing a specific pattern at a regular time interval.

Periodicity: also known as cyclical time series, it is an ordered repetitive unique pattern.

Trends: it refers to an increase or a decrease over time of observations taken.

Abrupt changes: it represents how fast variations occur in the sequence. It is also known as the frequency of the observations.

Whilst time series data may contain such properties, two types of analysis can arise:

Offline analysis: in an offline setting, the whole time series observations are available prior to investigations. More accurate key properties can therefore be computed.

Online analysis: also assimilated to real time, near time, or linear time series analysis, it refers to investigations where not all of the observations are available yet, and the decision has to be made given all observed data so far.

Although there are no hard boundaries between the following classification made, fault diagnosis methods in time series can be classified into two main classes, namely quantitative and qualitative methods.

2.4.1 Quantitative Methods

Two classes of quantitative methods can be distinguished. There are model-driven and data-driven methods. Each class is also divided into subclasses, as shown in Figure 2-6. The following section describes each subclass.

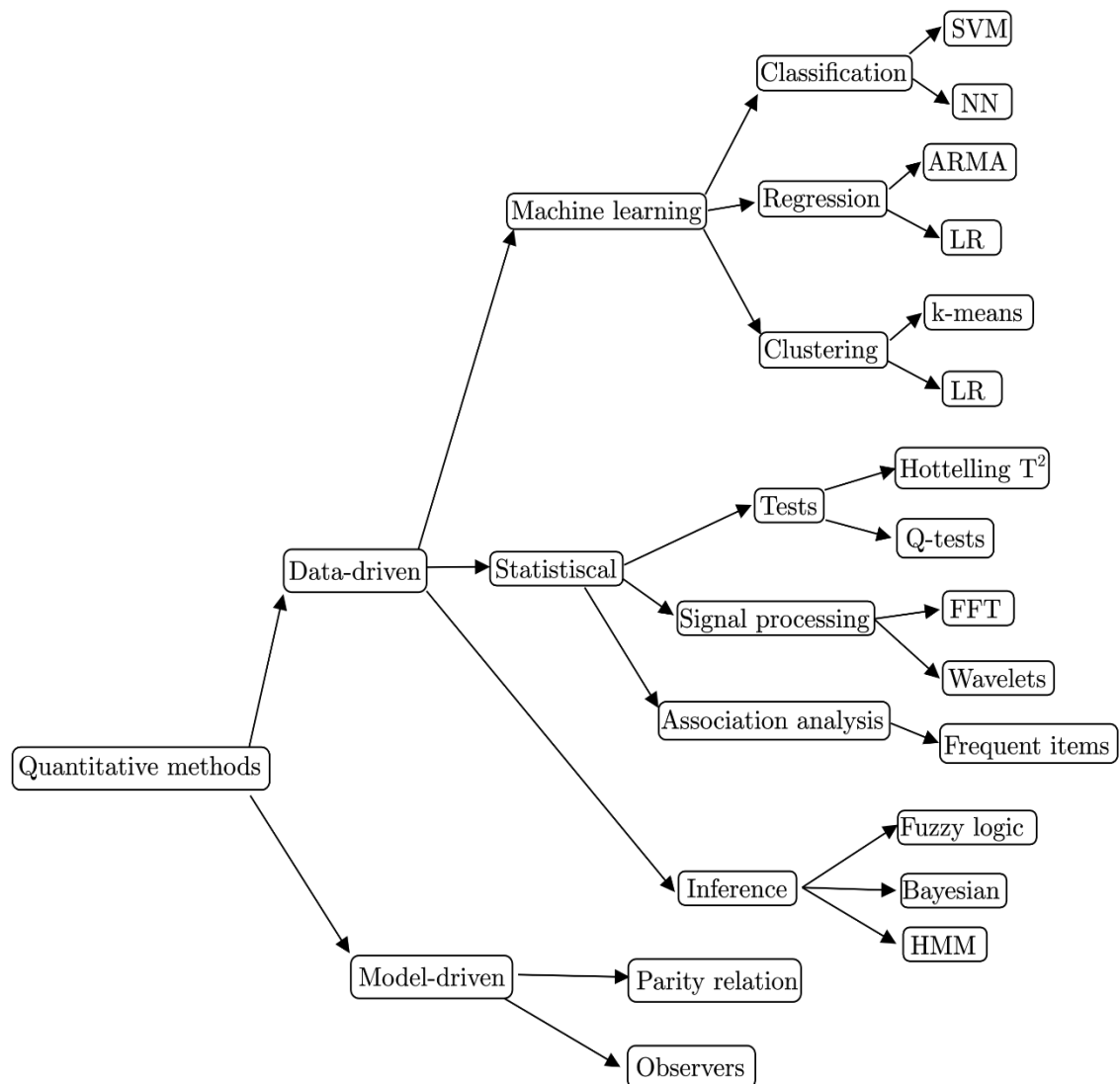


Figure 2-6 Quantitative methods.

2.4.1.1 Model-driven Methods

Model-driven methods aim to design a model of the studied system. This model is then used to generate the expected system response. With both system response and the expected system response obtained from the model, a residual analysis is made to diagnose faults. For efficient fault diagnosis, the designed model should sufficiently reflect the behaviour of the studied system. The main advantage of model-driven methods is their ability to operate online with a limited amount of data in the streams. The main drawbacks of such models is their scalability and their implementation challenges in complex systems. For example, in the case where a component has to be replaced in the designed system, the model may need to be updated to take into account the substitution of the new component.

In literature, much research has been done in the line of engineering model-driven methods. Nyberg and Frisk (2019) provided a comprehensive overview and materials in the algebraic point of view of these methods. Moreover, Martinez Guerra and Mata Machuca (2005), and Gertler (2017), provided analytic and tools with real application domains. Comprehensive reviews include Patton and Chen, (1991), and Venkatasubramanian et al., (2003). In model-driven methods, three main categories exist:

Observers: observer-based methods aim to produce a set of robust residuals, which are each able to detect and identify different faults. The robustness is very important, as the model should be insensitive to disturbances such as noise and sensor measurement imprecisions. In the presence of a fault, only the observer that is sensitive to the fault shows significant residual deviation, leading to detection and isolation. Numerous investigations have been undertaken by applying observer-based models to real-world engineering problems. In the automotive industry, the analysis includes engine suspension (Jesussek and Ellermann, 2014) where Kalman filters were applied to diagnose full-scale railway engine suspension; engine fault detection Dong et al., (2015); Belyaev et al.,

(2018); and active wheelset control (Hong Li, 1998), and Strano and Terzo (2019).

Parity Relation: parity relation-based methods are input-output state-space models used to generate residuals, also known as parity vectors, for fault diagnosis. The underlying idea is that the residual generated should be zero in fault-free cases and non-zero otherwise. In real-case application, more analysis needs to be done in the parity vector. In fact, due to the presence of noise, sensor reading issues, and model inaccuracies, the resulting parity vectors are non-zeros. Recent application parity-based methods in automotive include engine batteries health monitoring analysis (Zhang et al., 2011), diesel engine fault detection (Krishnaswami, Luh and Rizzoni, 1995; Simani, 2013), linear sequential system analysis with partial inputs (Zhong, Song and Ding, 2015), electromechanical brake systems analysis (Cui and Sun, 2019), and closed-loop system fault detection (Nyberg and Frisk, 2019).

In this research context, due to the limited availability of key system input knowledge and data, the model-driven methods are not applicable. This led to the exploration of data-driven methods that are described next.

2.4.1.2 Data-driven Methods

Data-driven methods have gained a lot of attention with the increasing availability of data. Unlike model-driven methods, they do not require mathematical modelling of the studied system to diagnose a fault. Therefore, they allow a better scalability of the models. Important factors of efficient data-driven models rely on the quantity as well as the quality of the monitored data. In addition, domain knowledge is key as it helps to enhance the model through either feature selection or model selection. Data-driven methods often suffer from uncertainties related to the data, model accuracies, and their complexity leading to the inability of the models to operate fully online. Most data-driven methods include machine-learning methods, which are composed of regression, classification, and clustering methods. Additionally, they include statistical methods which are

composed of statistical tests, signal processing, and association rule methods, and inference-based methods. Inference based methods are composed of fuzzy logic, Bayesian, hidden Markov models (HMM) based methods.

2.4.1.2.1 Machine Learning

Recent discussions about machine intelligence have focussed on how it could improve productivity, health, and wellbeing. Machine learning appears to be a way of saving lives, improving business, and optimising resources, as well as tackling many challenges such as climate change, general waste, and security. It can be defined as the technology that allows computer programs to learn and evolve through experiences. Often assimilated with artificial intelligence, data mining, computer vision, computational intelligence or knowledge discovery, it is still a growing area with many branches of analytics. The following section describes common categories of techniques used in the transportation systems and especially in the railway domain.

Classification Methods: also known as supervised learning, consists of the instance assignments from a given domain set, defined by a set of continuous or discrete attributes, into a predefined discrete data class also called the target class (Rajagopalan and Krovi, 2015). A classification method learns from a specific dataset called a training set and builds the models that would serve for future instance class assignments. Prior to real classification phases, these models are validated using specific data generally assimilated to the validation set. For fault diagnosis, data are often labelled normal and faulty before application of the learning model. A major problem that classification methods suffer from is imbalanced data learning. In fact, most real-world condition monitoring data are fault-free, resulting in significant disproportions between the number of observations belonging to the normal class and those that are faulty. This creates bias during the learning phase, leading to the inability of the resulting model to separate normalities to fault during a real case scenario. In a time series context, this creates an additional challenge as techniques such as normalising

may cause a loss in the temporal structures of the observations. Moreover, classification algorithms suffer from their inability to detect an unseen fault during the training phase of the model.

Whilst various classification techniques can be found in the literature, recent years have witnessed a growth of two main groups of methods: neural networks (NN) and support vector machines (SVM).

- **Neural Networks**

It is one of the most used data-driven methods in complex systems. Inspired by biological neural system operation, a neural network's structure is composed of multiple processing units called neurons, often organised into layers. Figure 2-7 below provides a straightforward structure of neural networks. The input observations are weighted and summed together through a transfer function. The results of this computation and thresholds values are used as an input of a defined activation function to decide whether the neurons should be activated or not. The activation function often outputs values within the interval $[0, 1]$ also known as the firing rate.

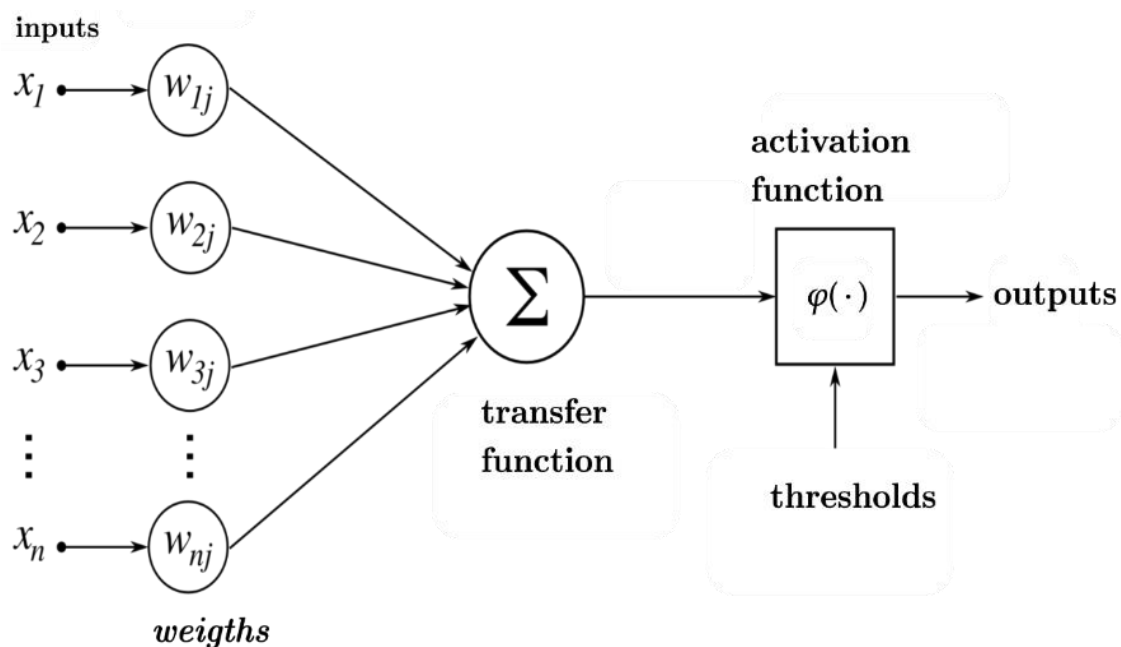


Figure 2-7 Example of a neural network structure.

From such a structure, the main goal is to estimate the suitable weights, transfer function, and activation function that will maximise the separation of the target classes throughout a learning process, using appropriate training data. Several learning methods exist in the literature. Typical examples are back-propagation and gradient descent methods.

For fault diagnosis in the railway domain, neural networks-based methods have been used to improve equipment conditions. In recent work, Lu and Shen (2020) applied a back-propagation neural network for track spikes detection, which helped to reduce labour costs. In a similar area, James *et al.*, (2018), CHEN *et al.*, (2019), and Yuan, Zhu and Zhai (2020) implemented a deep convolutional network for track damage detection. Further research includes fault diagnosis for pipeline inspection (Santur, Karakose and Akin, 2018), and cracks characterisation (Rowshandel *et al.*, 2018). An extensive overview of neural network based applications in engineering fault detection comprises research undertaken by (Liu *et al.*, 2018), (Mohd Amiruddin *et al.*, 2020), and (Jiao *et al.*, 2020).

Challenges in neural networks include classical classification implementation problems already mentioned above, as well as the difficulty in designing the adequate neural network structure during the learning process. In fact, investigations still need to be undertaken to fully understand how they work instead of simply using the neural networks models as a black box.

- **Support Vector Machine**

Also widely used, support vector machine (SVM) methods operate by building a set of hyperplanes that can be used for observation separation into classes. Classes can be finite, discrete (classification), or continuous (regression). Throughout a learning process from training observation sets, SVM methods optimise cost functions, where relevant coefficients are estimated to build the structures of the hyperplanes. For fault diagnosis, SVM methods embed generalisation capabilities, which allow efficient separation between normal and abnormal observations.

SVM based fault diagnosis methods include rotor bar analysis and rolling bearing faults (Seryasat et al., 2010), (Zhang et al., 2015), and (Lin et al., 2019). In addition, extensive reviews of SVM methods encompass Maraini and Nataraj (2015), Cui and Lv, (2019), and Xian and Zeng (2009).

Like neural networks, SVM based methods are often difficult to implement as models can be very complex. Therefore, the optimisation solvers are unable to find global maxima enabling better class separations.

Regression methods: can be seen as classification methods with continuous classes. Fassois and Sakellariou (2007) provided a comprehensive review of regression methods used for time series fault diagnosis in vibration structures. Methods applied include linear regression, exponential regression, autoregressive moving average, and autoregressive integrated moving average (ARIMA). Vagnoli, Remenye-Prescott and Andrews (2018) reviewed structural health monitoring in railway bridges. They showed that robust Bayesian regression methods had been used for bridge fault detection through deviation analysis. The main drawbacks of regression-based methods are issues pertaining to overfitting and underfitting and suitable regularisation parameters are often required to improve model accuracy.

Clustering Methods: The concept of clustering is close to classification methods, but differs from it by its lack of predetermined target classes for assignment. The methods consist of creating subsets, also known as clusters, of observable data based on their similarity or dissimilarity and then using them to build a model. Hence, the model can be used to associate new observation to these clusters. Similarity and dissimilarity measures are often distance or probabilistic based. Yoon *et al.*, (2016), and Du *et al.*, (2019) proposed dynamic time warping based clustering methods to detect railway point machine faults. For railway vehicle suspension faults detection, Yin and Huang (2015) proposed fuzzy C-means clustering methods. A comprehensive survey of clustering methods applied to the railway domain is proposed by Komalan and Chauhan (2017). Description of applied clustering methods also include K-means and fuzzy logic-based

algorithms. Growing areas of univariate time series clustering algorithm that can be used for fault diagnosis are shapelets analysis (Beggel et al., 2018) and matrix profile (Yeh, Kavantzias and Keogh, 2017), (Yeh et al., 2018), in which a unified view of motif discoveries as well as anomaly detection can be made.

The main drawback of clustering methods is their inability to explain the root causes of the fault. Furthermore, most clustering methods are distance-based, making the clustering methods less robust to noise.

2.4.1.2.2 Statistical Methods

Statistical methods are part of the traditional techniques used for fault diagnosis. Mainly used for analysing mechanical systems such as bearing, gearbox, and actuators in general, statistical methods help to diagnose fault through a combination of relevant features that can reflect faulty and healthy conditions. Representative statistical methods include classic tests often used for process control, signal processing methods, and association analysis methods.

Statistical tests: are traditional tools for fault detection. They have mostly been used in the statistical process control (SPC) domain in which upper bound and lower bound thresholds are estimated from the processes historical data. Depending on the context, features extracted are often related to means, the standard deviations, the variances, the correlation coefficients, the kurtosis, the gradients, the overshoots, the peaks, the root mean squares, and the crest factors. Ji *et al.*, (2017) detects incipient faults using jointly smoothing techniques and SPC methods. Ruiz-Carcel and Starr (2018) presented a methodology for diagnosing different stages of fault in linear actuators by using the Hotelling test and the Q-test to control the upper and the lower bound estimation, respectively. Any deviations from the intervals are characterised as faults. Jin, Fan and Chow (2019) applied multivariate statistical process control to diagnose bearing elements.

As for clustering methods, the main drawback of statistical methods is their lack of efficiency for fault explanation and root cause determination.

Association analysis: also known as collaborative filtering, it consists of discovering reliable relationships or dependencies between data points in a set. The set can represent temporal or sequential events. Formally, it tends to discover rules of the following form:

$$X \rightarrow Y \text{ or } \forall x \in X, \forall y \in Y, x \rightarrow y \quad (2-1)$$

In other terms, this indicated that when X occurs, Y occurs with a certain likelihood (confidence). With distance measures which compute similarity between items, such as Apriori-based methods (Agrawal and Srikant, 1994), and frequent-item based algorithms such as Toivonen algorithms (Anand and Jeffrey, 2011), the filtering approach is broadly used in commercial tools for item suggestion. For example, rules such as “Those who buy a computer also buy a bag” or “Those who listen to Mozart also listen to Beethoven” allow Amazon and Google to somewhat confidently suggest additional items to buy or videos to watch. Implementations in the railway domain include infrequent fault analysis in temporal data (Sammouri et al., 2013).

Signal Processing Methods: they are a type of statistical method that seeks to transform the initial signal into another domain more suitable for abnormality detection. Many signal processing techniques exist in the literature. Main methods are Fourier and wavelet transform methods.

- **Fourier Transform (FT):** it transforms the input signal in the time domain into its frequency domain, highlighting faulty frequency amplitudes that cannot necessarily be seen in the original signal. Latest applications are fault symptoms feature extraction using fast Fourier transform (FFT). Examples of applications include work carried out by S. Lu *et al.*, (2014), Xu and Zhou, (2016), and Trachi *et al.*, (2016).

Fourier transform-based signal processing methods are efficient for some stationary, periodic, and sinusoidal signals. However, when the signal is non-stationary, Fourier transform often shows deficiencies. Another problem

arising with the Fourier transform is the loss of time information, which has led to the proposition of multiresolution analysis, also known as wavelets.

- **Wavelets:** the basic idea of wavelets analysis is to provide a time resolution of the frequency domain signal obtained in the Fourier transform case. To that end, a trade-off between time information and the frequency resolution is made by convolving the original signal with another signal carrying no frequency and containing at least one vanishing moment called a mother wavelet. Many mother wavelets exist in the literature. Examples in the engineering fault diagnosis include Morlet wavelets, Gaussian wavelets, Mexican hat, and Daubechies wavelets (Novikov, 2006). Toliyat *et al.*, (2003) used wavelet packets for railway defect diagnosis. In a similar area, Jia and Dhanasekar (2007) applied wavelet transform for railway flat tyre detection. Recent applications tend to include hybrid approaches. For example, for gearbox analysis, continuous wavelets transform (CWT) combined with neuro-fuzzy analysis can be used (Ahuja, Ramteke and Parey, 2020).

The main drawbacks of wavelets analysis are their complexity when transforming multivariate signals, making the analysis and the visualisation of the time-frequency representation difficult. Moreover, choosing the appropriate wavelet is not an easy task. In real application cases, traditional mother wavelets are inefficient and may require contextual wavelets design (Novikov, 2006).

2.4.1.2.3 Inference methods

A way of diagnosing fault is to combine the statistical distribution of the underlying processes with inferences methods. Probability theory contains dominant notions and concepts to achieve these objectives. Existing models widely used are Bayesian-based methods, fuzzy logic, and hidden Markov models (HMM).

Bayesian methods: Bayesian approaches have been widely used for fault detection as they allow prior distribution settings and posterior estimations. Moreover, important characteristics of Bayesian methods are their ability to handle uncertainties and complex nonlinear relationships. Models include naïve

Bayes, Bayesian networks, and Bayesian beliefs (Bishop, 2013). Sankararaman and Mahadevan (2013) implemented a Bayesian inference approach for damage diagnosis. Verbert, Babuška and De Schutter (2017) provided a comparative study between Bayesian and Dempster–Shafer reasoning approaches. They found that the choice of the model might depend on the available knowledge of the system. The most up-to-date research in the railway domain using time series data currently available is the work undertaken by Wang, Ni and Wang (2020) where a Bayesian forecasting model is used for online wheels fault detection.

One of the main weaknesses of Bayesian approaches is that misleading prior assumptions may lead to model inefficiency. Moreover, in long time series data, the inference time complexity is exponential.

Fuzzy logic methods: often seen as an extension of Boolean algebra, fuzzy logic methods aim to provide analytical methods to deal with imprecise or subjective information. A typical example is the weather condition, where two different observers may not have the same feeling of warmth given the same temperature. Related to the possibility theory instead of probability, fuzzy logic key concepts include fuzzy sets, membership functions, and production rules (Zadeh, 1965). It is often jointly used with neural networks in general for railway equipment fault detection. Examples of applications include railway junction distributed fault diagnosis (Roberts et al., 2002), automotive diesel engine fault diagnosis (Simani, 2013), and track circuit fault diagnosis (Chen, Roberts and Weston, 2008).

Hidden Markov Models: built from Markov chains and processes, hidden Markov models (HMM) methods assume that the modelled process embeds non-directly-measurable states. HMM design can be defined in five main steps (Rabiner, 1989):

- the number of the hidden state of the model, which is often contextual to the application domain.
- the number of observations in each state, which is related to the number of output observation symbols.

- the initial probability distribution.
- the state transition probability distribution.
- the observation symbols probability distribution.

HMM have been used to diagnose faults in physical assets such as rotating machinery (Xin, Hamzaoui and Antoni, 2018; Dong et al., 2006; and Miao and Makis, 2007). Recent work encompasses Xu, Liu and Zhao, (2018) and Darong et al., (2018) who respectively diagnose turnout faults in a motor driving system.

Like Bayesian-based models, one of the drawbacks of HMM models is prior parameter and distribution settings. Additionally, model training is often computationally expensive for sequential data.

2.4.1.3 Summary

Quantitative methods are built from numeric measurement of system behaviours by applying computational techniques such as statistics, physics, and mathematics throughout learning processes. In the case of fault diagnosis methods, quantitative methods comprise model-driven and data-driven methods. Model-driven methods build the expected system response and use it to compare with real observed system output to seek for deviations. These models are accurate when the appropriate model is built. However, it is often difficult to design and implement suitable models in complex systems. In addition, these models are often too specific and therefore not scalable.

Data-driven methods aim to discover hidden relationships in historical data. Classes of methods include machine learning, statistical, and inference-based methods. These methods are scalable and flexible. Nevertheless, main drawbacks comprise their computational learning complexity and their inefficiency with data quality issues.

The data-driven approaches are more suitable in the present study. Therefore, the research undertaken explores analytics in this area.

In the next section, qualitative methods are described.

2.4.2 Qualitative Methods

Qualitative models are often implemented when there is high availability of the domain knowledge of the studied system. They are mostly composed of expert system-based methods and graph-based methods, as shown in Figure 2-8.

Graph based methods include Fault Tree methods and signed digraph (SDG) methods.

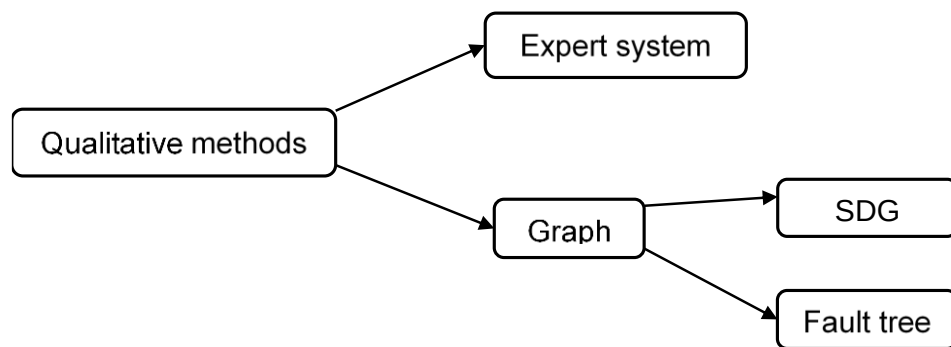


Figure 2-8 Qualitative methods.

2.4.2.1 Expert systems

Also known as knowledge-based systems, they have been widely used in engineering for fault identification and location. An expert system consists of a fact base, which contains declarative knowledge on the known facts; a rule base, which is operating knowledge allowing deductions; and an inference engine, which forms the reasoning program. The fact base and the rule base form the knowledge base. Figure 2-9 shows an example of an expert system structure. Through the expert system interface, the knowledge base is populated. From the user interface, operating knowledge can be used jointly with the inference engine to diagnose a fault.

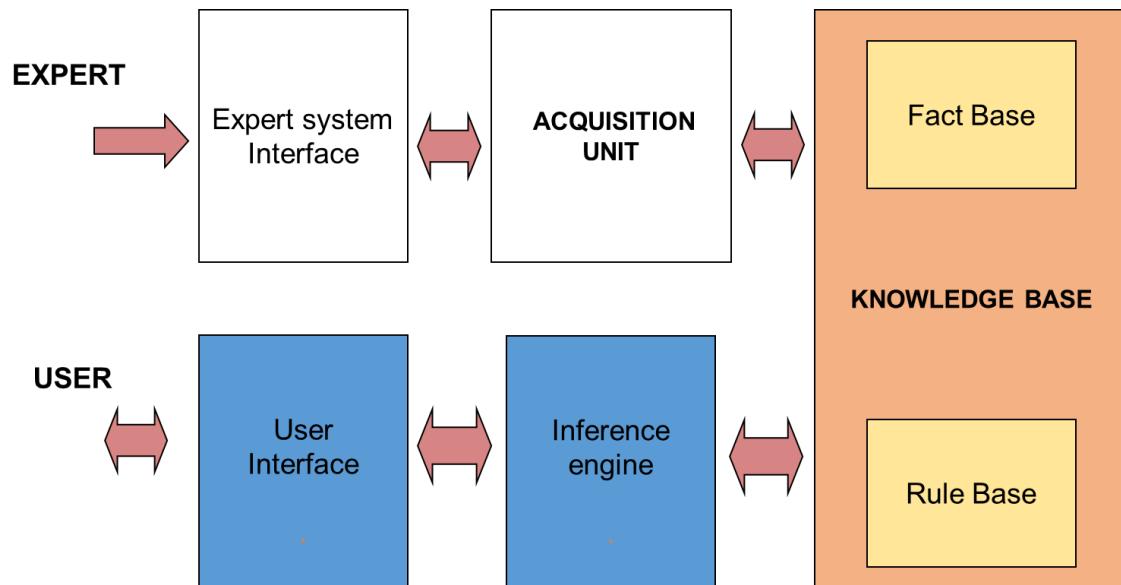


Figure 2-9 Example of the structure of an expert system.

In terms of applications, Xie and Li (2002) proposed a reasoning-based system for traction substation fault diagnosis. More recently, Zhi-Ling *et al.*, (2012) applied an expert system for gearbox fault diagnosis. Xu *et al.*, (2017) investigated an expert system model for marine diesel engine fault diagnosis. Aronson *et al.*, (2020) diagnosed steam turbine components by applying a jointly expert system and probability.

The main weaknesses of an expert system are that it requires a high level of expertise for model efficiency, and the knowledge base often contains conflicting rules. Moreover, it suffers from scalability. The knowledge base is often so narrow in a specific domain that it is difficult to extend its application to other domains.

2.4.2.2 Graph-based methods

Graph-based methods aim to represent a relationship amongst variables. Widely-used methods for fault diagnosis are Signed Digraph (SDG) and Fault Tree methods.

Signed Digraph methods: SDG-based methods represent the studied system through a graph where nodes represent variables and the edges represent their interactions. In the presence of faults, the structure of the graph (weights of the edges) may change (Venkatasubramanian, Rengaswamy and Kavuri, 2003). Xu

et al., (2016) applied SDG for diagnosing traction faults. The latest work includes cause-effect relationship modelling for analysing railway urban rail plug doors (Peng *et al.*, 2019). In this work, a Petri network is jointly combined with SDG for efficient fault diagnosis. The main advantages of SDG methods are their ability to handle uncertainties and incomplete information of a system. However, like expert systems, they suffer from scalability issues.

Fault tree methods: Fault tree methods can be seen as a special digraph where nodes can use Boolean operators such as AND, OR, and XOR (Venkatasubramanian, Rengaswamy and Kavuri, 2003). From an operating knowledge, a propagation search can be undertaken to look for the root causes. Time-based fault tree application includes the work undertaken by Y. Lu *et al.*, (2014), where Time Fault Tree is used to analyse railway incidents.

The weaknesses of fault tree methods are their building complexity and their scalability.

In general, qualitative methods are not applicable for time series analysis. The reason is that they are more focused on knowledge modelling than the temporal measurements and behaviour of the studied systems.

2.5 Conclusion and Gap in Knowledge

This chapter focused on a broad literature review in the dominant areas that this research evolves. Firstly, it described basic maintenance strategies and highlighted their advantages and weaknesses. Diagnosing faults is key for cost reduction, productivity, and the improvement of competitiveness, as well as passenger journey improvement. However, it comes with data collection, transmission, storage, processing, and visualisation challenges. Secondly, big data concepts and challenges were presented, with an emphasis on data storage, management, and analysis tools. Various tools exist in the literature, keep evolving and selecting appropriate tools depends on the type of collected data (structured or not) and the selected type of analysis (streaming or offline,

classification, clustering, or regression). There is a poor implementation of specific time series tools in general. Standard tools are often used for the implementation and often lead to the loss of the temporal structure of the series. Lastly, this chapter reviewed and compared fault diagnosis methods with application to the railway domain in particular and the transportation domain in general. Fault diagnosis methods were classified into qualitative and quantitative methods. Quantitative methods can also be divided into data-driven and model-driven methods. Table 2-2 summarises the advantages and weaknesses of the described methods. Qualitative methods are interesting when an in-depth knowledge of the studied system is available, whereas quantitative methods are useful when partial information of systems exist or important historical data are available.

Table 2-2 Advantages and drawbacks of fault diagnosis methods.

	Qualitative methods	Quantitative methods	
		Model-driven	Data-driven
Advantages	More reliable Expert knowledge Do not necessarily require operating data Can handle uncertainties	Do not require a significant amount of operating data Can handle unknown fault Can handle uncertainties	Scalable and flexible Require less contextual knowledge Numerous methods are available
Drawbacks	Difficult to implement in complex systems Cannot handle unknown new fault Less scalable to other domains	Less flexible Required prior knowledge of the system Difficult to implement in complex systems	Data quality and quantity dependant Expensive computational cost

Focusing on the quantitative methods, various techniques have been built and used for condition monitoring throughout the extraction of relevant features. However, these models are not readily available for any fault detection diagnosis. These methods often generate a high rate of false alarms leading to fault misdetection, creating additional maintenance cost and activities. Furthermore, assumptions taken into account for building the models do not reflect real-world

case scenarios. Hence, these led to the need for the development of new diagnosis methods and methodologies to solve the aforementioned problems. This present work focuses on developing time series data-based methods with application to real-world train assets data, namely train engines, doors, and wheels. Data-based methods are chosen due to data availability and also because of the limited knowledge of the studied system.

The next chapter describes the applicative context and its underlying data monitoring challenges.

3 PREDICTIVE MAINTENANCE FOR RAILWAY ASSETS

3.1 Introduction

Maintenance has always been an essential task for most industries, particularly within the manufacturing domain. It is an evolving area since new technologies offer better equipment condition monitoring and therefore allow the possibility and accessibility of a quick diagnosis and prediction of an incipient fault arising in the systems.

In railway industries, maintenance is fundamental in order to face the growth of the number of customers. For example, according to Transport Network magazine (Network, 2017), the number of rail passengers is increasing by three per cent every year and will keep growing in upcoming years at the same rate, at least. From an ecological point of view, companies have to meet evolving compliances, such as new carbon emission regulations, or else pay high tax rates. Moreover, railway companies need to consider competition, customer satisfaction, costs, and efficiency in order to prosper. Therefore, it is necessary to keep the trains' assets highly available, avoid unnecessary maintenance, and prevent failure by employing earlier maintenance activities with a "good" maintenance strategy (e.g., predictive maintenance).

This chapter describes the research application area with a focus on the studied assets' operations, namely engines, doors, and wheel slips of Great Western Railway (GWR) and West Midlands Trains (WMT). It also describes uncertainty and data-related issues. Assets such as diesel engines, doors, and wheels are not new systems; they are well known and have been subject to numerous studies. However, it is still important to continue researching and analysing these complex and nonlinear systems in order to improve the assets and ensure optimal behaviour. These assets are evolving with the operating environment, including internal combustion, driver behaviours, vehicle load and inertia, track surface, and environmental conditions.

3.2 Condition Monitoring in the Railway Industry

Railway infrastructure assets are perpetually exposed to usage and the environments. As a result, it is important to continuously evaluate the states of the assets to ensure that they are operating correctly and that they are safe, especially during peak periods. Principal investigations took place in the following areas:

- Track analysis: it comprises geometry deformation (Shenton, 1985) (Indraratna and Nimbalkar, 2013); defects (Fink, Zio and Weidmann, 2014); circuits (Lin-Hai, Jian-Ping and Yi-Kui, 2012); general inspection (Kobayashi, Naganuma and Tsunashima, 2013); gauging; and rail switches (Samé et al., 2011).
- External power sources: pantographs (Vinnakota, 1996), and electric line deformation (Wang et al., 2019).
- Vegetation monitoring: buildings, trees, stones, and external disturbances, such as animals (Nyberg et al., 2014).

The rolling stock monitored assets are mainly vehicle systems and subsystems. Recent analysis focuses on systems such as doors, element bearings (Pennacchi et al., 2011), gearboxes (Hu et al., 2017), wheels and axles (Liang et al., 2013), engines, and brakes.

Further details of the applied techniques are presented in the literature review chapter in section 2.4. A focus is made in this study on the condition monitoring of three sub-systems: the engines, the doors, and the wheels.

3.3 Doors Basic Operations

Train door systems have evolved over the decades. Early door systems, also known as slam doors, were fully mechanical systems which required substantial human force to open and close them. In the first slam-door systems, passengers were able to open or close the doors whilst the train operated. The main drawbacks of this type of door are injuries caused by reaction opening forces, and accidents due to passengers falling out of the train. Another significant

negative aspect of slam doors was that the opening system was often not suitable for children and disabled passengers. To reduce the number of accidents and passengers falling from the train, a centralised locking system was later introduced, enabling global lock and release of the doors.

Recent door systems use powered doors, commonly known as automatic doors which are composed of one or two leaves moving in the opposite direction; a power source; transmission and locking systems; a control system; and traction system. These types of systems create effortless opening, as well as full control of the doors. An example of a two doors system is shown in Figure 3-1. Drivers can lock or release the doors without passenger involvement and can also release specific doors, which is crucial when the train is longer than the platform. The drawback of powered door systems in comparison to slam-door systems is that they are complex to design, implement, install, monitor, and maintain, causing additional costs and time spent on them.



Figure 3-1 Example of doors on GWR train.

3.3.1 Classical Motions of Automatic Doors

Three main basic door designs can be distinguished: sliding, sliding plug, and swing. Each design can be explained through the door panel movements.

3.3.1.1 Simple Sliding Motion

The simple sliding door motion is commonly used in Diesel Multiple Units (DMU) trains. It is an unidimensional motion where the door slides parallel to the body of the train, as shown in Figure 3-2. Pneumatic linear actuators are often used for this system. The main issues encountered with sliding doors are the accumulation of debris, dust, and moisture in the sliding area causing component corrosion or jamming, and ultimately affecting the doors' normal operations. Additionally, the doors can also fail due to actuator or power source failures. Door data analysed in this project are from this type of system where the battery current and the opening and closing motion are recorded over time.



Figure 3-2 Sliding Door Motion.

3.3.1.2 Sliding Plug

This type of door system presents two distinctive dimensional motions during the opening and the closing phases, as shown in Figure 3-3. In the opening phase, the door first moves perpendicularly to the carriage, creating a small gap between the door and the train body. The second movement of the door is similar to the simple sliding motion with a parallel move to the train body (Railsystem, 2015). This type of design helps to reduce several issues encountered with the simple sliding door systems such as debris, wear, and moisture. The downsides of the

sliding plug system include their complex design as well as them being difficult to install and maintain.



Figure 3-3 Sliding Plug Door Motion.

3.3.1.3 Swing

Like sliding plug door systems, swing door systems also have a motion in two dimensions, however, the motion is different. Instead of an initial perpendicular move of the door, a rotational movement is made, as illustrated in Figure 3-4. The benefit of this type of design, in comparison to the sliding plug, is the design simplicity brought by the rotational mechanism.

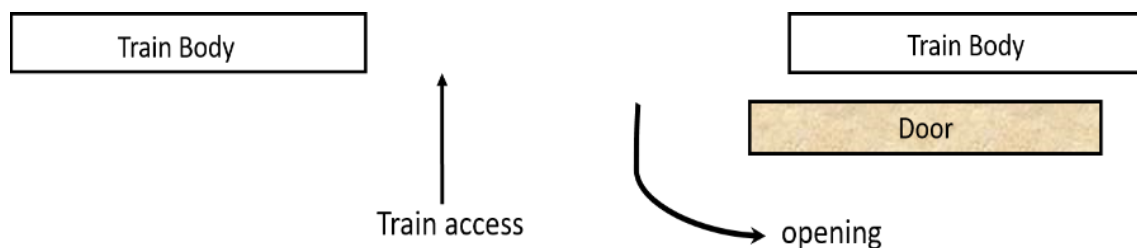


Figure 3-4 Swing Plug Door Motion.

3.3.2 Doors Failures

Several types of door failures exist and are mainly due to the degradation of its main parts such as the door panel, power sources, and mechanical systems like actuators or transmission. The most common failures include:

- Slow opening/closing: this is when doors take more time than expected (often around four seconds) to open or close, which can lead to traffic delays.
- Fast opening/closing: often linked to either an actuator, sensor, or power source fault, fast opening and closing can cause damage to internal components (due to the slamming) and can cause injury to passengers.
- Faulty locking system: this is related to the ability to open the door whilst the train operates, which is unsafe. This is critical as passengers can fall out of the train carriage during movement.
- Passenger general-purpose input/output (GPIO) opening/closing buttons fault: the opening and closing buttons or handles do not respond to passengers' actions. This can create significant delays due to passengers not being able to board the train on time.
- Vibrating doors: this failure can be observed at the door steady-states. This can create a disturbing noise for passengers, as well as damage door panels and other train components.
- Change in the door speed and overshoots: it is preferable that the doors move at a constant speed. Faulty sensors or power source can lead to door speed fluctuations. Fast speed can hurt passengers and damage door panels and low door speed can create train delays.
- Displacement or misalignment: this type of fault affects the opening and closing of the door, for example the door might only half-open or might not fully close, leaving a gap between the door and the lock. This can also cause train delays.

3.3.3 Data collected

The data collected are mainly the motions of doors at train stations during train operations. It is approximately one tera octets data of train door operations. The data attributes are as follows:

- Opening GPIO: it is a binary signal representing passenger actions on the opening buttons.

- Closing GPIO: it is a binary signal representing passenger actions on the closing button.
- Current: it represents the current provided by the power sources to either open or close the door.
- Speed: it represents the door motion over time during the opening and the closing profile.

The datasets are available in Cranfield repository (Namoano et al., 2020b).

3.4 Basic Operations of Diesel Engines

3.4.1 Diesel Engines Operation

Diesel engines belong to the internal combustion family. Unlike steam engines, diesel engine combustion occurs within cylinders where power is produced. This method wastes less generated energy because the heat does not have to be moved from the production location to the cylinders. Usually, the engines operate periodically following four main stages, also called strokes, in which the crankshaft rotates twice, as shown in Figure 3-5:

- **Intake:** in this stage, the air throughout the inlet valve is drawn into the cylinder whilst the piston moves down.
- **Compression:** after the piston reaches its maximal position, the inlet valves close and the piston moves back up, compressing and heating up the air mixture. Fuel is then injected into the heated air through a specific valve and is ignited.
- **Power:** the mixture burns and pushes the piston down, driving the crankshaft.
- **Exhaust:** the outlet valve opens, and the return of the piston pushes the exhaust gases out.

Notably, not all diesel engines have four strokes; some have only two strokes with a slightly different operation.

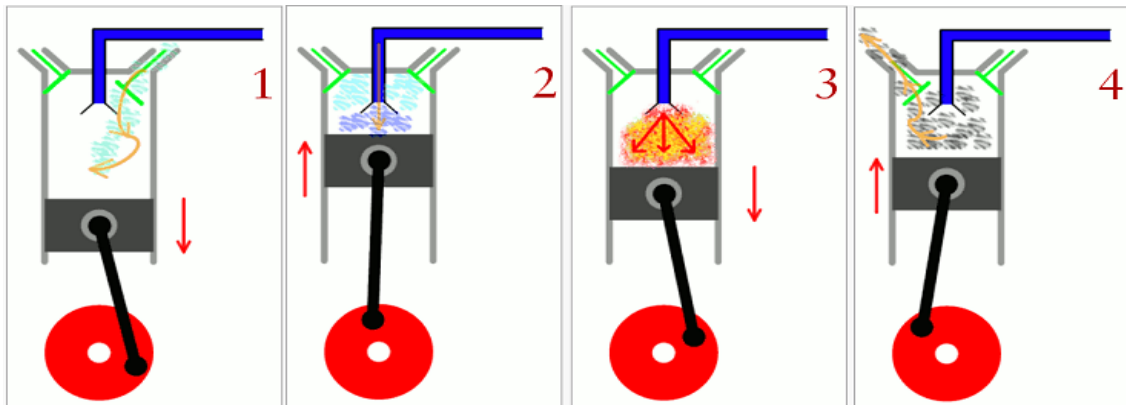


Figure 3-5 Diesel engine combustion stages (Woodfort, 2014).

3.4.2 Cooling System

The cooling system is an essential part of combustion engines. As the combustion of the mixture of air and fuel produces heat, some of this energy is expelled with the exhaust gases as well as the lubrication system, but not all. A cooling system is required in order to remove the remaining heat inside the cylinders in order to obtain the optimal compression ratio and power. A dysfunctional cooling system can lead to severe damage. For example, a higher temperature in the engine can cause the lubrication fuel to burn, which can, in turn, lead to significant damage to the cylinders and can also cause the engine to explode and the surrounding components to deteriorate. Alternatively, very low engine temperature can affect the performance of the combustion by causing low power events, as the combustion level is not optimal to produce the required power.

There are two types of cooling systems: air cooling systems and liquid cooling systems. Depending on purpose and design, an engine may have either or both types.

3.4.2.1 Air cooling systems

In these systems, cold air is injected through air inlet valves and flows next to the engine, where a fan dispatches the air, as shown in Figure 3-6. The energy

released by the engine heats this air, which then escapes through the air outlet. One of the advantages of using an air-cooling system is that there is no risk of freezing, like liquid cooling systems.

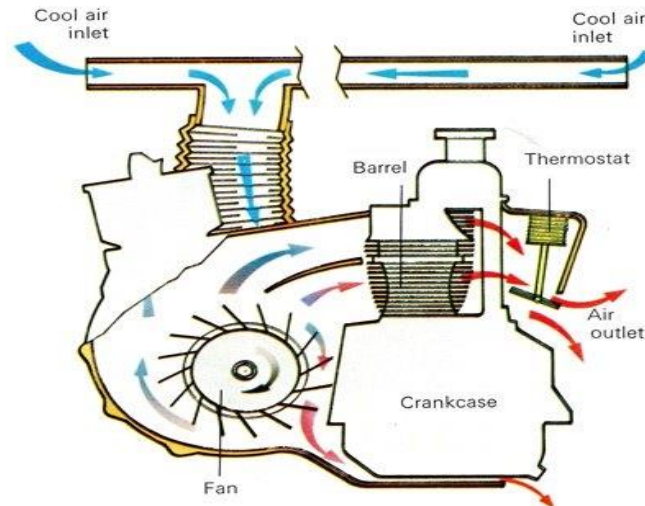


Figure 3-6 Air cooling system (Hajam, 2014).

3.4.2.2 Liquid Cooling Systems

In liquid cooling systems (water oil), water is the most used coolant as it reduces mechanical noise, is easily available and thus cheaper, and it regulates the temperature around the engines. A schematic representation of a cooling water system is shown in Figure 3-7. A low temperature cooling liquid runs through the pipes and gets heated by the engine energy released. The hot liquid goes through radiators that cool it down.

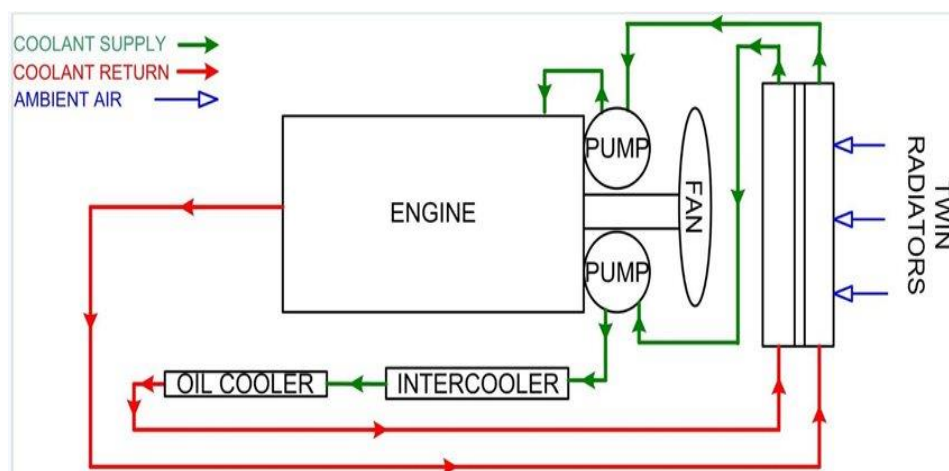


Figure 3-7 Example of engine cooling system (Hajam, 2014).

3.4.3 GWR and WMT Trains

GWR and WMT trains are a fleet of diesel multiple units (DMUs). DMUs engines were built in 1990's in York for British Rail. These train locomotives cannot be separated because the engines are integrated into the carriages. Figure 3-8 shows an example of DMUs. Their main characteristics are their self-propelled vehicles, the distribution of the traction load, and their medium acceleration.

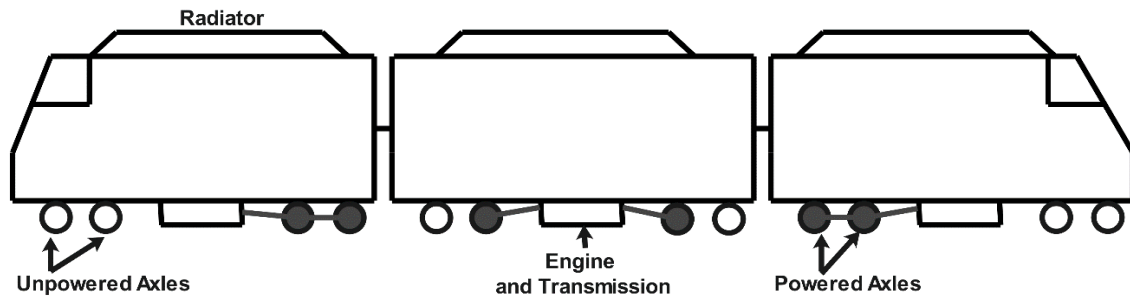


Figure 3-8 Example of a DMU train schematic design.

3.4.3.1 Engine Failure

Engine systems are complex, and even if they are well known, owing to their complexity, there are many reasons why engine systems fail. Explanations may be that the failure types depend on the design and architecture of the engine as well as their operating conditions and environment. However, common failures amongst all engine systems are fuel and air injection systems, the piston or the cylinder, the cooling system, and the bearing. These failures prevent the driver from having the necessary power to meet the driver acceleration demands.

One of the significant observed faults in DMUs engines is a low power event. This is where the engines fail to produce enough power at the same time to move the train. An example of this is when at least one engine may lose its power during acceleration, leading to a drop in the engine frequency.

One of the significant observed faults in DMU engines is the inability to generate the necessary amount of power to move the train. For example, during the acceleration period, at least one engine may lose its power, leading to a drop in the engine frequency. This event is called a low power event. Figure 3-9

represents an example of a low power event occurring in the three DMU train engines that were studied. The patterns include the engine in idle mode, in which the frequency is around 800 rpm as well as acceleration mode, in which the frequencies increase from 800 rpm to a maximum speed of around 1800 rpm. This is followed by a gearshift which is represented in the graph by a slight decrease in the frequency to approximately 1340 rpm, before another acceleration occurs. Then, the three engines decelerate into idle mode. As shown in the example, Engine 1 remains at an abnormal frequency level during the first acceleration. A reason for this unexpected frequency level could be due to a physical deterioration of a component, which will affect the efficiency of the combustion or the transmission. Another reason could be leaves getting stuck in the tank or having dirty valves that clog the fuel or air flow, leading to a failure. Such faults cause train delays and therefore affect the entire railway transport system. Moreover, low power events occurring in specific conditions like extreme gradient profiles (for example the train going up hills or a tilted stop) are severe and may lead to accidents as the lack of power can cause the train to fall down or drive backwards.

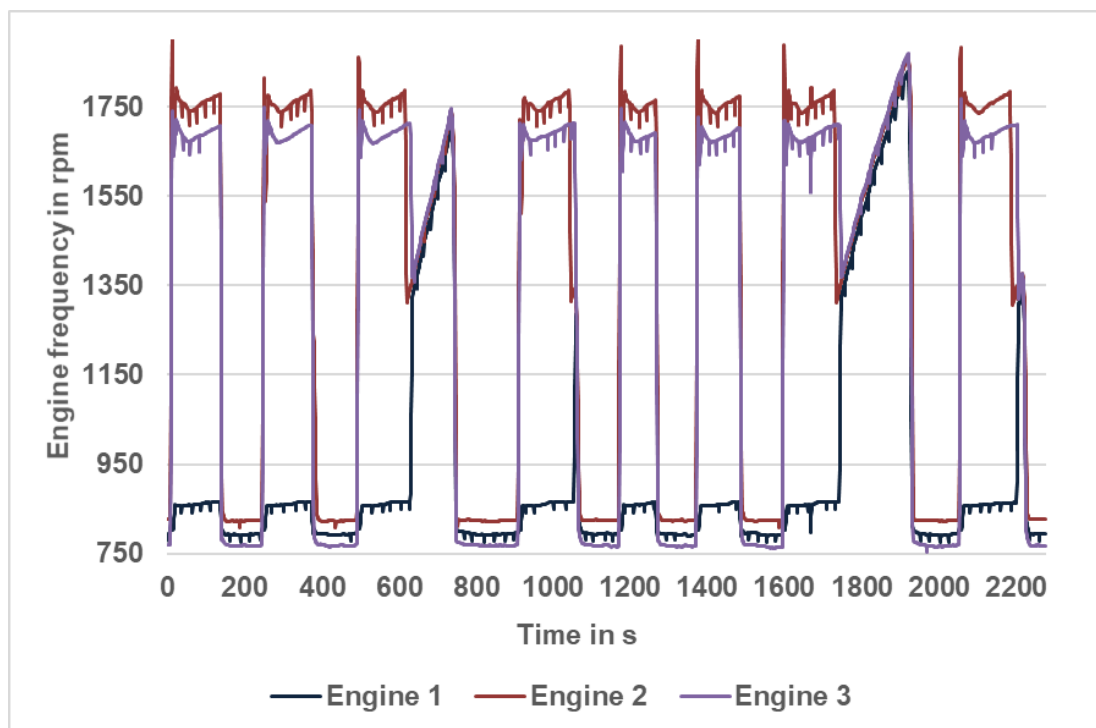


Figure 3-9 Low power event of a DMU train with three engines.

3.4.3.2 Data collected

The trains' usual operations were recorded for two years, totalling around two tera octets of data. The data are from the DMUs engines' aforementioned sub-systems temporal measurement. Each engine shares the following signals:

- ambient temperature
- charge air temperature and pressure
- exhaust temperature
- fuel temperature and pressure
- engine frequency
- in and out oil cooler
- oil pressure control
- radiator fan frequency
- the air in and out temperatures
- cooling water inlet and outlet temperatures
- water inlet and outlet pressures
- the second stage oil

Several raw datasets are available in the Cranfield repository (Namoano et al., 2020c).

3.5 Wheels-rail Contact

3.5.1 Wheel-rail adhesion

Understanding wheel-rail contact is an important step to understand wheel slip behaviours. Figure 3-10 shows an example of a wheel-rail system.



Figure 3-10 Wheel-rail contact with a sanding system (Ricardo, 2019).

During the acceleration periods, there is an increase in the rotating force in the vehicle axles. The level of friction (rolling resistance) between the wheel and the rail is less than the amount of force generated by the increased torque, which often results in over-rotation, also known as wheel slip activity. This wheel slip activity causes damage to both the track and the wheels.

Another type of wheel behaviour that can occur under braking conditions is under-rotation. It is different from the aforementioned wheel slip as the force being applied by the brakes is a resistance force (the brake pad pressing against the wheel), as opposed to the turning force from the torque under traction. If the amount of force applied via the brakes exceeds the resistive force of the wheel on the track (also including other variables such as vehicle weight and train speed), then the wheel will under-rotate. This lock-up causes a flat spot on the wheels. A flat spot is a serious incident for a train wheel as the slapping effect causes significant damage to the wheel as well as the track.

3.5.2 Data Collected

The data collected for the wheel are axles and the sanding measurement. Dimensions include:

- Axles speed,
- GPS speed, coordinates, time,
- Brake pressures applied to the axles,
- Emergency brakes,
- Electro Dynamic Braking,
- Leading vehicle during the journey,
- Onboard wheel slip detection label for each vehicle,
- Sand level and variation.

3.6 Data Issues and Challenges

Although data are gathered from different train vehicles from the same units, they suffer from similar issues. Main issues and challenges are highlighted in the following sections

3.6.1 Poor Knowledge of the System Behaviour

One of the biggest challenges in this study was to interpret and associate a meaning to the patterns observed in the datasets. Some data lacked critical details and information such as the exact types of engines used and the class, or the types of doors that were used (like pneumatic or electric). This brought significant challenges in understanding asset operations throughout the patterns observed in the data.

3.6.2 Lack of Maintenance Records

Maintenance records are crucial for understanding the health condition of an asset. For example, maintenance records provide the diagnosed fault type of equipment during a specific period, the maintenance activities that have been carried out, and can help identify if a change in the data is a result of a maintenance activity or abnormal behaviour. For instance, systems such as engines are very complex and behave accordingly to their environment, thus an abrupt drop in the monitored signal is not always characteristic of a fault. A good example of this is a sudden break or an emergency break. However, the maintenance records help to understand the intermittent fault occurrence, its severity, and its distribution over time. This would mean that a lack of maintenance records may prevent the ability to label the datasets, thus imposing unsupervised learning methods as primary approaches. The problem with this is that the insights extracted from the data may go unexplained, and unsupervised learning approaches can lead to poor accuracy in the fault diagnosis process.

3.6.3 Lack of Input Parameters and Essential Attributes

It is mostly model-based analysis used in the diagnosis of mechanical systems, particularly engines. This is because their behaviours are less stochastic and their physical behaviour can be modelled. Over the past few years, there has been an increase in model-based and data-based hybrid methods being used for engine control and simulation. These methods present a better efficiency when the input and output signals are known or when the captured signals present most of the system's behaviour. In this study, fundamental parameters are missing, such as

the drivers' action; the fuel and airflow; the traction load; the gear speed level; the power generated; the engine torque; or even the vehicle speed. These missing parameters make it more difficult to accurately establish the fault diagnosis, as there is no obvious link between the attributes.

3.6.4 Telemetry Issues and Desynchronisation

Telemetry issues significantly affect the quality of data. The two major issues encountered in telemetry are burst data and considerable gaps, due to transmission errors. In the context of a time series, removing the duplicated data is not easier, as the decision on which data should be kept or removed is not often obvious. Likewise, filling in the missing data through a suitable data imputation technique is a challenging task. Figure 3-11 gives an overview of the gap lengths taken from this study.

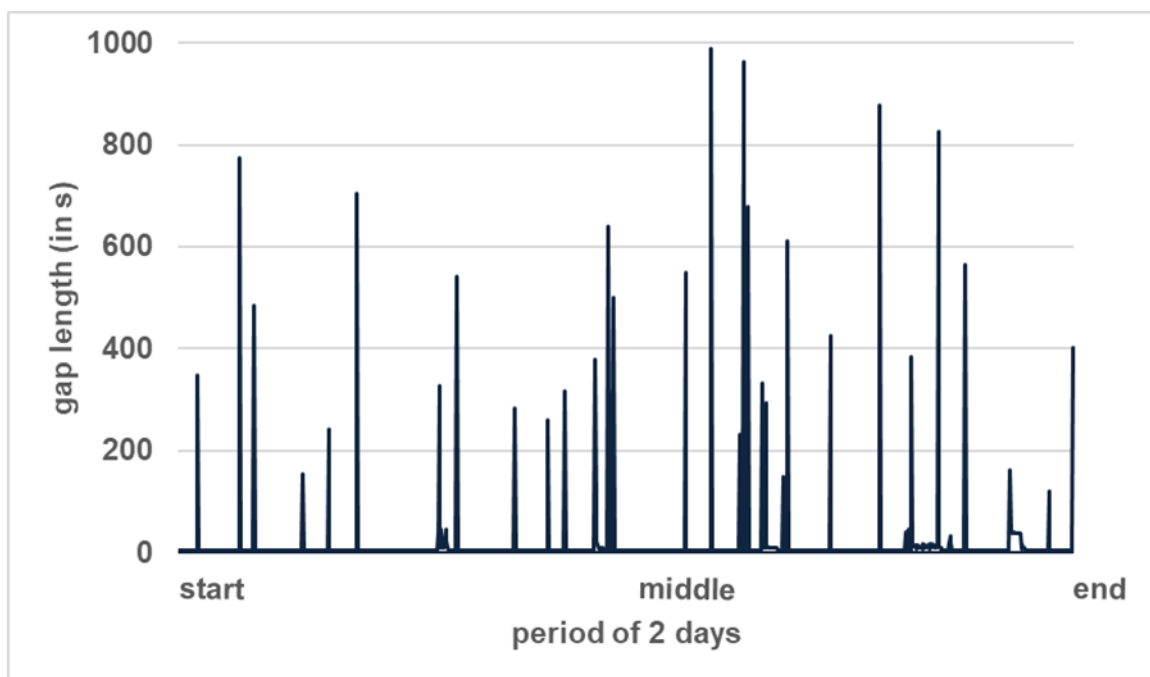


Figure 3-11 Missing data length over two days during running hours.

For example, in a ten minutes interval where no data is observed, the train may have stopped to let off passengers, meanwhile signals such as outlet temperatures could have decreased. Current studies suggest under-sampling techniques to overcome these issues. In this specific context, under-sampling

leads to missing events such as driver acceleration periods, in which the gradient may give valuable information about an arising fault in the system. The methodology adopted here is segmenting the multidimensional dataset into journeys. A journey can represent starting, idling, accelerating, and deceleration.

Another significant issue to resolve in the dataset is time desynchronisation, as demonstrated in Figure 3-12. The graph represents three DMU train engine frequencies measured in revolutions per minute taken with an interval of one second. It can be seen that the signal of Engine 3 is not synchronous with the two other engines. Signal alignment is necessary to resynchronise the data.

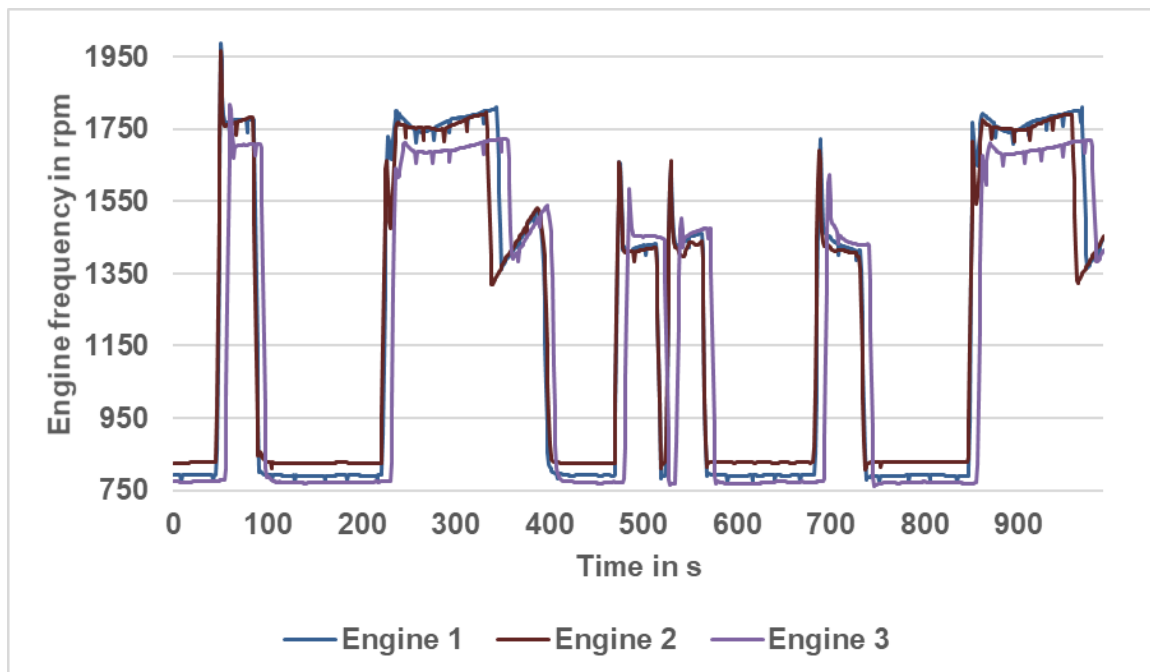


Figure 3-12 Train DMU engines frequency over time.

3.6.5 Issues with Noise and Sensor Readings

Issues with noise and sensor readings are common in real-world datasets. For instance, in Figure 3-12, for each engine frequency plotted, slightly regular dips, which represent noise, can be seen. Its magnitude often depends on the position and the environment of the train (for example stationary or moving, uphill or downhill, accelerating or braking), and the state of the sensors as they deteriorate over time. Therefore, the reliability and accuracy of the data decrease over time.

In the datasets, extreme values can be interpreted as errors and are easy to remove within the control limit. However, sometimes sensors can freeze at a specific value within the normal range of expected values and these are difficult to reduce.

3.6.6 Lack of System Fault in the Datasets

Most of the data were recorded during the train's regular operation. Hence, the vast majority of the analysed datasets do not highlight any apparent system faults. This makes the analysis more difficult as it is necessary to understand the train equipment's normal operating mode as well as its failure mode in order to build an efficient model that is able to recognise incipient faults in the systems and provide an accurate diagnosis.

3.7 Conclusion

This third chapter has presented how this study can be applied to solve real-world issues found in DMU engine diagnosis. Three case studies that investigate DMU engines, doors, and wheels were introduced and discussed. Each system faces different issues that require a solution, such as signal desynchronisation, missing data, noise, a lack of maintenance records, and different fault patterns. In order to tackle these challenges and diagnose potential faults in each case study, it is essential to define an appropriate generic architecture for time series analysis.

The next chapter introduces the most common data fusion architectures and proposes an innovative time series architecture for each of the case studies.

4 BUILDING TIME SERIES ARCHITECTURE

4.1 Introduction

Multi-sensors data fusion have widely evolved leading to the point of a new discipline being developed. For example, early investigations in finding formal processes for disparate information integration have arisen in the defence industry. This is due to the need to assess threats in battlefields by obtaining data from different channels in order to respond with the correct actions. As a result, significant progress in proposed architectures and application domains has been made. In the past, application domains were mainly in traffic management, medicine, and metrology fields. With the improvement of sensor technology making low-cost but accurate sensors more accessible, most of today's devices embed sensors allowing the enhancement of their conditions and measurements. These sensors, also known as smart materials, structures, or data providers, are able to record their surrounding environment, which is their main goal, whilst also monitoring their own internal conditions. The sensors can be classified into two main groups: active and passive. Active sensors require an external power supply to operate, whereas passive sensors produce a continuous output without any external triggers. Both types may contain a single or distributed. It is relevant to take into account the sensor type for building efficient architectures. Whilst progress has been made in building architectures, some special cases, such as time series, still require investigations. This chapter reviews the main available architectures and applications. An innovative architecture for time series analysis is also introduced and discussed.

4.2 Review of Data Fusion Architectures

Data fusion architectures have been developed to integrate sensor capabilities and their environment, as well as the available knowledge of the studied system and decision-making processes. It is important to understand how the single sensor model works in order to understand the upcoming proposed models that will be discussed later in this thesis. This is because it contains most of the essential steps for all of the remaining architectures.

4.2.1 Single Sensor Fusion Architecture

A straightforward single sensor fusion architecture is highlighted in Figure 4-1 below.

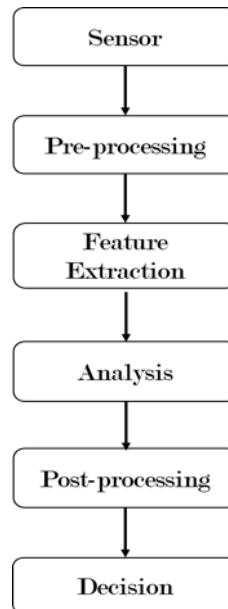


Figure 4-1 Single sensor fusion architecture.

The six main steps can be summarised as follows:

Sensor: it provides measurements of the monitored parameters characterising the studied environment. Examples include temperature sensors, which continuously measure the changes in temperature; proximity sensors, which can detect nearby objects; and gearbox accelerometers, which monitor gear conditions.

Pre-processing: this is also known as data preparation or data tidying. The main goal of this step is to reduce the uncertainties in the sensor measurements and to obtain coherent datasets. It encompasses signal selection; data cleaning, which seeks for noise as well as outlier reduction; and data imputation, which aims to manage missing values. For accurate data preparation, context knowledge of the system is often required.

Feature extraction: the feature extraction step generally aims to combine signals in order to obtain representative signals that better characterise analysis purposes (for instance, diagnosis and prognosis). A good example of this is when the analysis is being carried out in order to detect a fault, the features may be a combination of signals that best fit the normal behaviour and the fault deviations of the system. When analysing data for high dimensional observations, dimension reduction methods are important for feature extraction. In the case of measuring single signals, extracted features are often statistical properties of the signal such as the mean, the root mean square, kurtosis, and skewness. This step is often related to the context, and prior knowledge of the system may help to extract the appropriate features.

Analysis: the aim of the undertaken analysis has influence on previous explained steps. It is common to investigate both fault diagnosis and prognosis. A diagnosis analysis aims to detect, find the root causes, and classify or estimate the severity of equipment states. In contrast, a prognosis analysis is to help predict and estimate possible future occurrences.

Post-processing: this consists of representing the analysis results in a simple manner, enabling the decision-makers to take action. It often involves rule-based systems explaining the results of the analysis and data visualisation techniques.

Decision: this is the final step of the process. Humans or implemented automatic systems make the decisions. Simple rules and data visualisation are more suitable for human-made decisions, and a rule-based post-processing stage enables computers to trigger actions using their decision programs. In general, the lack of fault tolerance is a significant drawback in this type of architecture and any errors made during a step may lead to the process failing. For this reason, it is important to consider using multiple sensor architectures whenever feasible.

4.2.2 Multiple Sensors Fusion Architecture

Medical studies show that humans make more accurate decisions when more than one sense is involved. For example, a person's assessment of what they

are eating is more accurate when sight and taste are both involved (Hall and James, 2001). The same advantage occurs in sensor data fusion. Following (Esteban et al., 2005) highlights the following advantages:

- more coverage and better resolution of the system
- ability to obtain independent features, which provides a complete representation of the system
- more accurate results
- uncertainty reduction
- improvement of the robustness and the reliability of the analysis

Redundant signals are useful to improve fault tolerance level. For example, by applying robust methods, noise can be filtered out in redundant signals. Classical architectures include centralised architecture that covers feature, pattern, and multi-level architectures; joint directors of laboratories (JDL); Boyd; waterfall; and Omnibus models.

4.2.2.1 Centralised fusion architecture

Centralised fusion architecture enables a joint fusion at some step in the architecture process. In Figure 4-2 and Figure 4-3, the joint fusion is made at the feature level and pattern level respectively. This allows redundancy elimination and a combination of the different data sources to obtain new variables, enhancing the goal of the studies. The main advantage of this model is that meaningless data are not carried into other levels, which reduces potential additional costs (as this saves on time and server costs). In pattern fusion level, features are extracted independently and then analysed together. This type of architecture can be useful when there is less knowledge about the system. Black box algorithms, such as neural networks or evolutionary algorithms, can be applied in such situations (Dong et al., 2009).

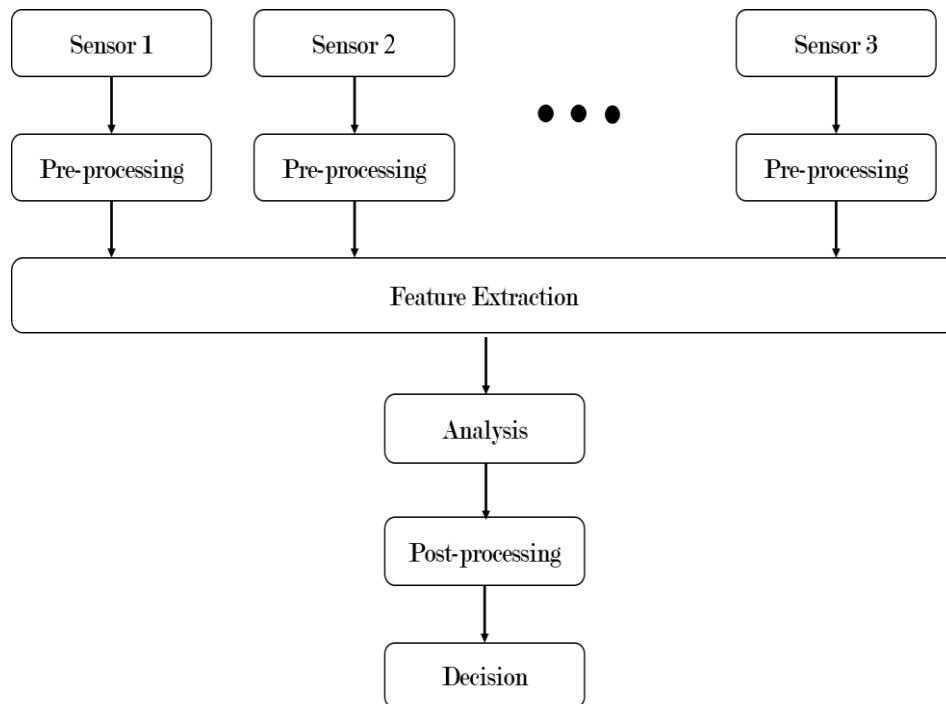


Figure 4-2 Centralised architecture: Feature level fusion.

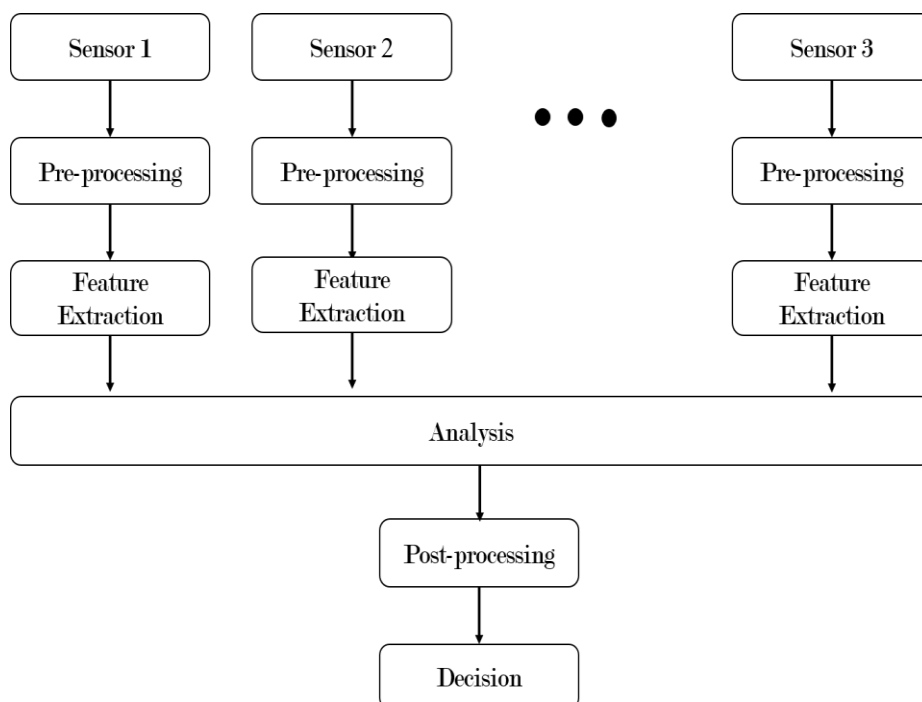


Figure 4-3 Centralised fusion: Pattern level or analysis level architecture.

A more advanced centralised fusion architecture is a multi-level fusion model, in which different joint fusion levels are made. Figure 4-4 provides an example of such architecture. For instance, it facilitates different levels of feature extractions and different steps of analysis, as well as joint different decision levels (Klein, 1999). However, these models are difficult to implement as they are often complex and can lead to duplicate feature extractions.

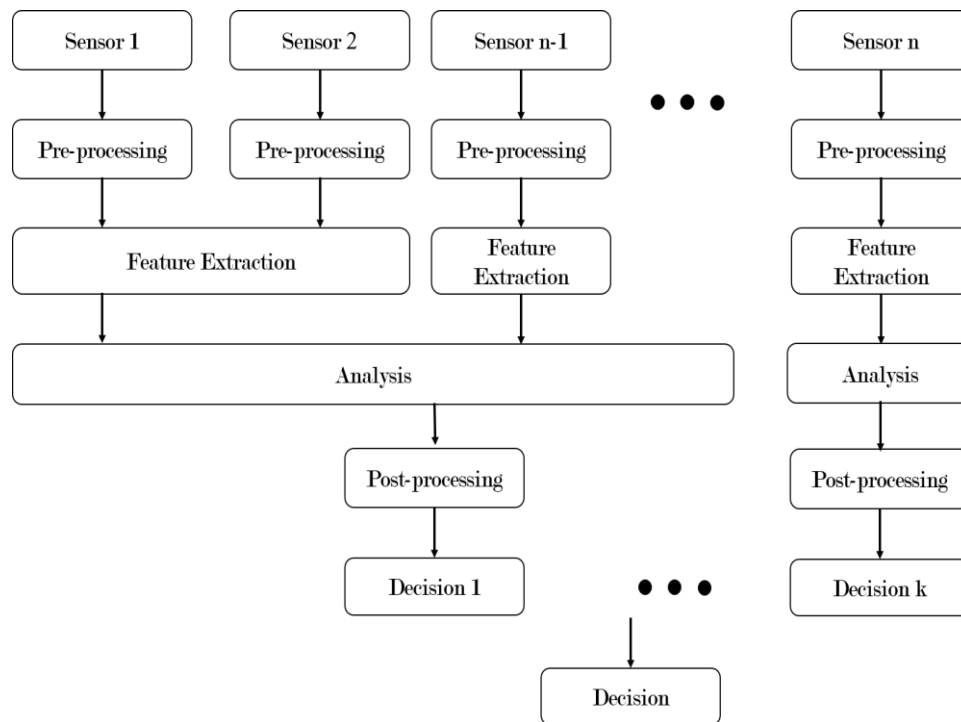


Figure 4-4 Centralised fusion: multiple fusion levels.

In general, centralised architectures are efficient for one-way decision-making processes. A typical real application of this is the waterfall model (Hall and James, 2001) which is a linear sequential task steps architecture. However, in more complex process cases, it might be more beneficial if the taken decisions were also influencing the architecture. Due to this limitation regarding one-way decision-making processes, other architectures integrating feedback loops were developed.

4.2.2.2 JDL model

Initially developed for military applications for coherent information exchange and fusion (Hall and Llinas, 1997), JDL has evolved and become one of the most used data fusion architectures. JDL consists of five main levels that interact with each other, as shown in Figure 4-5. The levels can be described as follows:

- Level 0 is the pre-processing stage described previously (section 4.2.1).
- Level 1 is the object assessment. The aim of this stage is to fuse multiple sensors in order to estimate object identity and position.
- Level 2 is the situation assessment stage which provides a general overview of the situation by taking into account information obtained in the previous levels.
- Level 3 involves an impact assessment which considers the overview made in level 2 to evaluate possibilities and opportunities for operation.
- Level 4 attempts to refine the process in order to improve results by continuous iteration of the four previous levels, adding additional sources and modifying each level process.

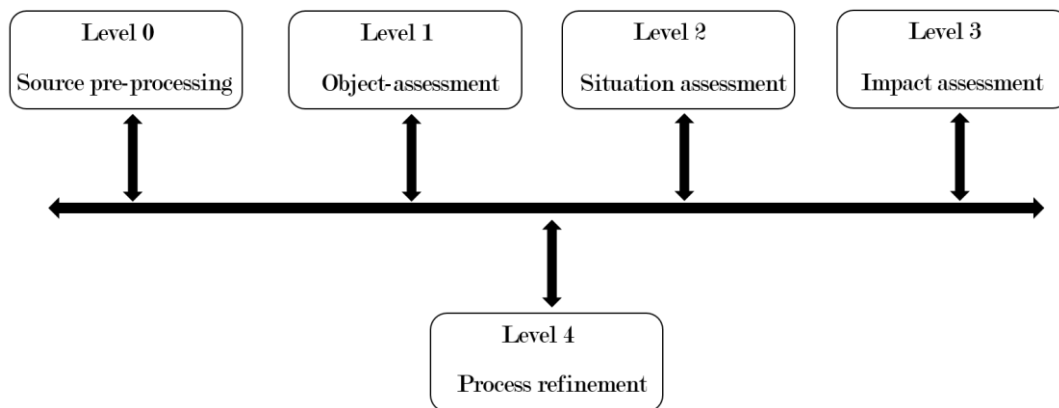


Figure 4-5 JDL data fusion architecture.

The main drawback of the JDL model is its restriction to the military domain. Extension to cover wider domains include research investigations of (Hall and Llinas, 1997), (Steinberg, Bowman and White, 1999), and (Llinas et al., 2004). However, in these extensions, having separate multiple interfaces (human and machine interfaces) at each level, as well as the multiple interactions between levels, can increase analysis result uncertainty.

4.2.2.3 Boyd model

Like the JDL model, the Boyd model was also developed for military command purposes. Also known as observe, orient, decide, and act (OODA) model (Boyd, 1976), it iterates through four primary stages, as shown in Figure 4-6.

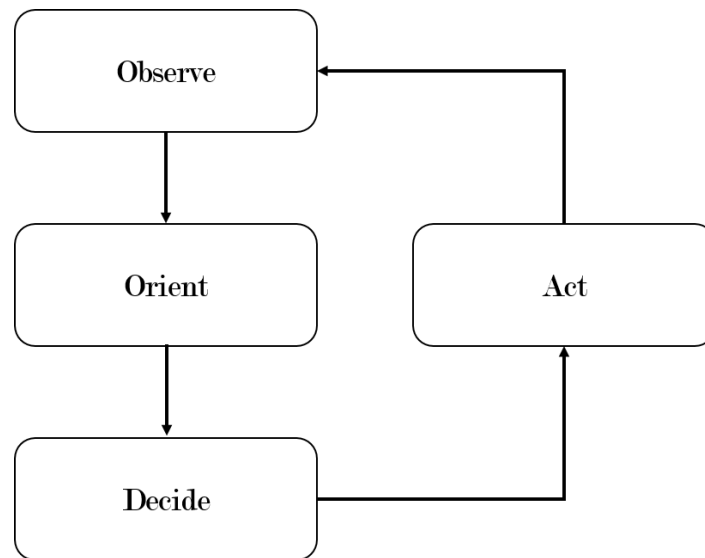


Figure 4-6 Architecture of the Boyd model.

- The Observe level is sensor signals acquisition and pre-processing.
- The Orient level is similar to joint level 1, 2, and 3 of the JDL model. It aims to provide a picture of the context and highlight eventual possibilities.
- The Decision level evaluates different actions that could be taken in order to find the best action to minimise risk and costs. This level is similar to the level 4 of the JDL model.
- The main goal of the Action level is to select actions and assess their impacts.

In other real-world applications, Boyd model levels are often parallelised with continuous refinement of each level. Although the Boyd model offers more flexibility than the JDL model empirically, it is more difficult to implement. There is no clear explanation of the steps. This has led to the creation of the Omnibus model described in the next section.

4.2.2.4 Omnibus model

The Omnibus model gives detailed steps and transitions to the Boyd model (Bedworth and O'Brien, 2000). At each level, detailed steps are explained using classical sensor fusion architecture, as described in the single sensor model in section 4.2.1.

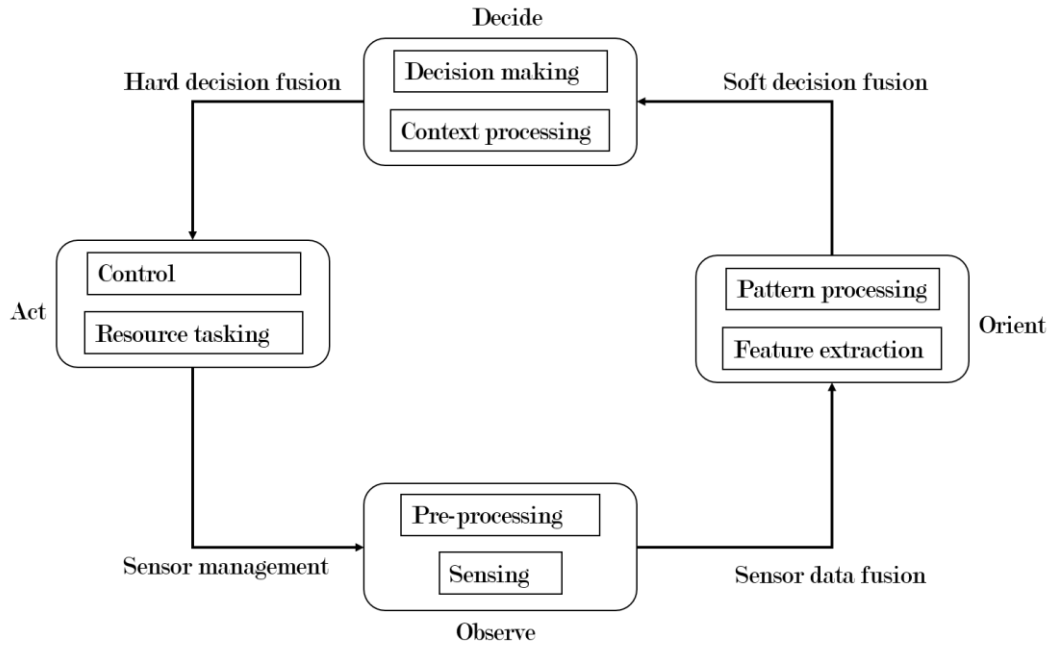


Figure 4-7 Omnibus data fusion architecture.

4.3 Proposed Architecture

As discussed, the existing architectures have been applied for real-world data fusion applications. Whilst some architectures suffer from the lack of feedback, which prevent models from evolving over time, others are specific and hence do not offer a broader application. As a way to resolve the issue, new generic architectures have been proposed by Thomopoulos, (1990), Hou, Shi and Lin, (2000), and Sanfeliu et al., (2010). However, they still suffer from adaptation, especially in multichannel time series. They are often generic and do not provide clear key steps for efficient time series analysis. There is therefore a need for a novel architecture for time series fusion, which this thesis proposes.

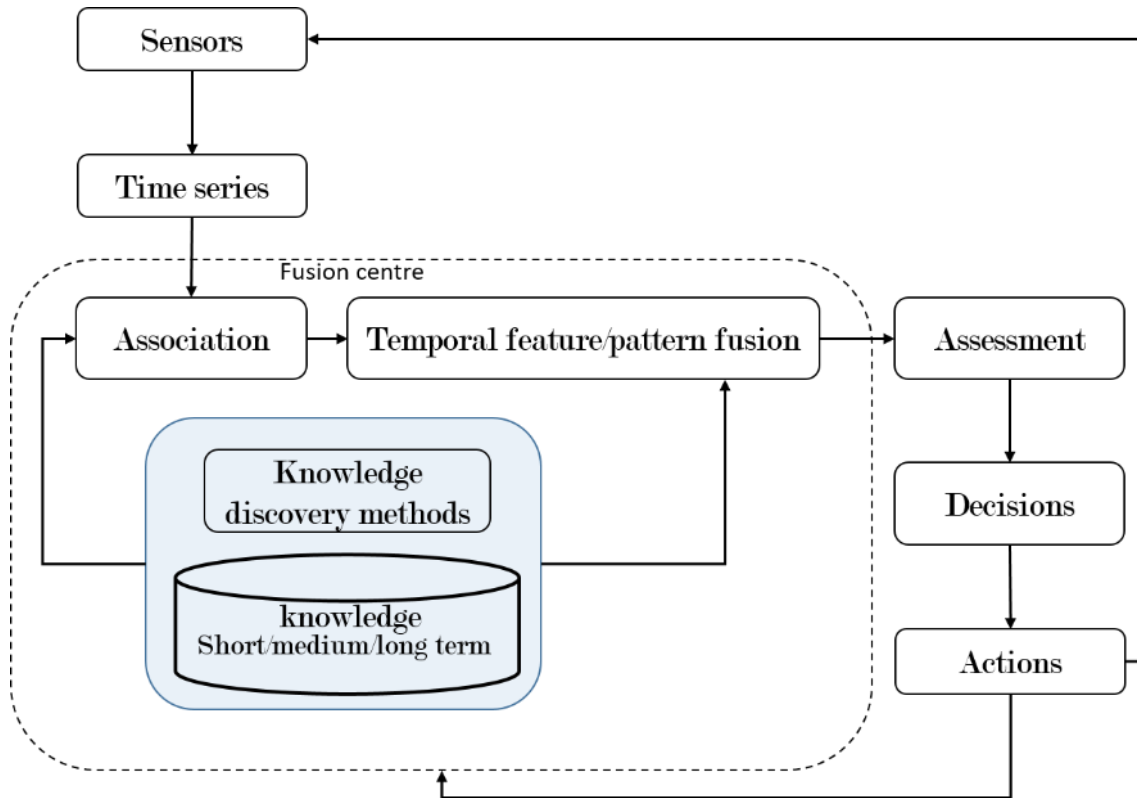


Figure 4-8 Novel time series architecture.

Figure 4-8 represents the novel architecture. It consists of seven main levels. The description is as follows:

- **Sensors:** Like the previous architectures, this step provides measurements of the environment. At this level, appropriate sensor signals for the purpose of the analysis have been selected. Critical elements that need consideration at this stage include a time series sensor system (for example transducer, radar, sonar, spectroscopy, and ultrasonic); the distribution (whether the sensors are distributed or not); the accuracy; the resolution; and the computational capabilities.
- **Time series:** This stage aims to define key properties of the time series generated by the sensors. The key properties are:
 - Time series type, which can be binary, categorical, numerical, and sequential text data or images.
 - Time format where the observations over time can be sparse events, regular, periodical, or continuous.

- Time series dimension. It can be unique or multichannel (fleet time series for instance), and each can be univariate or multivariate within each observation.
- **Association:** This step aims to study key factors enabling the time series fusion with the support of prior knowledge of the studied system and appropriate knowledge discovery techniques. Relevant steps in the association stage include:
 - *Synchronisation and time alignment.* This is a critical step in multivariate and multichannel time series analysis as real-world datasets often come with wrongly time-indexed signals and this step aims to provide time sequence coherence amongst different measurements.
 - *Pre-processing.* This is similar to pre-processing steps in previous architectures, as this is when signals are cleaned and imputed individually and collectively. Domain knowledge helps for both individual and collective pre-processing.
 - *Segmentation.* The segmentation step is often ignored in other architectures, but represents a key step for time series analysis. Real-world system signals are too complex to be analysed with a single model and therefore segmentation is key to divide the signal into sequences representing contexts, such as events or changes, to facilitate the analysis step. Like the pre-processing step, it may require short or medium-term knowledge.
 - *Feature extraction.* This step includes relevant feature extraction. It comprises individual or collective features. In the case of high dimensionality, a dimension reduction is often necessary to extract new features that are characteristic of the system. This step is also often knowledge dependent as the features may be related to the physical dynamics.
- **Pattern/feature fusion.** This aims to use extracted features to implement a method for the purpose of the analysis (diagnosis and prognosis, for

instance). Depending on the context knowledge and the features extracted, methods that are unique, hybrid, multiple, or ensemble can be built for efficient analysis. Visualisations are important for both evaluation and decision-making stages.

- **Assessment.** This is similar, for example, to the situation assessment in the JDL model. It assesses the model's performance and the pertinence of the results, using key indicators such as false alarm rates, accuracy, and homogeneity measures. Additionally, further evaluation is made through benchmarking procedures.
- **Decisions.** During this phase, the optimal decisions are made considering the evaluations in the previous stage.
- **Actions.** Actions include doing a new iteration to improve the performance and resource tasking.

Significant advantages of using this architecture are its ability to handle distributed and normal data fusion and its feedback system, which enables process improvement at important stages. Time series often requires a different level of fusion that depends on the importance and relation of the signal channels. Additionally, an online analysis may require prior offline training or parameter estimations. Furthermore, when the studied system evolves or is not well known, the use of iterative analysis is essential in order to achieve better fusion with integration of insights and knowledge obtained so far. The proposed architecture allows handling these contexts, which are often specific for real-world time dependent observation processing. The architecture is also suitable for sequential or spatial signals studies. Like other centralised multiple fusion architectures presented in Figure 4-4, this new architecture can be challenging to implement as it may involve an independent study of each sensor's records as well as an additional pre-processing step to establish the best way of combining them.

4.4 Conclusion

This chapter provided a review of the main data fusion architectures and proposed a novel architecture for time series fusion. The reviewed architectures included a single sensor, centralised fusion encompassing a waterfall model; JDL; Boyd; and Omnibus models. Each architecture has advantages and drawbacks; some of which are highlighted in Table 4-1. As there is a lack of appropriate architecture handling specific time series data, a novel time series architecture has been proposed. Although the architecture seems to be complex to implement, it allows flexibility and incremental iteration for efficient decision making. This architecture is used for the case studies that are explored in the next chapters.

Table 4-1 Advantages and drawbacks of data fusion architectures.

Architecture	Advantages	Drawbacks
Single sensor	Simple to implement	No feedback Less accurate in general
Centralised fusion	Flexible Efficient in linear signal	One way decision No feedback Complex to implement
JDL	Continuous process improvement	Restrictive to the military and strategical defence industry
Boyd	Continuous process improvement	Restrictive to the defence domain No clear explanation within the steps
Omnibus	Offer more flexibility than the Boyd model Continuous process improvement	Restricted to the defence industry
Proposed approach	Flexible Feedback and incremental process improvement	More focused on Time series analysis

Efficient decision-making sensor fusion architectures, like the proposed architecture, have to take into account:

- evolving environments. The sensing environment is important for appropriate signals selection (sensor accuracy and degradation) as well as the design of the transmission model. For example, high-risk environments require accurate and fault-tolerant architectures, and noisy and dangerous environments require specific sensors that can deal with such context.
- the nature of the sensors and their relevance to the purpose of the studies. The nature of the sensors includes their accuracy, topology (single, parallel, and distributed), and their frequency of measurements. By considering their relevance, this will help to select adequate signals that focus on achieving the aims of the analysis.
- a trade-off between genericity and specificity. Whilst generic architecture is applicable to a wide range of domains, it also comes with considerable customisation costs. Alternatively, architecture that is too specific can lead to adaptation costs for other applications.
- feedback process. Architectures should enable the integration of consequences and knowledge gained to improve different levels of the architecture and therefore enhance the process.
- knowledge of the underlying system. It is essential to encompass short, medium, and long-term knowledge at key stages for optimal decision making.

The next chapter presents a detailed diagnosis of the door faults case study using the proposed architecture.

5 DMU DOOR FAULTS DIAGNOSIS

5.1 Introduction

With the improvement of infrastructure and connectivity within and between cities, the railway sector has witnessed a growth in their customer numbers over the years. This growth creates challenges pertaining to capacity and transporting all the passengers, especially during peak travel times. Peak times often cause passenger congestion, which leads to door obstructions. Consequently, this significantly contributes towards their degradation.

Doors are one of the most frequently used assets on a train, and their failure directly affects passengers' journeys as it may cause train delays and injuries. Furthermore, door failure can be extremely costly for the TOC as it requires cancelling the train journey and redirecting the train to a specific depot for diagnosis and maintenance.

This chapter studies the diagnosis of train door faults. Data collected include door motion and current whilst opening and closing. From this historical data, features that are relevant to door degradation are extracted. The PCA method is then used to combine features in order to produce condition indicators. Due to the lack of degradation knowledge in the data, the k-means clustering algorithm is used to cluster the condition indicators at the first iteration. After the recognition of the clusters (normal and faulty), different degradation rates in accordance with the failure behaviour are simulated and injected into the original data for fault classification.

5.2 Background

Research on train door analysis is lacking. In current research, two types of approaches are used for door fault diagnosis: model-driven and data-driven.

Model-driven approaches seek to design door dynamics using a mathematical model. The output of the model is then compared with the real output measurement throughout the residual analysis. Several works in model-driven

methods encompass differential equation-based modelling (Dassanayake et al., 2009), and rough sets (Shuai et al., 2014). Model-driven approaches are accurate when not many components are involved and when a lot is known about the studied system. A low number of model-driven methods used for door fault diagnosis are available in the literature.

Data-driven approaches are the most used methods for door fault diagnosis. They build methods based on observations, separation (classification), or grouping (clustering). Methods include self-organising maps (SOM) (Lee et al., 2016); logistic regression (Yan and Lee, 2005); dynamic time warping (Yoon et al., 2016); and convolutional neural networks (CNN) (Ince et al., 2016); and SVM (Lee et al., 2016).

This study also explores data-driven approaches, but differs from the proposed methods in the literature because prior knowledge of the fault behaviour is not available and the door operation modes are not unique. Moreover, proposed methods in the literature are too specific to the door types and extracted signal properties are too generic and may not be able to detect any other changes.

5.3 Proposed Methodology

Figure 5-1 describes the workflow methodology used to combat the aforementioned challenges such as poor knowledge of the studied system, missing data values, noise, and outliers. This methodology workflow is an example of the architecture that was proposed in section 4.3.

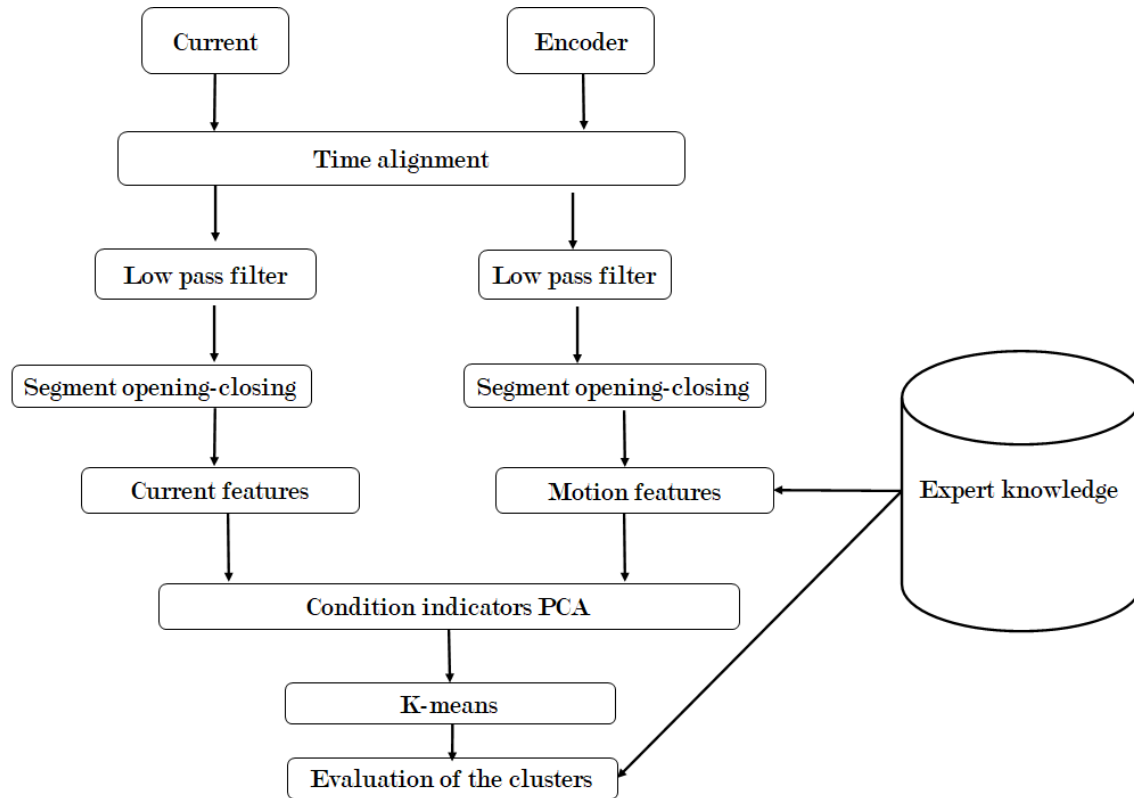


Figure 5-1 Methodology workflow.

5.3.1 Data and Time Alignment

The door data is collected at a frequency of 20 Hz and is taken from many doors from the same train that have similar opening and closing profiles. The time lag is often observed between the motion profile and the current. To align the time series, the dynamic time warping (DTW) method is used for the first alignment. After doing so, the starting time and settling time are used with the presumption that the current should increase before the door motion increases. Figure 5-2 and Figure 5-3 show the raw and cleansed data of the opening and closing profiles. In the opening profile, the speed, as well as the current, increased steadily up to its maximum. A slight flat curve follows it and then it decreases to zero. A dip can be seen during the decrease phase, at roughly half the maximum value, due to a regulating system (controller) avoiding a fast decrease, which could lead to severe damage to the door. As it reached its maximum open position, the door stayed in the “open state” for several seconds before starting the closing motion.

The closing profile follows a similar pattern with three main differences visible in the current (in ampere) motion:

- The peak of the current in the closing profile is lower than the opening, leading to a larger closing window.
- Depending on the position of the door to the ground floor (tilted or flat ground), an additional gravitational speed may be added. A controller, therefore, adjusts the door speed by decreasing or increasing the current. These fluctuations can be seen in the current speed profile within the time range from eight to ten seconds.
- At the end of the closing profile, an abrupt decrease in the current can be seen, followed by a slight bump, indicating an increase in speed. This helps to push the door to its maximum reachable position where a locking process can be triggered.

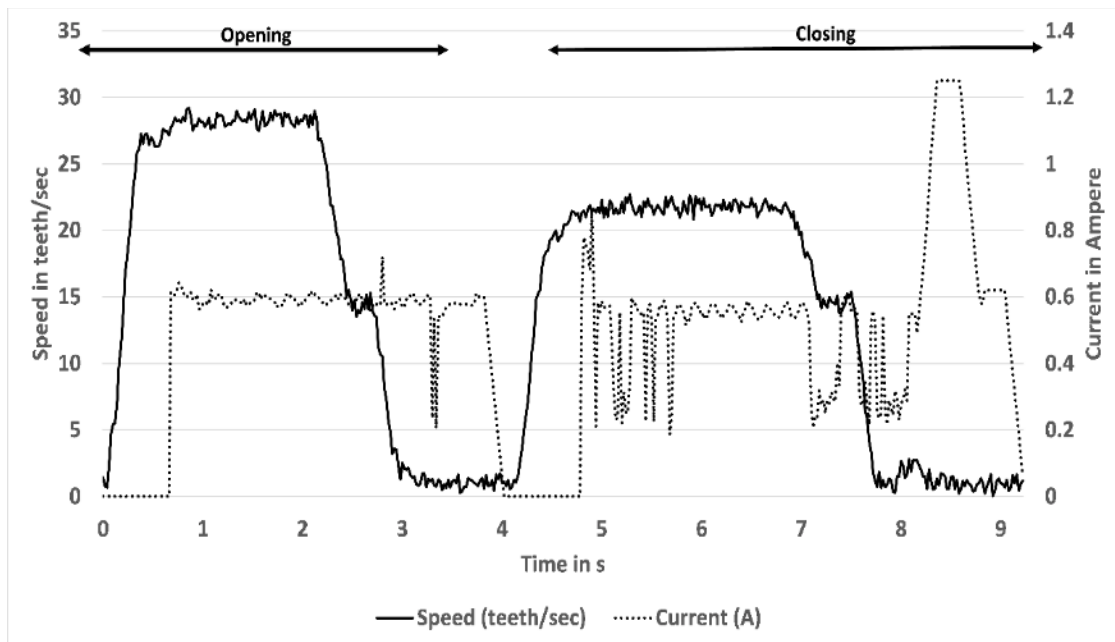


Figure 5-2 Example of a raw data profile of a door opening and closing.

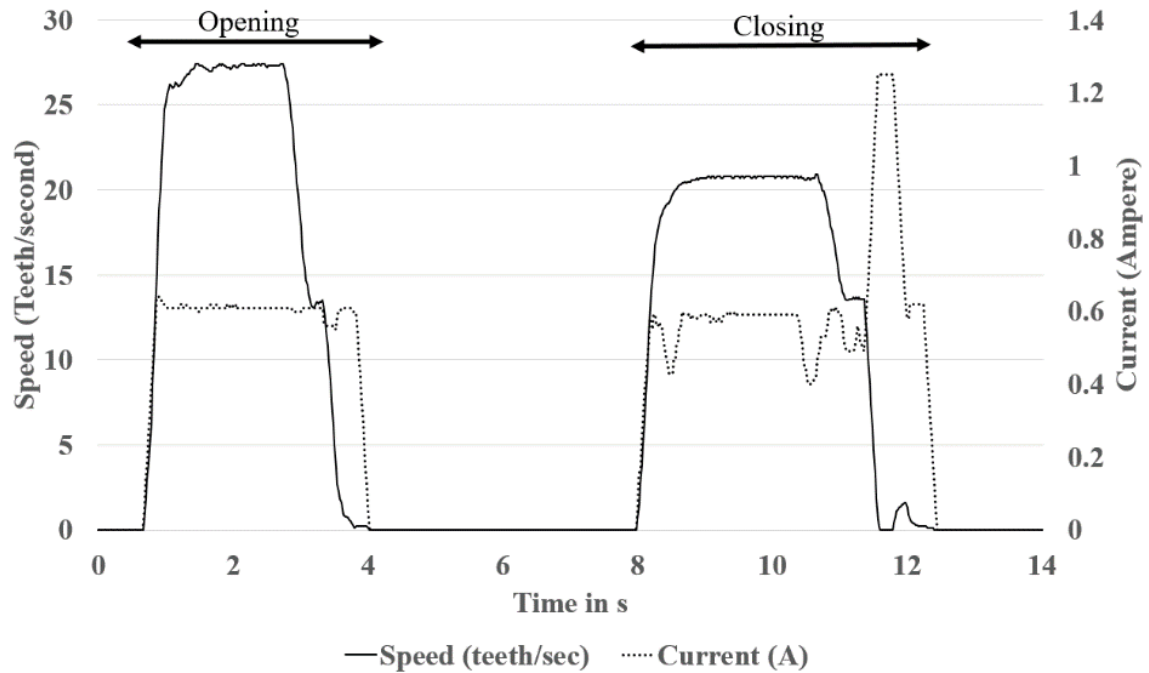


Figure 5-3 Example of a cleaned and aligned of a door opening and closing.

5.3.2 Cleaning, Segmentation and Local Features Extraction

To reduce noise carried by both current and encoder signals, a low pass filter is applied to a window of 0.25 seconds, representing five data points in a second. Afterwards, the opening and closing motion is segmented, by splitting the signal when the current drops to zero. To distinguish the opening and closing profile, the peak current is identified (during the closing motion, the peak of the current reaches 1.2 amperes). As described in Table 5-1, the following features for both current and encoder signals that are extracted are:

- Movement time and gradient: time/gradient required to reach the desired position.
- Overshoot: maximum peak value of the response signal measured from the desired steady-state value.
- Settling time: time elapsed from the application of a change in the motion profile until the system reaches steady operation.
- Steady-state means and standard deviation: the mean and the standard deviation of the values reached after the oscillations are extinguished (indication of the average current fed during the movement).

- Maximum peak in steady-state region: maximum deviation over the mean value of the signal after the settling time.

Table 5-1 Features extracted from door profiles.

Signal	Motion	Feature
Speed	Opening	Overshoot
		Steady-state mean
		Settling time
		Movement time
		Gradients
	Closing	Steady-state standard deviation
		Overshoot
		Steady-state mean
		Settling time
		Gradients
Current	Opening	Steady-state standard deviation
		Overshoot
		Steady-state mean
		Settling time
		Movement time
	Closing	Gradients
		Steady-state standard deviation
		Overshoot
		Steady-state mean
		Settling time
Current	Closing	Movement time
		Gradients
		Steady-state standard deviation
		Overshoot
		Steady-state mean

5.3.3 Condition Indicators Extraction

Condition indicators are extracted using PCA as it allows the projection of features into lower dimensional space (Chiang, Russell and Braatz, 2001). Previous related work has found PCA useful for condition indicator extraction and asset health assessment (Ruiz-Carcel and Starr, 2018; Ham et al., 2019). Given matrix $\mathbf{X}$ with n observations, m variables, and covariance $\mathbf{S}$, loading vectors $\mathbf{V}$ can be obtained through $\mathbf{S}$ eigenvalues decomposition methods such that:

$$\mathbf{S} = \frac{1}{n-1} \mathbf{X}^T \mathbf{X} = \mathbf{V} \mathbf{D} \mathbf{V}^T \quad (5-1)$$

Where $\mathbf{D}$, a diagonal matrix, contains the eigenvalues representing the amount of variance expressed. The loading vector $\mathbf{V}$ is ordered according to these eigenvalues. The k largest loading vectors associated with the top k largest eigenvalues are selected, represented by $\mathbf{R}$, and representing the new orthogonal basis of the feature space. Original features are projected on this basis, which has a lower dimension ($a < n$). This projection allows systematic trends to be captured as well as the reduction of random noise and measurement inaccuracies (Chiang, Russell and Braatz, 2001). The score matrix $\mathbf{W}$ and the residual $\mathbf{E}$ represent the new projected features, and the difference between the original features $\mathbf{X}$ and the projection of $\mathbf{W}$ back to the original m dimensional space are as follows:

$$\mathbf{W} = \mathbf{X} \mathbf{R} \quad (5-2)$$

$$\mathbf{E} = \mathbf{X} - \mathbf{W} \mathbf{R}^T \quad (5-3)$$

5.3.4 Clustering

K-means algorithm is used to cluster condition indicators represented by the matrix $\mathbf{W}$. Given $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_p, \dots, \mathbf{w}_n)$ where each $\mathbf{w}_p$ is the p^{th} feature and a the number of centroids, k-means++ aims to optimise the following equation:

$$\min_{C_i} \sum_{i=0}^{i=a} \sum_{x \in P_i} ||x - C_i||_2 \quad (5-4)$$

Where C_i is the i^{th} cluster; x a feature; P_i the set of the features belonging to the cluster; i and $|| \cdot ||_2$ is the L₂-norm. Instead of random initialisation of the centroid, the k-means++ algorithm is used (Arthur and Vassilvitskii, 2007).

5.3.5 Classification Methods

K-nearest neighbour (k-NN) classification methods are used to classify the labelled observations with the injected faults. It is an instance-based learning method that applies neighbour majority vote to assign a new given observation. During the training phase, the algorithm constructs a list of observations which is used in the classification phase to estimate a kernel assignment function (Bishop, 2013).

5.3.6 Evaluation

Three performance indicators are involved in the assessment of the classification approaches: the recall, the precision, and the F1-score. The precision measures the percentage of annotation correctness, whereas the recall measures the percentage of annotated change to be a true one. The F1-score is the test accuracy metric representing the correctness of all identified cases.

$$\text{Precision}(P) = \frac{\text{Number of Correct annotations}}{\text{Number of annotation}} \cdot 100 \quad (5-5)$$

$$\text{Recall}(R) = \frac{\text{Number of Correct annotations}}{\text{Number of true annotations}} \cdot 100 \quad (5-6)$$

$$\text{F1-score}(F1) = 2 \frac{P \cdot R}{P + R} \quad (5-7)$$

The k-NN method is evaluated using these performance indicators. It is also compared with alternative methods, namely decision tree, SVM, random forest, multilayer perceptron neural networks (MLP), and naïve Bayes.

5.4 Experiments

Experiments are carried out using 1,000,000 observations, each representing a door's current and speed (in encoder steps per second). For each signal, ten features are extracted, as previously detailed in Table 5-1. Features are then column z-normalised to remove unit measurement dependency. Then, PCA is applied, and three principal components retaining 98% of the proportion of the variance are selected, as shown in Figure 5-4. The projection of the original feature into the new space is shown in Figure 5-5. Four clusters appeared and were regrouped using the k-mean algorithm with the number of clusters $a = 4$.

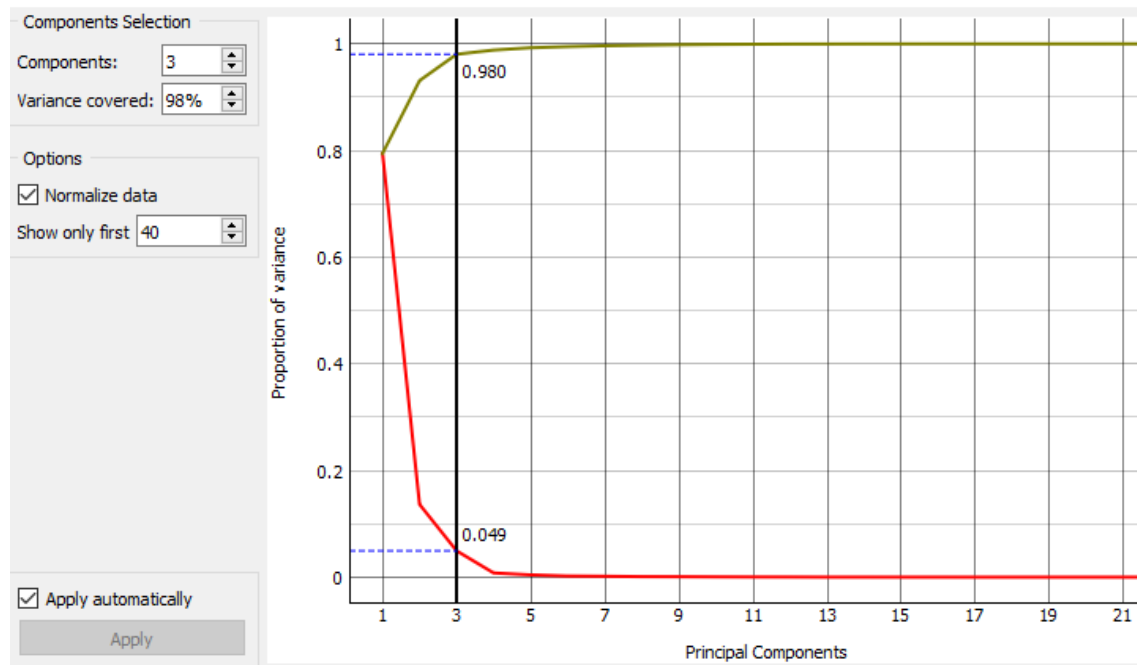


Figure 5-4 PCA principal components selection.

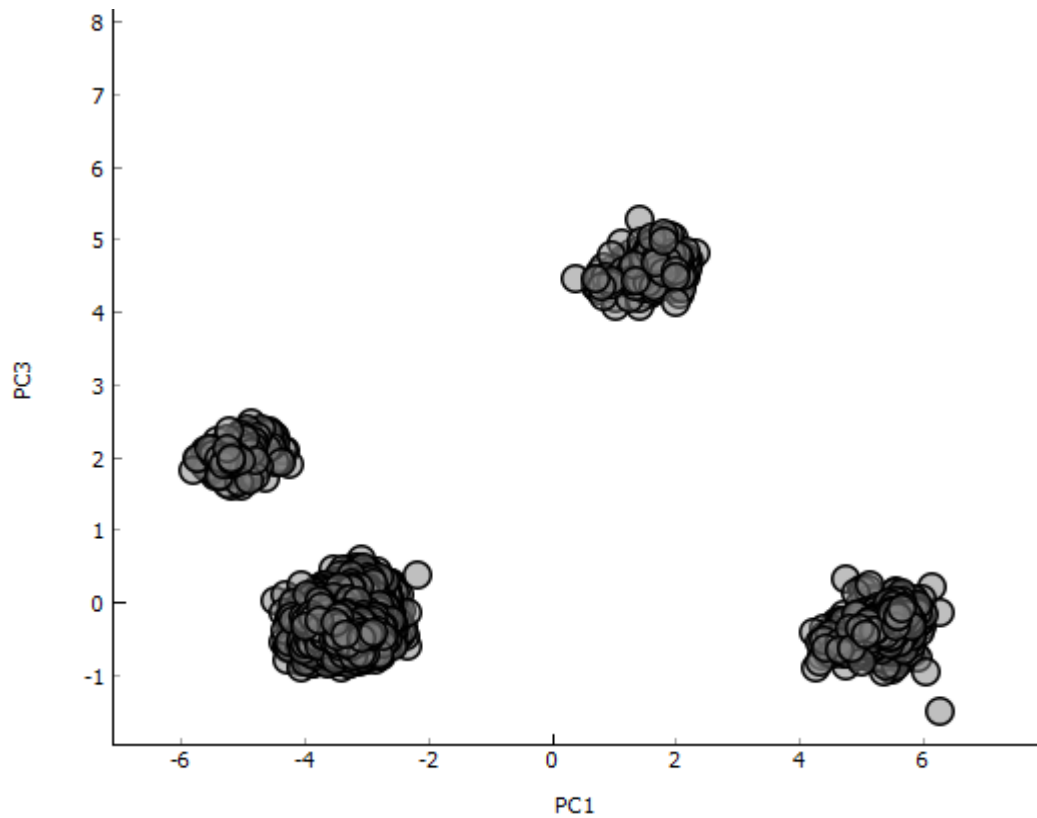


Figure 5-5 PCA condition indicators.

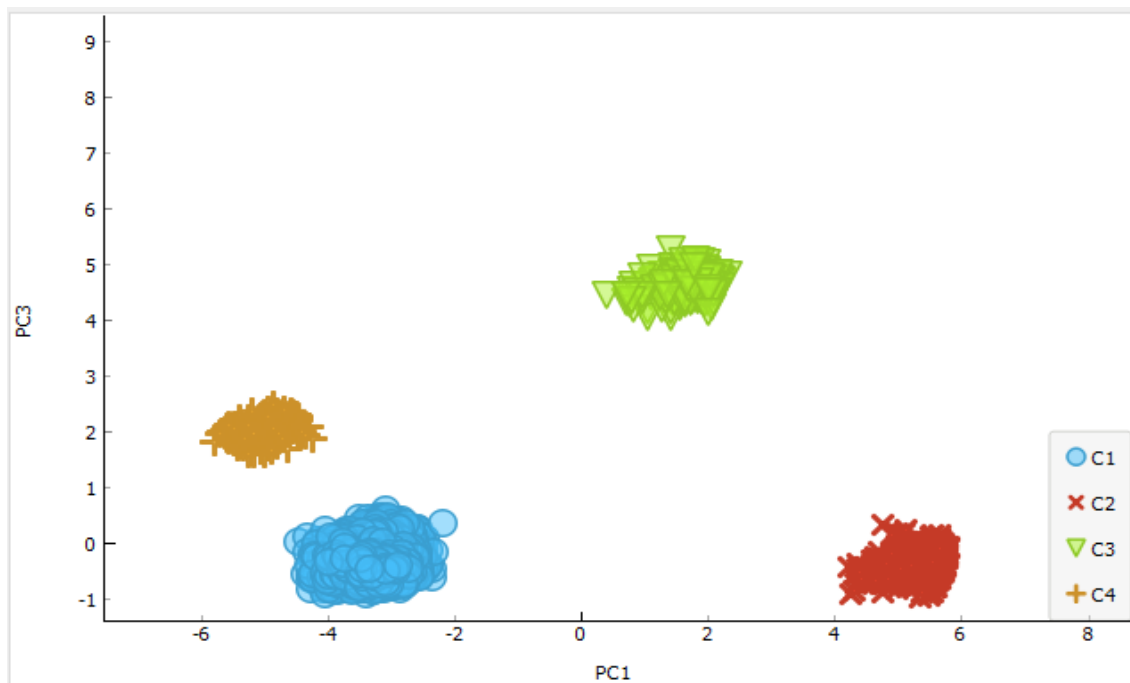


Figure 5-6 K-means clustering results with the clusters C1, C2, C3, and C4.

5.5 Results and Discussion

Results unexpectedly show four very distinct clusters appearing in the k-means clustering algorithm with no transitional observations within the clusters. The first conclusion that can be made is that, as most of the data represent normal conditions, the cluster C1 (containing the majority of the observations) represents the normal operation of the doors, whereas the remaining clusters are the faults occurring in the system. However, joint in-depth investigations with experts show that the doors are operating with two normal operation modes. Clusters C1 and C2 are their characteristics and contain the majority of observations. Clusters C3 and C4 are a manifestation of the same intermittent actuator fault in the two operating modes. A change of a normal mode affects the magnitude of the fault, causing the apparition of clusters C3 and C4. Figure 5-7 and Figure 5-8 represent examples of each normal operation mode and the fault behaviour observed, respectively. Red circles show that initial features extracted such as the speed profile gradients, the overshoot, peak current, and mean values during the steady-state are sensitive to this particular observed fault.

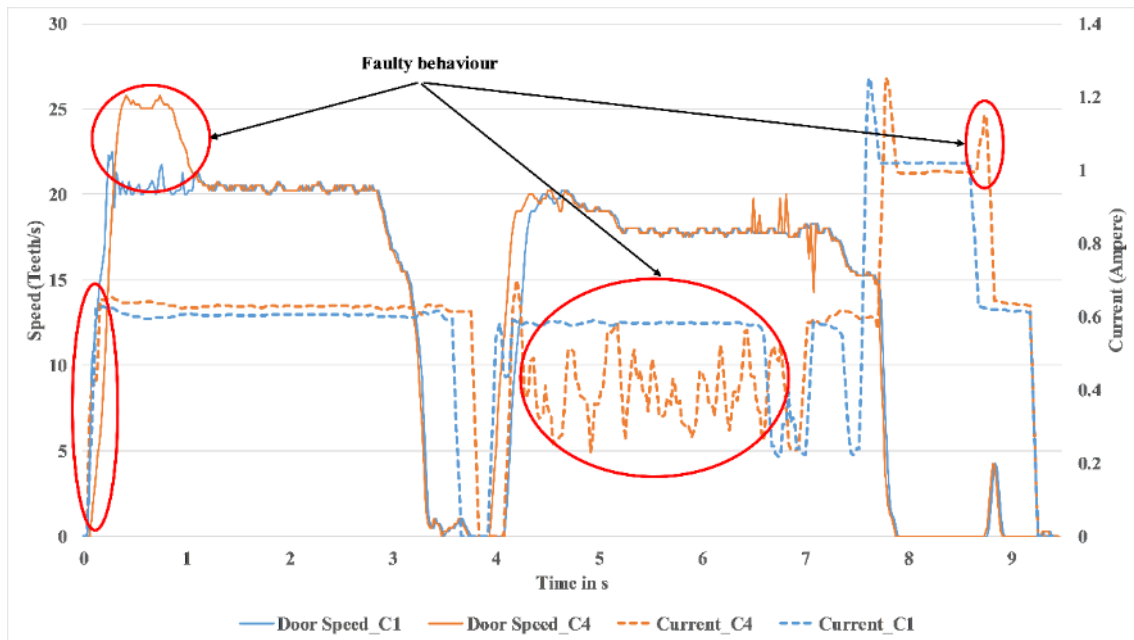


Figure 5-7 Cluster C1 and intermittent fault associated in cluster C4.

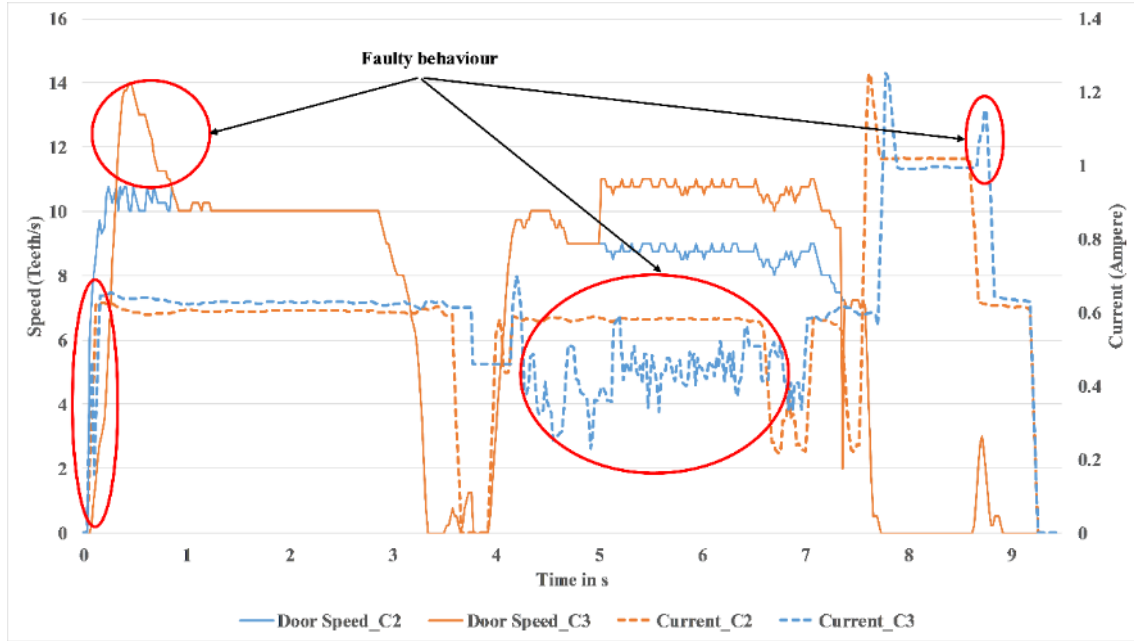


Figure 5-8 Cluster C2 and intermittent faults associated in cluster C3.

In order to achieve fault classification, artificial faults have been injected into the original data using the centroid profile of each cluster with a degradation rate of 10%, 30%, and 60%. The amount of injected fault is 90% the amount of normal behaviour to solve the imbalance issues that arose in the classification. The newly labelled data are segmented into 60% of training data, 20% of testing data, and 20% of validation data. Classification is made using k-NN and comparative alternative methods, such as decision tree, SVM, random forest, artificial feedforward neural network, and naïve Bayes methods. Table 5-2 shows empirically evaluated parameters used for each method.

Table 5-2 Parameter setting for the classification methods.

	Parameters
Data	Number of observations: 1 000 000
	Number of PCA features: 3
k-NN	Number of neighbours: 20
	Distance metric: Mahalanobis
	Weights: Uniform
Decision tree	Minimal leaves: 20
	Maximal tree depth: 100
SVM	Cost: 1
	Regression loss: 0.20
	Kernel: RBF
Random forest	Number of trees: 10
Neural network	Hidden neurons per layer: 50, 25,10
	Activation per layer: reLu, sigmoid, sigmoid
	Solver: Adam
	Adam regularisation parameter: 0.05

Table 5-3 shows the results of the experiments carried out by segmenting the data in each operating mode before classification (split) as well as the classification without prior segmentation into operating mode. The colour represents a visualisation of the performance for each metric. Green indicates better performance. In the split tests, the Naïve Bayes method shows a lower F1-score (74.83%), followed by the decision tree (92.08%). The neural network, random forest, and SVM show similar performance in F1-score, precision, and recall. The k-NN method highlights the highest performance score out of all of the indicators. Without a data split, the SVM shows the lowest F1-score due to poor recall performance (77.37%); neural networks, random forest, and the decision tree present slightly similar results.

Table 5-3 Validation results with and without split mode before training.

Model	Split			Without split		
	F1	Precision	Recall	F1	Precision	Recall
k-NN	98.45	98.69	98.29	97.61	97.73	97.49
Decision tree	92.08	91.97	92.2	92.18	92.05	92.31
SVM	94.37	94.59	94.15	80.28	83.42	77.37
Random forest	93.43	93.38	93.48	94.09	94.05	94.13
Neural network	93.91	94.13	93.69	93.63	93.84	93.43
Naive Bayes	74.83	72.47	77.35	80.91	80.37	81.46

In general, k-NN, random forest, neural networks, and decision tree show good results in both types of tests. The k-NN methods have the best performance with consistently more than 97% of prediction accuracy in all of the indicators. Additionally, results suggest detecting the mode (split) before fault detection before performing the classification.

A limitation of the proposed approaches is the empirical tests on the parameters. Although many parameter values have been tested, there is no formal guarantee that the best parameter values used are indeed optimal. Further drawbacks of the k-NN method include the temporal complexity with large observations as well as the classifier learning stage often being time consuming.

5.6 Conclusion

This chapter proposed a straightforward methodology derived from the proposed architecture of the fault diagnosis using door data. Relevant features such as the overshoot, gradients, and steady-state standard deviations were extracted in order to have a good representation of the system dynamics. PCA technique was applied to the extracted features to extract condition indicators. With expert knowledge and k-means techniques, two operating modes of the doors and faults were found which therefore allows it different degradation stage modelling for faults stage classification. The k-NN, SVM, decision tree, naïve Bayes, and feedforward artificial neural network performance were investigated using F1-score, precision, and recall indicators.

Analysis results on the classification methods highlight that the k-NN approach consistently shows the best F1-score, precision, and recall (above 97%). Additionally, results suggest that it is better to first detect in which mode the door operated before applying the classification methods.

A main limitation of the present investigation is that the classification methods' parameters have been tested empirically and do not guarantee optimality. This can lead to a bias in the comparison of the approaches. An additional limitation is the computational efficiency of the selected classification method (k-NN), as classifying large observations can be time consuming.

Areas of the presented methodology that require improvement include:

- A further investigation in real data with different fault conditions and different degradation stages. The analysis was undertaken using real data in the first stage. However, faults were already found in the last stage, leading to the injection of artificial intermediate degradation stage data for the classification. Furthermore, it would have been useful to capture more door data for a thorough analysis.
- Parameter estimations. As already mentioned, the parameters have been tested empirically and therefore may not reflect the true performance of the methods. The further research direction is automatic parameter estimations for better method evaluations.

6 DMU TRAIN ENGINES FAULT DIAGNOSIS

6.1 Introduction

Diesel engines are not new. Their primary design and implementation date back to Dr. Rudolf Diesel's work in the nineteenth century. His work was dedicated to creating engines that enabled the continuous injection, compression, and combustion of gas (air) and fuel (diesel) to produce power. Since then, they have evolved with a wide range of applications within transportation (cars, trains, ships, and aircraft); agriculture (water pumps and trucks); the military; and others such as electricity generation and grinding mills. As the engines are the core elements of these systems, maintaining them is crucial, not only to improve their efficiency, but to also keep passengers safe.

This is also the case within the railway arena. Maintaining engines has multiple benefits, such as reducing maintenance costs and improving passenger journey by detecting early faults and employing proactive maintenance. Moreover, it is environmentally conscious as early reparation or replacement leads to low carbon emissions in these engines. In fact, earlier maintenance allows optimal combustion, which reduces monoxide carbon emissions. Fault diagnosis and condition monitoring are essential for regular assessments of the engines.

In-depth investigations have already been carried out for diesel engine fault diagnosis because their physical dynamics are very well known. However, the analysis often suffers from specificity as physical variables that are not monitored in real life are taken into account.

In this chapter, a novel data-driven fault detection method is proposed in order to fuse multiple channel time series data with less knowledge of the diesel engines. The model uses the extracted key features, representing the dynamics of the engines, to detect the fault. Features extracted are sensitive to deviations from normality. The model estimates conditional dependency amongst features and annotates all of the observations through clustering methods, meaning that the same faulty observations are clustered under the same label.

Experimental results carried out with the DMU engines data show that although the proposed method is difficult to implement and to make prior Gaussian assumptions and i.i.d between features, it is more accurate and efficient than alternative methods.

6.2 Related Work

Diesel engines are systems made up of complex subsystems with different operations and failure modes. Each sub-system may be built with varying types of materials enabling different amounts of heat and power to be generated, depending on the manufacturer. Hence, different types of components may lead to different types of fault characteristics. To guarantee safety and lower risks, manufacturers embed a fault diagnosis system onboard that detects well-known faults. These faults are often diagnosed by expert systems to find fault type, location, and severity, which then trigger important actions such as a user warning in the short term, and maintenance in the medium and long term. However, a further diagnosis still needs to be undertaken as behaviour and degradation of the engines often depends on:

- External conditions, including the evolving environment (external temperature, humidity, altitude); global utilisation rate; and user actions (driver behaviour, for example).
- Internal behaviour such as fuel and air quality injected into the engines; the cooling systems; and leaking transmission systems.

The most common faults observed in diesel engines are power loss; a deficient lubricating system that leads to wear or leaks; thermal overloads; exhaust, smoke; change in the exhaust temperature and pressure; carbon emissions; and change in the rotational speed. Table 6-1 describes the principal components of diesel engines, their associated faults, and the monitored signals for fault diagnosis purpose.

Table 6-1 Diesel engines fault and associated signals monitored.

Components	Faults	Examples of monitored signals
Injectors	Imprecise timing Imprecise pressure Clogged and erode seepages	Acoustic measurements
Cylinder, ring, piston	Misfire Fatigue Knocking Corrosion Defectuous ring Power loss	Rotation speed Cylinder pressure Oil contamination Fuel contamination Temperature
Lubrication system	Incorrect oil pressure High lubricant temperature Oil contamination Oil degradation	Oil Coolant temperature and pressure Timing Particle rate
Inlet, exhaust	Air filter blockages Leakage Excessive emission	Viscosity Acidity Smoke (composition and colour) Vibration frequencies
Heat	High coolant temperature	Exhaust fan temperature
Mechanical	Bearing wear	Acoustic measurements

Several of the monitored signal analysis are described in the following sections.

6.2.1 Noise and Vibrations Analysis

Most of the engine noises and vibrations are a result of combustion and exhaust gases, or some of the engine components making physical contact with each other (Jones and Li, 2000). Key methods for vibration monitoring analysis are:

- Signal processing methods, including transfer function-based methods
- Frequency domain analysis such as FFT
- Wavelets based analysis.

Chiatti, Chiavola and Recco (2014) applied frequency-based analysis for univariate signal vibration analysis in railway engines. Latest work examples include various load failing injector vibration analysis using time-frequency and spectral features or transfer functions (Dykas and Harris, 2017; Taghizadeh-Alisaraei and Mahdavian, 2019; Dong, Nguyen and Lee, 2020; Yang et al., 2020).

6.2.2 Fuel and Oil Analysis

Fuel and oil quality are essential for efficient power generation. In order to evaluate fuel and oil conditions, it is important to consider characteristics such as fuel contamination, timing, oil acidity, viscosity, water ingress, and contamination by foreign particles. An overview of the analysis carried out in the subject is provided by Mwangi *et al.*, (2015). Methods encompass Bayesian-based; fault tree (Ren et al., 2020); pattern recognition approaches such as SVM and Neural Networks (Cai, Zong and Zhang, 2016; Pestana-Viana et al., 2019); and model-driven methods (Dong et al., 2015; Simani, 2013; Krishnaswami, Luh and Rizzoni, 1995).

6.2.3 Exhaust Analysis

Several faults can be diagnosed through exhaust analysis. For example, exhaust gas composition, temperature, and pressure can help to diagnose fuel quality, combustion efficiency, cylinder set, and injection equipment. Fuel as oil analysis methods include model-based, especially observers (Lee, Lee and Sunwoo, 2014; Dubuc et al., 2018).

6.2.4 Speed and Torque

Speed and torque analysis are important to detect loss in power in diesel engines. Most studies are model-driven, as a feedback loop system can be designed when given key inputs such as the demand, injection, and previous speed. These models can then be compared to the real speed measured for fault detection. Examples of this work include (Zweiri, Whidborne and Seneviratne, 2001), (Jung and Sundstrom, 2017), and (Belyaev et al., 2018).

6.3 Challenges and Open Research Issues

In general, speed and torque analysis methods are model-driven, which means designing models that are able to estimate the outputs when given the input observations. This estimation is then compared with the real system output to determine fault diagnosis. However, when a complete picture of the system is not available (for instance, only several output observations are monitored), existing methods are inefficient for fault detection. Moreover, there is a lack of fault diagnosis investigation in such context using multichannel engines. This leaves room to propose a new way to detect faults. In response to this, this thesis proposes a graphical model for multichannel time series fault detection.

Challenges in this analysis encompass understanding patterns highlighted in the data, cleaning and aligning the signals, and proposing a novel method that helps to detect and to localise faults in time throughout annotation.

6.4 Proposed Methodology

The methodology workflow presented in Figure 6-1 is employed. Each channel time series is cleaned, aligned, and segmented before feature extraction, which are jointly used with the proposed annotated methods for fault diagnosis. The results of the proposed annotation method are evaluated and compared with alternative methods. The steps in the methodology process are described in the following sections.

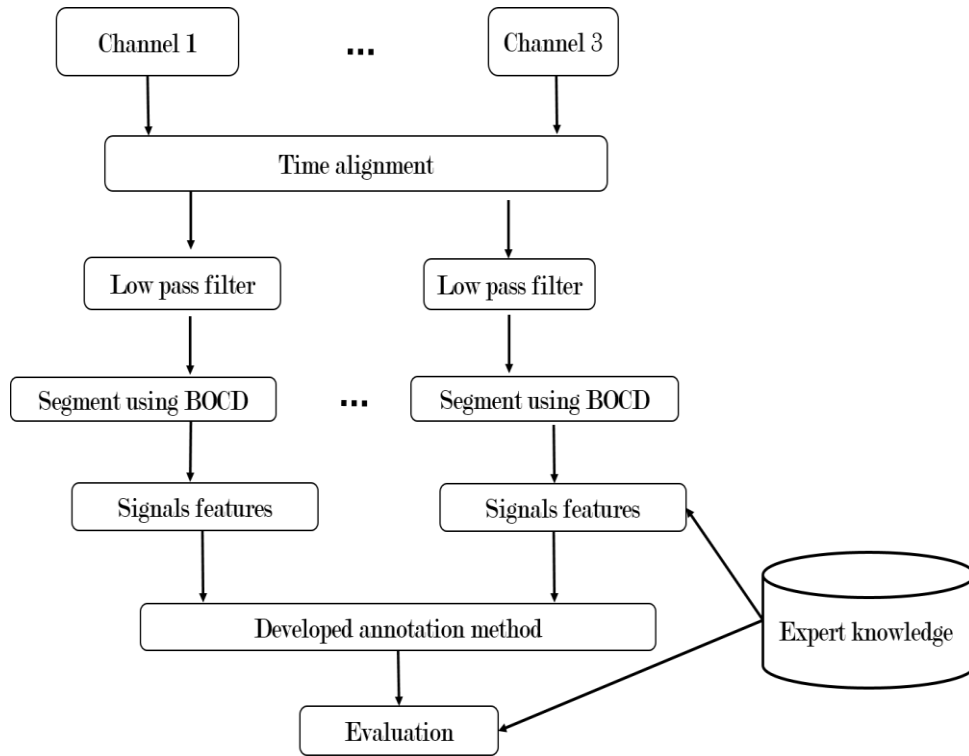


Figure 6-1 Methodology workflow.

6.4.1 Data

Figure 6-2 is an example of the studied DMU train. Each engine represents a time series channel that participates in achieving the unknown driver acceleration demand by producing an amount of traction force. In each engine, sensors retrieve information on the ambient temperature, the charge air temperature and pressure, the engine frequency, the second stage oil, oil cooler temperature and pressure, and the exhaust temperature every second.

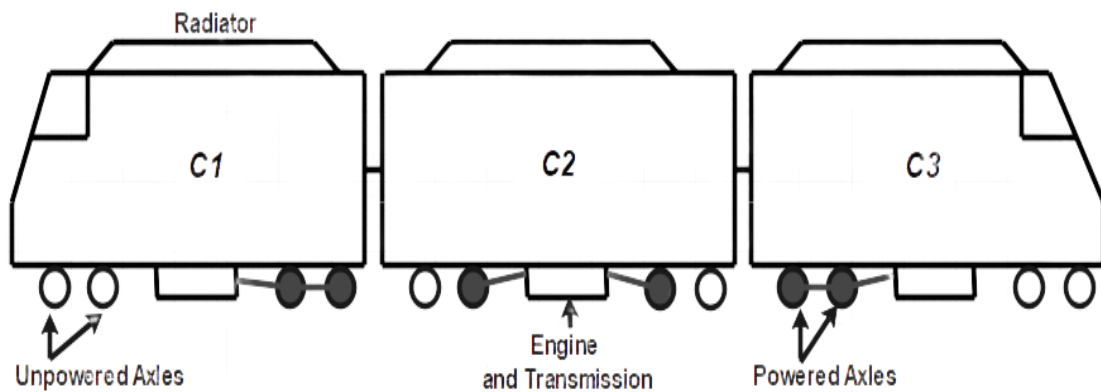


Figure 6-2 DMUs engines in carriage C1, C2 and C3.

Considering, for example, the engine C1; Figure 6-3 shows an example of its rotational speed and its charge air pressure over time. An interesting repetitive pattern can be seen; each pattern is composed of segments representing idling and acceleration to a limited speed (around 1750 rpm). This is followed by a steady speed for several seconds, and then there is a deceleration representing a gear change. An acceleration with a steady upwards slope follows this and then finally, a deceleration to the idling state. Analysing these groups of patterns across all of the engines may be useful in order to detect an evolving deviation in the system.

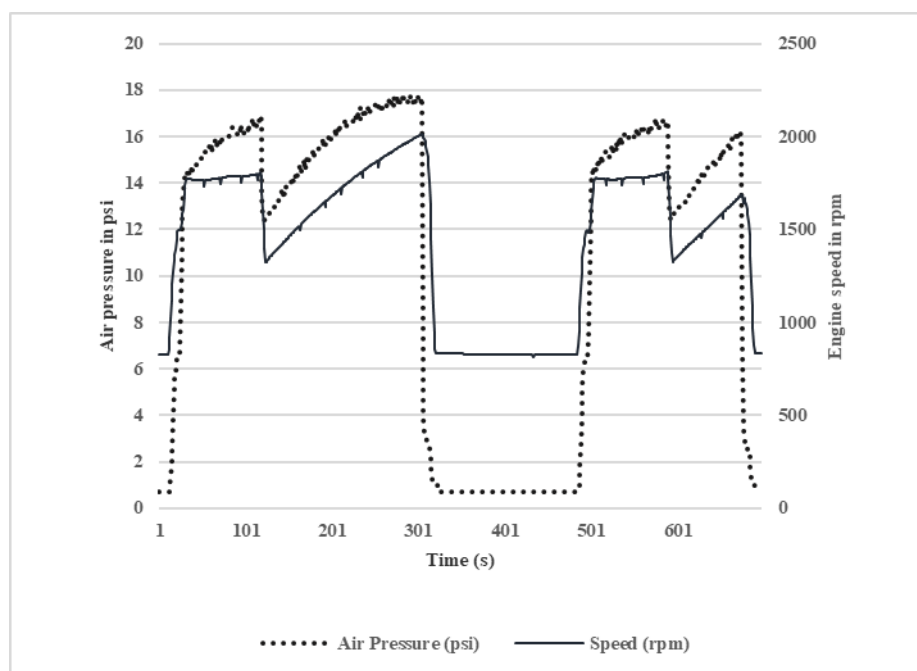


Figure 6-3 Engine frequency and associated charge air pressure over time.

Two types of datasets have been used for the analysis. The first category represents a working condition with white noise where, due to some unknown root cause, the charge air valve has unusual behaviour, leading to a shortage of the air in one of the engines. In the second category, one of the engines often does not generate sufficient power to meet demand, causing low power events.

6.4.2 Time Alignment and Cleaning

Within each channel, there is often time desynchronisation between signal attributes. Therefore, in order to align the data, it is important to choose a common signal between all of the attributes. The easiest signal to work with is linked to the speed of the engine as it is a (semi) controlled signal (by the train driver). For this reason, it has been decided to use the engine frequency as the reference signal to align the collected data. Using this reference signal, the DTW method is used for the alignment.

In order to complete the time alignment within channels, the same method is applied to each carriage engine and then applied to align all of the engines together. This allows the time alignment of all signal attributes of different carriage engines to be completed.

In the pre-processing stage, outliers with fast transitions were removed and replaced with averaged values. Averaged values have also been used to impute the missing values within short windows (less than ten seconds). Missing data in long windows are segmented.

6.4.3 Segmentation

In order to extract relevant features, it is necessary to segment the signals. After comparing the performance of BOCD and Martingale methods, BOCD was chosen for the segmentation as it highlights the best performance indicators (Namoano et al., 2020a).). Long window data gaps have also been used for the segmentation.

6.4.4 Feature Extraction

As already mentioned, feature extraction is critical in order for the methods to be the most effective, as it enables the capture of system dynamics, which is sensitive to deviations from normality.

- Movement time and gradient: time/gradient required to reach the desired position.

- Overshoot: maximum peak value of the response signal measured from the desired steady-state value.
- Settling time: time elapsed from the application of a change in the motion profile until the system reaches steady operation.
- Steady-state mean and standard deviation: the mean and the standard deviation of the values reached after the oscillation are extinguished (indication of the average current fed during the movement).

Maximum peak in steady-state region: maximum deviation over the mean value of the signal after the settling time. Table 6-2 presents the features extracted from the engines' profiles. The features are extracted from acceleration, gear change, deceleration, and the steady-state window. Such features are important to help detect sensor and engine faults.

Table 6-2 Features extracted from Engine datasets.

Signal	Motion	Feature
Frequency	Acceleration	Overshoot
		Settling time
		Gradient
Frequency	Gear change	Steady-state standard deviation
		Gradient
		Dips
Frequency	Deceleration	gradient
		Overshoot
		Settling time
Pressure	Acceleration	Gradient
		Steady-state standard deviation
		Change duration
Pressure	Gear change	Gradient
		Dips
		Gradients
Temperature	All cases	Unexpected peaks
		Steady standard deviation

6.4.5 Proposed Annotation Method

6.4.5.1 Problem Definition

Given a multichannel time series, $\mathbf{T} = \{\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_K\}$, where each $\mathbf{T}_k \in \mathbb{R}^{np}$ a multivariate time series and $\mathbf{T}_{k,i}$ the observation at the time index i of the time series $\mathbf{T}_k$. Each $\mathbf{T}_{k,i}$ is a vector of p variables entries values. The objective is to annotate the K time series observations with S symbols. To examine the dependency between the multivariate time series, two latent variables $\mathbf{C}$ and $\mathbf{X}_k$ are introduced, such that:

$$\mathbf{T}_k = \mathbf{C} + \mathbf{X}_k \quad (6-1)$$

where $\mathbf{C}$ captures the common variations amongst all of the multivariate time series and $\mathbf{X}_k$ the variation within the specific multivariate time series $\mathbf{T}_k$. In some conditions, $\mathbf{C}$ may represent the behaviour shared by all the processes generating such time series, whereas $\mathbf{X}_k$ can be seen as a local behaviour of the component. When all of the time series have the same behaviour, for example the train engine system mentioned above, the $\mathbf{X}_k$ represents component faults or noise measurement as it captures the deviation of the local component from the common behaviour. The present work assumes that latent variables $\mathbf{C}$ and $\{\mathbf{X}_k\}, k = 1, 2, \dots, K$ are mutually independent and follows a Gaussian distribution. It has been shown in such settings (Xie, Liu and Valdar, 2016) that:

$$\text{var}(\mathbf{C}) = \text{cov}(\mathbf{T}_k, \mathbf{T}_l) \quad k \neq l \quad (6-2)$$

The main goal is to affect a row of $\mathbf{X}_{k,i}, k = 1, 2, \dots, K$ samples of the observations to one of the clusters $\mathbf{P}_s, s = 1, 2, \dots, S$, whilst taking into account the temporal consistency. Considering $\boldsymbol{\theta}_s$ the concentration matrix of the cluster $\mathbf{P}_s$, the component layer aims to estimate $\boldsymbol{\theta}_s$ for all of the K multivariate time series and $\boldsymbol{\theta}_{0,s}$ such that:

$$\text{var}(\mathbf{C}_s) = \boldsymbol{\theta}_{0,s}^{-1} \quad (6-3)$$

At the system layer, estimation of the concentration matrix $\boldsymbol{\theta}_{T_k}$ is as follows:

$$\boldsymbol{\theta}_{T_k, s} = (\boldsymbol{\theta}_{0, s}^{-1} + \boldsymbol{\theta}_{k, s}^{-1})^{-1} \quad (6-4)$$

In these settings, the following likelihood is estimated:

$$\underset{\boldsymbol{\theta}, \mathbf{P}}{\operatorname{argmin}} \sum_{s=1}^S \sum_{k=1}^K \left[\overbrace{\|\lambda * \boldsymbol{\theta}_{T_k, s}\|_1}^{\text{sparsity}} + \sum_{T_{k,i} \in \mathbf{P}_s} \overbrace{-LL(\mathbf{T}_{k,i}, \boldsymbol{\theta}_{T_k, s})}^{\text{log likelihood}} + \beta \overbrace{f(\mathbf{T}_{k,i}, \mathbf{T}_{k,i-1}, \mathbf{P}_s, \mathbf{P}_{s-1})}^{\text{Temporal consistency}} \right] \quad (6-5)$$

with λ representing a non-negative tuning parameter guaranteeing the sparsity. A large λ value, for instance, makes the entries of the concentration matrix converging to zeros,

$\|\cdot\|_1$ is the L₁-norm,

$*$, the element-wise product is also known as the Hadamard product,

β represents a penalty cost to assure temporal consistency. A higher β value results in assigning all of the observations into one cluster.

$$LL(\mathbf{T}, \boldsymbol{\theta}_{u,s}) = -\frac{1}{2} [(\mathbf{T} - \boldsymbol{\mu}_{u,s}) \boldsymbol{\theta}_{u,s} (\mathbf{T} - \boldsymbol{\mu}_{u,s}) + \log \det(\boldsymbol{\theta}_{u,s}) + \gamma] \quad (6-6)$$

$\boldsymbol{\mu}_s$ is the empirical mean of the samples belonging to the cluster $\mathbf{P}_s$,

γ a constant,

$\det(\boldsymbol{\theta}_{u,s})$ is the determinant of the matrix $\boldsymbol{\theta}_{u,s}$.

$$\overbrace{f(\mathbf{T}_{k,i}, \mathbf{T}_{k,i-1}, \mathbf{P}_s, \mathbf{P}_{s-1})}^{\text{Temporal consistency}} = \begin{cases} 0, & \mathbf{T}_{k,i} \in \mathbf{P}_s \text{ and } \mathbf{T}_{k,i-1} \in \mathbf{P}_s \\ 0, & \mathbf{T}_{k,i} \in \mathbf{P}_s \text{ and } \mathbf{T}_{k,i-1} \in \mathbf{P}_{s-1} \\ 1, & \text{otherwise} \end{cases} \quad (6-7)$$

The function f ensures the temporal consistency of the observations by firstly making sure that the neighbourhoods are affected by the same cluster. Secondly, when two observations do not belong to the same cluster, it should be, therefore, a change point.

A slight sub-problem of the optimisation equation (6-5) has been investigated in Hallac et al. (2018) for only one multivariate time series ($S=1$). A similar resolution approach is taken here, as the optimisation problem shares the same characteristics, i.e. it is a highly non-convex, mixed discrete and continuous optimisation problem, with two groups of variables: θ and P being the sets of the precision matrix estimation and the assignment cluster set. Therefore, the optimisation problem is divided into two alternating steps:

- The continuous optimisation problem is solved using the ADMM algorithm.
- The discrete optimisation problem is solved using a modified version of the ‘assign to point cluster’ methods introduced in Hallac et al. (2018), which is the minimal optimal path cost of the Viterbi algorithm (Backurs and Tzamos, 2016).

6.4.5.2 Problem Resolution

- Continuous optimisation through the Graphical Lasso method: a closed form of the continuous optimisation for each multivariate time series T_k in (6-5) can be rewritten as follows:

$$\underset{\theta}{\text{maximise}} \frac{1}{2} \sum_{s=1}^S \left(|P_s| \left\{ \log \det \theta_{T_{k,s}} - \text{tr}(\mathbf{R}_{T_{k,s}} \theta_{T_{k,s}}) - \lambda \|\mathbf{Z}_{T_{k,s}}\|_1 \right\} \right) \quad (6-8)$$

Subject to $\theta_{T_{k,s}}$ positive and $\mathbf{Z}_{T_{k,s}} = \theta_{T_{k,s}}$,

where $|P_s|$ denotes the number of data point assigned to the cluster P_s ; $\text{tr}(\mathbf{R}\theta)$ is the trace of the matrix $\mathbf{R}\theta$; $\mathbf{R}$ the empirical covariance matrix and $\theta = \{\theta_{T_{k,s}}\}; s = 1..S$. $\mathbf{Z}$ a consensus variable (Danaher, Wang and Witten, 2014). The augmented Lagrangian multipliers scaled by Frobenius norm of the minimisation form of the problem (6-8) defined by (Boyd, 2010) is given by:

$$\begin{aligned}
L_\rho(\boldsymbol{\theta}_{T_k}, \mathbf{Z}_{T_k}, \mathbf{U}_{T_k}) &= \sum_{s=1}^S \left(|\mathbf{P}_s| \left\{ -\log \det \boldsymbol{\theta}_{T_k,s} + \text{tr}(\mathbf{S}_{T_k,s} \boldsymbol{\theta}_{T_k,s}) + \lambda \|\mathbf{Z}_{T_k,s}\|_1 \right\} \right. \\
&\quad \left. + \frac{\rho}{2} \|\boldsymbol{\theta}_{T_k,s} - \mathbf{Z}_{T_k,s} + \mathbf{U}_{T_k,s}\|_F^2 - \frac{\rho}{2} \|\mathbf{U}_{T_k,s}\|_F^2 \right) \quad (6-9)
\end{aligned}$$

where $\rho > 0$ is the ADMM penalty parameter, $\mathbf{U}_{T_k,s}$ are dual variables. $\boldsymbol{\theta}_{T_k} = \{\boldsymbol{\theta}_{T_k,s}\}$, $\mathbf{Z}_{T_k} = \{\mathbf{Z}_{T_k,s}\}$, $\mathbf{U}_{T_k} = \{\mathbf{U}_{T_k,s}\}$, $s = 1..S$, $\|\cdot\|_F^2$ is the Frobenius norm. the optimal solution to the problem (6-9) provides an estimation of the concentration matrix $\boldsymbol{\theta}_{T_k,s}$'s. ADMM for multivariate works in three mains step as follows:

At the i^{th} iteration, update the parameter $\boldsymbol{\theta}_{T_k}, \mathbf{Z}_{T_k}, \mathbf{U}_{T_k}$:

- (a) **$\boldsymbol{\theta}$ – Update:** $\boldsymbol{\theta}_{T_k}^{(i)} = \underset{\boldsymbol{\theta}}{\text{argmin}} \{L_\rho(\boldsymbol{\theta}, \mathbf{Z}_{T_k}^{(i-1)}, \mathbf{U}_{T_k}^{(i-1)})\}$
- (b) **$\mathbf{Z}$ – Update:** $\mathbf{Z}_{T_k}^{(i)} = \underset{\mathbf{Z}}{\text{argmin}} \{L_\rho(\boldsymbol{\theta}_{T_k}^{(i)}, \mathbf{Z}_{T_k}, \mathbf{U}_{T_k}^{(i-1)})\}$
- (c) **$\mathbf{U}$ – Update:** $\mathbf{U}_{T_k}^{(i)} = \mathbf{U}_{T_k}^{(i-1)} + (\boldsymbol{\theta}_{T_k}^{(i)} - \mathbf{Z}_{T_k}^{(i)})$

The general algorithm of the multivariate $\mathbf{Z}_{T_k}$ is presented below:

- (a) For each T_k $k = 1..K$
 - (b) Initialise the variables: $\boldsymbol{\theta}_{T_k,s} = \mathbf{I}$, $\mathbf{Z}_{T_k} = 0$, and $\mathbf{U}_{T_k,s} = 0$ for each $s = 1..S$
 - (c) Initialise $\rho > 0$
 - For each $i = 1..$ until convergence
 - (i) For $s = 1..S$

$\boldsymbol{\theta}$ – Update: $\boldsymbol{\theta}_{T_k,s}^{(i)} = \underset{\boldsymbol{\theta}}{\text{argmin}} \{L_\rho(\boldsymbol{\theta}, \mathbf{Z}_{T_k,s}^{(i-1)}, \mathbf{U}_{T_k,s}^{(i-1)})\}$ using eigen decomposition method proposed by Witten and Tibshirani, (2009).
 - (ii) **$\mathbf{Z}$ – Update:** $\mathbf{Z}_{T_k}^{(i)} = \underset{\mathbf{Z}}{\text{argmin}} \{L_\rho(\boldsymbol{\theta}_{T_k,s}^{(i)}, \mathbf{Z}, \mathbf{U}_{T_k,s}^{(i-1)})\}$
 - (iii) For $s = 1..S$

$\mathbf{U}$ – Update:

$$\mathbf{U}_{T_k,s}^{(i)} = \mathbf{U}_{T_k,s}^{(i-1)} + (\boldsymbol{\theta}_{T_k,s}^{(i)} - \mathbf{Z}_{T_k,s}^{(i)})$$

The graphical lasso method presented above provides a concentration matrix for each multivariate segment. A concentration matrix of the system at the cluster $\mathbf{P}_s$

is matrix Σ_s where the diagonal represents components layer concentration matrices.

$$\Sigma_s = \begin{pmatrix} \boldsymbol{\theta}_{0,s} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \boldsymbol{\theta}_{k,s} \end{pmatrix} \quad (6-10)$$

So far, the estimation of the parameters concerns only $\boldsymbol{\theta}_{T_k}$'s. A natural way of $\boldsymbol{\theta}_{0,s}$ is computing the $\boldsymbol{\theta}_{0,s}^{-1}$'s representing the variance through a regression-based algorithm, by using all pairs of variables of the observations. Xie, Liu and Valdar (2016) showed that its estimation for the observation belonging to the cluster $\mathbf{P}_s$ can be done using:

$$\boldsymbol{\theta}_{0,s}^{-1} = \frac{1}{K(K-1)N} \sum_{m \neq l=1}^{|P_s|} \sum_{i=1}^n \mathbf{T}_{k,i} \cdot \mathbf{T}_{k,i}^T \quad (6-11)$$

Therefore the $\boldsymbol{\theta}_{k,s}$ can be deduced from the equation (4):

$$\boldsymbol{\theta}_{k,s} = (\boldsymbol{\theta}_{0,s}^{-1} - \boldsymbol{\theta}_{T_{k,s}}^{-1})^{-1} \quad (6-12)$$

To guarantee the positiveness of $\boldsymbol{\theta}_{k,s}$'s, semidefinite positiveness projection (Xu and Shao, 2012) can be used as follow:

$$\boldsymbol{\theta}_{k,s} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left\| \boldsymbol{\theta} - \boldsymbol{\theta}_{k,s} \right\|_{\infty} \quad k = 0..K, s = 0..S \quad (6-13)$$

Where $\|\cdot\|_{\infty}$ is the infinite norm.

After that each $\boldsymbol{\theta}_{k,s}$'s values are obtained, the cluster assignment algorithm can be used for the component layer clustering.

- Discrete optimisation: At each iteration of the concentration matrix estimation, clustering each $\mathbf{T}_{k,i}$, which is the i^{th} row of the observation $\mathbf{T}_k$ can be done with respect to the temporal consistency of the observations. With $\mathbf{P} = \{\mathbf{P}_s\}$ $s=1..S$, the goal here is to affect each $\mathbf{T}_{k,i}$ to only one $\mathbf{P}_s$ using the log-likelihood computed at this iteration, the function f and the penalty β to guarantee the neighbourhood assignment to the same cluster. Given the

concentration matrix estimated at one specific step of the multiple graphical Lasso algorithm, the log-likelihood LL function can be computed. Therefore, an oriented graph can be built where each node represents a symbolic representation of $-LL(\mathbf{T}_{k,i}, \boldsymbol{\theta}_{\mathbf{T}_{k,s}})$, the minimum likelihood obtained so far, and the edges are the penalty computed using $\beta f(\cdot)$. An illustration of the graph can be seen Figure 6-4.

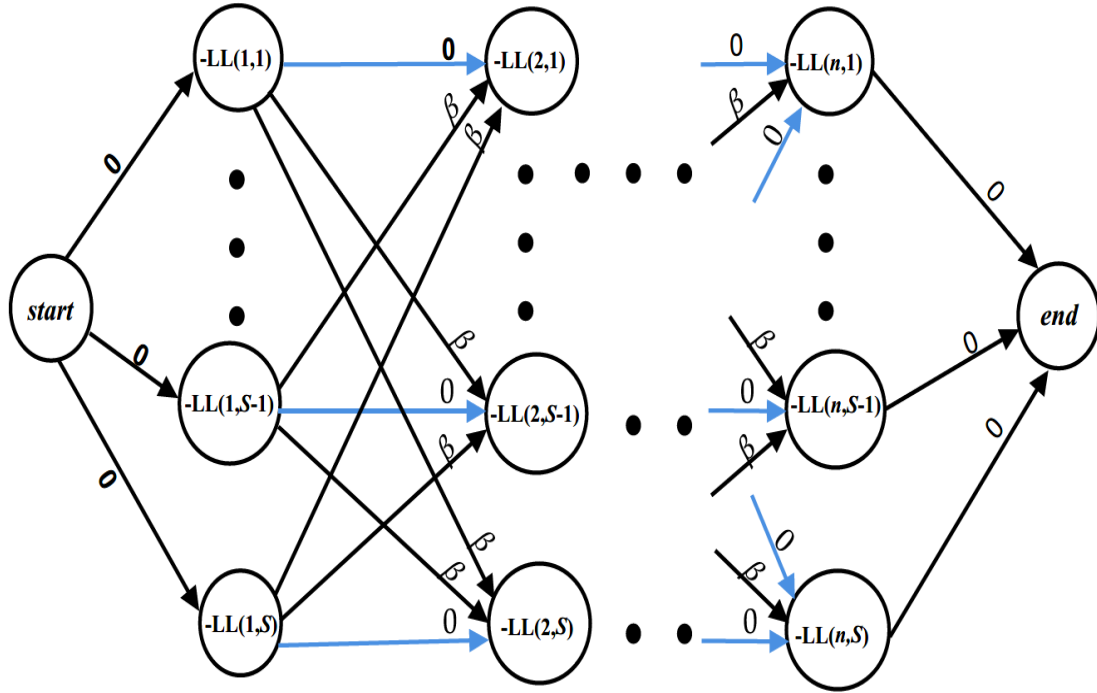


Figure 6-4 Clustering graph obtained.

To solve the clustering assignment problem, one proposes finding the minimum Viterbi path cost (Hallac et al., 2018). An adapted version to handle each case is presented in Table 6-3.

Table 6-3 Minimal path cost algorithm used.

Algorithm 1 Minimal path cost search algorithm	
Inputs: β , the penalty constant, $\beta > 0$ $-LL(i, s)$, the log likelihood of the cluster computed	
Intermediates: $Pcost$, the previous costs vector, $Ppath$, the previous path vectors, $Ccost$, the current cost vectors, $Cpath$, the current path vectors.	
1:	procedure
2:	Set $Pcost \leftarrow \{0, \dots, 0\}$, a vector of S entries 0 values,
3:	Set $Ppath \leftarrow \{\{\}, \dots \{\}\}$, a vector of S empty vectors,
4:	Set $Ccost \leftarrow \{0, \dots, 0\}$, a vector of S entries 0 values,
5:	Set $Cpath \leftarrow \{\{\}, \dots \{\}\}$, a vector of S empty vectors.
6:	for $i \leftarrow 1$ to n do
7:	Select $index$ such that $index = \min_t(Pcost[t])$
8:	for $s \leftarrow 1$ to S do
9:	Set $tmpCost \leftarrow -LL(i, s)$
10:	if $Pcost[index] + \beta > Pcost[s]$ then
11:	$index \leftarrow s$
12:	else
13:	Set $tmpCost \leftarrow tmpCost + \beta$
14:	end if
15:	$Ccost[s] \leftarrow Pcost[index] + tmpCost$
16:	$Cpath[s] \leftarrow Ppath[index] \cup s$
17:	end for
18:	$Pcost \leftarrow Ccost$
19:	$Ppath \leftarrow Cpath$
20:	end for
21:	Select $index$ such that $index = \min_t(Pcost[t])$
22:	Set $path \leftarrow Cpath[index]$
23:	return $Pcost[index], path$
24:	end procedure

Notice that the same algorithm can be used to cluster each T_k by using the $\theta_{T_k, s}'s$, X_k and $\theta_{k, s}$.

6.5 Experiments

In the experiments described, the aim is to annotate the ensemble of the group of local structures X_k 's to seek the unusual behaviour of the systems. The segmentation and annotation have therefore been carried out after removing the $\theta_{0,s}$, which is the shared structure, and therefore is not useful for abnormality detections. The datasets used exhibit different patterns and characteristics such as correlation within observations.

6.5.1 Algorithmic Parameter Settings

For effective detection, different parameter settings are applied to each category of the dataset. Table 6-4 shows the parameter values set for the test. It comprises:

- The number of multivariate used in each category (K)
- The number of observations in each multivariate time series (n)
- The number of extracted features (p)
- The number of annotations (S)
- The temporal penalty (β)
- The sparsity penalty (λ)
- The ADMM regularisation parameter (ρ) applied to the model.

These parameter values are obtained through empirical testing to find the best suitable values.

Table 6-4 Parameters setting used for the model.

Datasets	K	n	p	S	β	λ	ρ
Train_1	3	12,000,000	18	6	300	0.11	1
Train_2	3	12,000,000	18	6	300	0.11	1

6.5.2 Compared Methods and Evaluation Metric

The performance of the current approach is compared with alternative methods. These are Gaussian mixture models (GMM) and k-means, and multiple models such as a combination of Bayesian-based segmentation (BOCD), followed by a

Hotelling test (T^2 - test) statistic for the clustering, and martingale-based segmentation, and a distance-based clustering. Table 6-5 summarises the parameter value obtained by empirical testing on both martingale and BOCD methods. As the martingale test method requires a strangeness function, root mean squared (RMS) is used for all of the datasets in this experiment. The threshold λ_b for detecting a change, which is a characteristic of a new segment, has been empirically evaluated. For BOCD, prior Gaussian distribution assumption is made. Hazard parameters used alongside with the Gaussian distribution for sufficient statistics computation has also been tested empirically. The number of clusters for the methods has been set to six for the Engine_1 dataset and six for the Engine_2 dataset, in accordance with the values of **S** in the proposed approach parameter settings in Table 6-4.

Table 6-5 Parameters settings used in the compared methods.

Dataset	Martingale	BOCD
<i>Engines_1</i>	RMS, $\lambda_b = 14$	Gaussian, hazard = 100
<i>Engines_2</i>	RMS, $\lambda_b = 14$	Gaussian, hazard = 100

Three performance indicators are involved in the assessment of the approaches: recall, precision, and the F1-score. The precision measures the percentage of annotation correctness; the recall measures the rate of annotated change to be a true one; and the F1-score is the test accuracy metric representing the correctness of all identified cases.

$$Precision(P) = \frac{Number\ of\ Correct\ annotations}{Number\ of\ annotation} \cdot 100$$

$$Recall\ (R) = \frac{Number\ of\ Correct\ annotations}{Number\ of\ true\ annotations} \cdot 100$$

$$F1 - score(F1) = 2 \frac{P \cdot R}{P + R}$$

The present context can be assimilated to a multiclass learning problem. The indicators computed are averaged values of the clusters' performance. Therefore, average precision and average recall are used to compute the F1-score.

6.6 Results and Discussion

Table 6-6 highlights the results of the experiments carried out with the datasets, as well as the models. Each colour represents a visualisation of the performance for each metric; for example, green indicates a better performance. In the BOCD based-test, the average precision, recall, and F1-score scores are higher in Engine_1 datasets as it is systematically consistently above 85%, compared to the Engine_2 dataset in which a lower accuracy can be seen (77%). One of the main reasons is that the BOCD poorly segments the Engine_2 data, leading to inefficiency in the clustering.

Table 6-6 Results of overall employed datasets.

Methods	Performance in %	Engine_1	Engine_2
BOCD + T^2 -test	<i>P</i>	89	75
	<i>R</i>	92	80
	<i>F1</i>	90	77
Martingale + T^2 -test	<i>P</i>	5	7
	<i>R</i>	4	6
	<i>F1</i>	4.5	6
K-means	<i>P</i>	37	33
	<i>R</i>	41	15
	<i>F1</i>	39	20
GMM	<i>P</i>	90	91
	<i>R</i>	92	90
	<i>F1</i>	91	91
Present model	<i>P</i>	96	98
	<i>R</i>	97	97
	<i>F1</i>	96	98

The martingale-based test shows low performance in both engine datasets. Furthermore, the segmentation in the Martingale test is inefficient in all of the used datasets. This may be linked to abrupt changes in a short time period

causing change misdetection, as change does not reach the threshold λ_b , in turn lowering λ_b value and resulting in false change detection.

Like martingale-based clustering, the k-means clustering poorly performs in all of the studied datasets. The lack of temporality and the non-robustness of the distance-based metric in k-means may be the main causes of the lower performance. The GMM method consistently shows the F1-score to be over 90% for the three datasets. The presented approach shows highest precision, recall, and F1-score in both engine datasets.

In general, Gaussian distribution assumption-based approaches, namely BOCD, GMM, and the present approach exhibit better performance with all of the datasets, in comparison to the distance-based approaches. The present model over-performs BOCD and GMM when the data do not contain similar repetitive patterns.

Main limitations of the analysis include:

- Parameter and hyperparameter tuning: more parameter and hyperparameter tuning need to be undertaken in order to better evaluate the methods. For example, in a Martingale test, finding the suitable strangeness and betting function may improve the algorithm performance. Unsuccessful attempts have been made to use strangeness functions such as incremental support vector machines and the nearest neighbours. Similarly, finding the best k-means distance function and initial cluster centres may improve the performance.
- Average performance indicators: the evaluation takes into account the average precision, recall, and F1-score. Often known as macro performance indicators, the main drawback is their inability to show disparity within the performance, which may lead to inaccurate evaluations of the methods.
- Uncertainties: although pre-processing techniques have been used to reduce uncertainties such as noise and some unknown signal characteristics, they can still bias the results obtained.

6.7 Conclusion

This chapter proposed a dependent graphical based model for annotating more than one multivariate time series data with a shared structure. The segments were annotated to highlight the difference in behaviour within the multivariate time series. The model computed concentration matrices, which are used to annotate segments by computing the Viterbi's shortest path.

The proposed approach is tested using different real-world engineering datasets. The first type of dataset represents DMU engines' condition monitoring with faulty behaviours.

The study also provided an empirical evaluation and comparison of the proposed method with other alternative methods. Two methods, namely the k-means and Gaussian mixture model, are used to compare the developed approach and a combination of temporal segmentation methods (Bayesian online change detection, martingale change detection), and a clustering method (T^2 -test) are used for the comparison. Evaluation criteria are used in order to compare the methods' averaged precision, recall, and F1-score. The evaluation showed that the proposed approach over performs the compared methods. The average F1-score showed, on average, 5% better accuracy than the Gaussian mixture model and the Bayesian online change detection-based annotation methods. The k-means and martingale method performed poorly in comparison to the other selected methods. The limitation of the proposed method is mainly the number of parameters, which may be complex to estimate.

Areas of improvement that need to be addressed comprise performance enhancement, online modelling, contextual knowledge modelling, parameter optimisation, and multiple time series alignment. Further investigations need to be undertaken in the following areas:

Estimation of the number of cluster S : the present work assumes prior knowledge of the number of clusters. The main reason is that this number depends on the evolving context. However, in many cases, this number can be

unknown. Hence, further investigations can be done to automatically estimate the number of clusters by adding a penalty each time a new cluster is used. Lavielle, (1999) suggests adding the penalty $\beta_n \underline{S}$ where $\underline{S}$ is the maximal possible clusters, with upper bound, the number of observation rows n and β_n is such that $\beta_n \rightarrow 0$.

Unequal dimension analysis: the present work assumes that all of the multivariate time series have the same length and dimension. This may not be the case, so an extension can be made to handle this. Moreover, the present analysis investigated datasets, where the number of observations is higher than the dimension of the associated univariate time series further analysis adding a penalty function.

Extensive analysis on time series with different characteristics: there is a need for more investigation on real-world datasets, including those that highlight abrupt change (the stock market, for example).

Concentration estimation model: the present model applies graphical lasso recursive form for the concentration matrix estimation. It may be useful to compare model performance using other methods such as quadratic approximation for sparse inverse covariance estimation, also known as QUIC (Hsieh et al., 2011).

7 WHEEL SLIP ACTIVITIES DIAGNOSIS

7.1 Introduction

One of the major problems encountered in railway maintenance operation is wheel slip activity. Often characterised by a lower adhesion between the track and the tyres, wheel slip can cause significant damage to the track as well as to the train wheels. Wheel slip activities are complex to study, as they involve many factors. These factors include ambient conditions (ambient temperature, air pressure, humidity, snow, wet rail), global vehicle load, track and axle tyre quality, leaves and objects present on the rail, and braking forces applied to the wheels. Two main types of wheel slip can be observed:

- The friction between the track and the vehicle tyre is so low that the tire slides freely without moving the vehicle. In this context, the slipping axle speed shows short fast acceleration over time.
- Within a specific context, such as braking or acceleration, the vehicle wheels have transient periods of sleep. The slipping axles often lock in short periods, showing short fast speed drop in axle speed profiles. An example is shown in Figure 7-1 below.

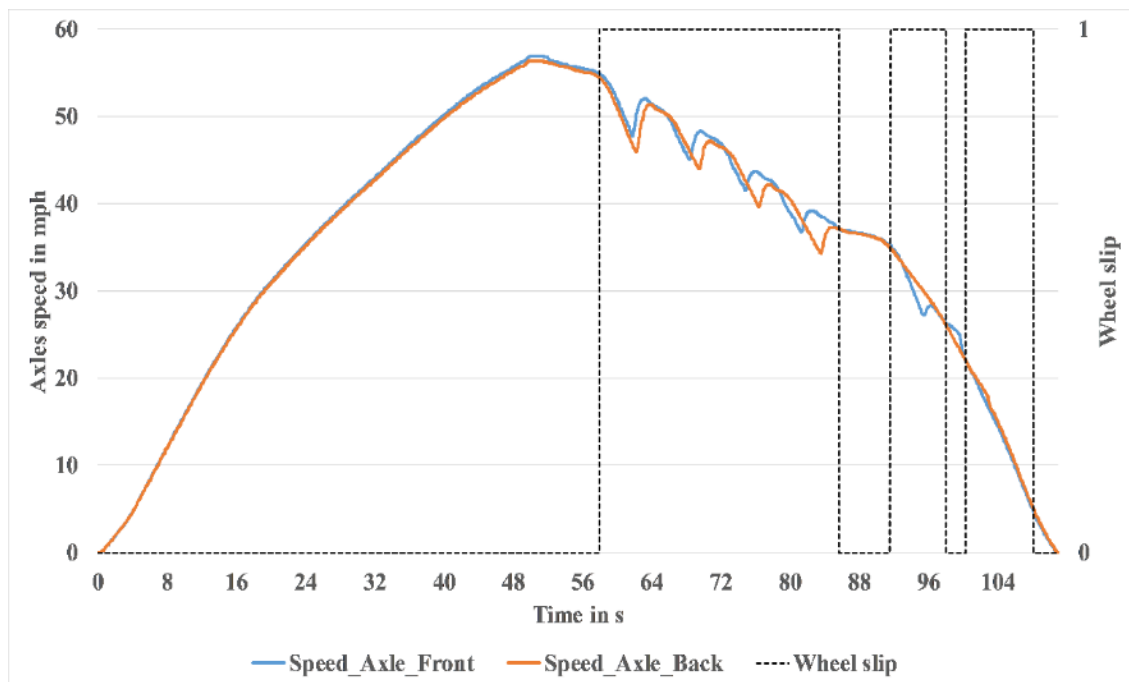


Figure 7-1 Wheel slip activity in axle speed profile.

In this work, the second category of the wheel slip is investigated in urban rail trains. Figure 7-2 shows an example of such trains with the powered axles analysed. The driver's brakes are instantaneously applied into the powered axles of three carriages of the train.

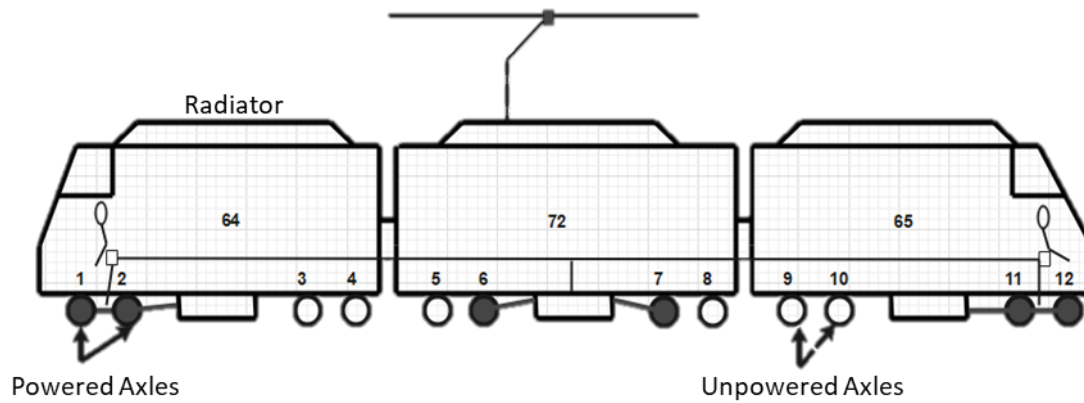


Figure 7-2 Urban DMU train powered axles investigated.

In this context, studying the behaviour of the gathered condition-monitoring data helps to diagnose wheel slip activity. The data gathered are related to old and new sanding systems, which are often adopted to increase the adhesion between the track and the wheels. Figure 7-3 shows an example of the sanding system. Compressed air pushes the sand through a tube (sand nozzle) where the sand is spread between the rail and the train wheels.

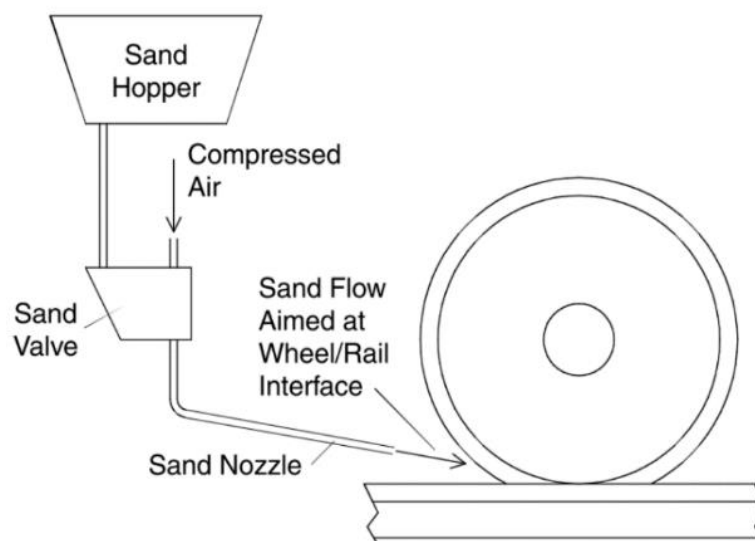


Figure 7-3 Sanding system (Olofsson and Lewis, 2006).

In the old sand system, a constant drop of sand on the track (2kg/km) is made; whilst in the new system, instead of a constant drop, the sand drop velocity is proportional to the vehicle speed. This thesis study aims to answer the underlying questions:

- Is there a more efficient way to detect wheel slip?
- Does the new sand system reduce wheel slip?

For the first question, this research first investigated univariate axle speed profiles with wavelet approaches, and then used a multivariate analysis approach with long short-term memory neural networks (LSTM).

Hypothesis testing through unequal group homogeneity testing is done in order to compare the old sand system and the new sand system.

7.2 Background

Wheel slip activity has been broadly studied through wheel-rail adhesion condition analysis. For example, substance contaminants (oil, snow, grease, water, leaves, wear, oxides, ballasts, and dust) present on the wheel-rail interfaces significantly influence the adhesion coefficients that represent longitudinal and normal force ratio (Iwnicki, 2006; Bosso et al., 2019). These adhesion coefficients need to be in line with defined safety standards. For this reason, manufacturers build assets such as braking systems, wheels, and rails that undergo various tests in different degradation conditions, in order to ensure that the equipment meets such standards (UIC, 2005; Barna, 2011). Offline analysis undertaken is often model-driven, where either vehicle dynamics such as braking distance, vehicle velocity range, wheel lock limit or slide limit, and compressed air pressure (Barna, 2011) are modelled. Additional analysis on improving adhesion through studying the effects of contaminants has been made (Lewis and Olofsson, 2009; Garnham and Davis, 2009). Limitations of the investigated model-driven methods are their implementation challenges on real-world application, as assumptions such as constant vehicle mass, speed, and friction coefficients are not realistic. Moreover, models may include many parameters, which add supplementary costs and data management challenges.

The tackled wheel slip activity problems show patterns in speed profiles. The amount of data gathered enabled the detection to be investigated through data-driven approaches.

7.3 Proposed Methodology

Figure 7-4 shows the analysis workflow undertaken. The signals selected are axle speeds and brake pressure variables. The main steps include time alignment, cleaning with a low pass filter technique, segmentation, feature extraction, models building, and evaluation. Each step is described in the following sections.

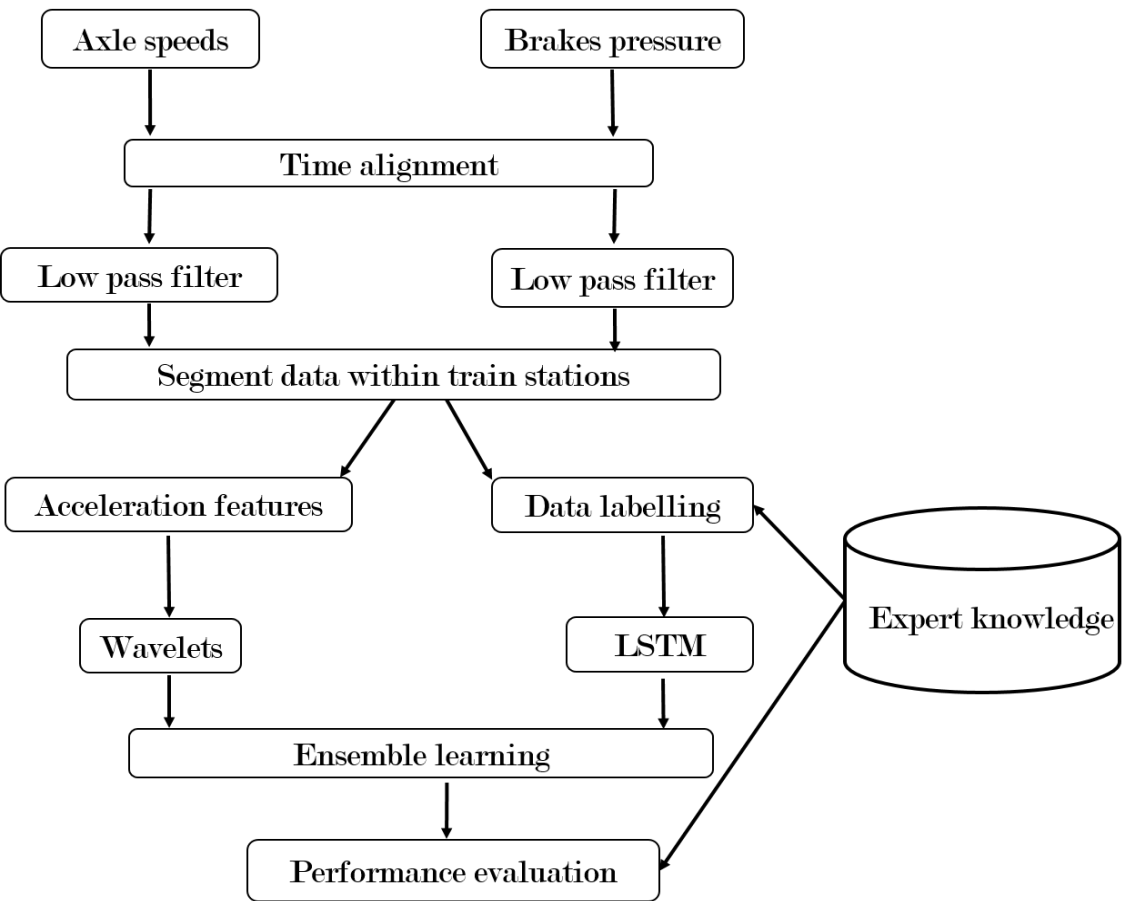


Figure 7-4 methodology workflow.

7.3.1 Data Cleaning and Time Alignment

Data gathered are collected at a frequency of 20 Hz from axles on different carriages of urban trains under different conditions and geographical positions. Figure 7-2 is an example of such trains. The axles are enumerated from 1 to 12 with axles 1, 2, 6, 7, 11, and 12 being the powered axles. Data attributes comprise of GPS location, axle speed, the mechanical and electrodynamic brake pressures applied, sanding information (sand limit, sand variations) for each axle, and the leading vehicle information. Additional computed attributes are added in the investigation carried out. These attributes are longitudinal and latitudinal acceleration, calculated from GPS coordinates; acceleration, derived from axle speeds; and brake levels (emergency brakes and regular brakes). As the data come from different sensors in different locations on the train, desynchronisation and time lag is often observed. In order to align the data, the first level of alignment is made by synchronising the GPS time recorded with the train axles sensors time. Subsequently, a second alignment is made by applying DTW (Keogh and Ratanamahatana, 2005) with L_2 -norm in the axle speeds. Noises and disturbances on the braking and speed signal are reduced using exponential moving average techniques with a window of five time steps, representing 0.25 seconds.

7.3.2 Data Segmentation, Feature Extraction and Labelling

In the segmentation step, data are broken into events representing journeys. A journey represents the starting of a train from a station to its stop at the next station. In order to achieve the segmentation, two variables are taken into account. The first variable are the axle speeds that drop to zero for a minimum period of two minutes. The second variable is the release of the doors (unlock, opening, or closing). Combining these two variables, a heuristic rule such as “if the doors open and the stopping period is longer than two minutes, then it is at a station” can be used to segment the data. To be more accurate, train station GPS coordinates are used to compute the distance between the train position and the

closest station. This distance should be lower than an empirically-set threshold of approximately 200 meters.

For the univariate speed analysis, gradient angle features are extracted from the axle speed profiles by computing an angular function $\varphi(t)$:

$$\varphi(t) = \tan^{-1} \frac{Y_{t+k} - Y_t}{X_{t+k} - X_t} \quad (7-1)$$

$\tan^{-1}$ represents the inverse tangent function, Y_t represents the speed at the time t , and X_t represents the time. Considering quantization error and noise, the k value has been set to 5 instead of 1, representing a time variation of 0.25 seconds.

In parallel, with Instrumentel and Unipart rail experts, a consequent amount of data has been labelled into normal and faulty behaviour. This enables the application of classification data-driven techniques.

7.3.3 Pattern Analysis

7.3.3.1 Univariate Wavelet Analysis

In Figure 7-1 above, the axle speed profile highlights wheel slip activities during the deceleration phase. Wheel slips represent a fast transition between normal to slip condition for a short period of time. A natural method to investigate fast transition changes are through wavelet analysis.

A basic function ψ is a wavelet, also known as a mother wavelet if it satisfies the admissibility condition, which is:

$$\int_{-\infty}^{+\infty} \frac{|\tilde{\Psi}(w)|}{|w|} dw < \infty \quad (7-2)$$

where $\tilde{\Psi}$ is the Fourier transform of ψ . It especially implies that the wavelet should integrate to zero. Many wavelets have been proposed over time, for example, Haar, Daubechies, Gaussian, and Morlet wavelets (Misiti et al., 2006). They differ from each other by their vanishing moments, which give different local

regularities. The Morlet wavelet is used in this context as it was seen to be close to human vision (Daugman, 1985). The scaling (dilatation) of the wavelet ψ with a factor α is defined as follows:

$$\psi_t(t) = \frac{1}{\alpha} \psi\left(\frac{t}{\alpha}\right) \quad (7-3)$$

the wavelet transforms C of a signal S of finite energy at the scale α at the time t is denoted by

$$C(\alpha, t) = S(t) * \psi_\alpha(t) \quad (7-4)$$

where $*$ is the convolution operation. The coefficients $C(\alpha, t)$ measure the variations of the signal S at the dilatation α . From a qualitative point of view, large $C(\alpha, t)$ value denotes irregularity of the signal S around the position t at the dilatation α . Conversely, small values of the coefficients indicate signal regularity.

7.3.3.2 Multivariate LSTM analysis

Traditional feed-forward neural networks are unable to take into consideration previous output predictions. This may be a major issue in most time series classification problems, as time series observations often have short or long term dependencies. Recurrent neural networks (Lukoševičius and Jaeger, 2009) tackle this issue in sequential learning problems with a feedback loop between the neural network layers. However, the recurrent neural networks show deficiencies known as exploding and vanishing gradient problems (Pascanu, Mikolov and Bengio, 2012) in long term dependency learning problems. LSTM was introduced to address exploding or vanishing challenges where gradients grow to infinite or quickly decay to zero (Hochreiter and Schmidhuber, 1997). Additional notions of “cell” and “gate” that allow gradient update selections were added to the classical neural networks to build LSTM networks.

One of the main justification of the application of LSTM method in wheel slip activity detection is that there is often a time lag between the cause of the wheel slip (hard braking, track condition) and the output observations. In addition, a wheel slip can cause severe damage leading to intermittent faults. Moreover,

LSTM networks are able to handle multivariate attribute with complex non-linear relationships which is in line with the present context.

7.3.4 Ensemble Learning Methods

A straightforward ensemble learning method is used for wheel slip activity recognition. Wavelet analysis is combined with LSTM methods with a voting method for efficient detection. Because wavelet analysis detects the activity when there is a fast change in shape curvatures, earlier detection is not possible. Therefore, when the LSTM detects wheel slip activity, the wavelet analysis is used as confirmation of this within the next two seconds post-detection.

7.3.5 Evaluation Methods

Four performance indicators are used to assess the classification approaches, namely the recall, the precision, the F1-score, and the area under the curve (AUC). The precision measures the percentage of annotation accuracy, whereas the recall measures the percentage of annotated change to be a true one. The F1-score is the test accuracy metric representing the accuracy of all identified cases. The AUC indicator measures the method's ability to separate observations and the AUC measurement is used in the present context since data are imbalanced.

$$Precision(P) = \frac{Number\ of\ Correct\ annotations}{Number\ of\ annotation} . 100 \quad (7-5)$$

$$Recall\ (R) = \frac{Number\ of\ Correct\ annotations}{Number\ of\ true\ annotations} . 100 \quad (7-6)$$

$$F1 - score(F1) = 2 \frac{P.R}{P + R} \quad (7-7)$$

7.4 Experiments

Experiments were carried out using the data displayed above ((7-5)(7-6)(7-7)). It consists of 12,000,000 observations representing six months of the urban train operations and 26 attributes, including the axle speed used for the wavelet analysis. Table 7-1 below presents parameters empirically set for each method.

The data has been divided into training, testing, and validation sets; 60% of the data has been used for training, 20% for testing, and 20% for the validation sets. The LSTM network is implemented using Keras and Tensorflow libraries.

Table 7-1 Methods parameter settings.

	Parameters
Data	Number of observations: 12 000 000
	Training: 7 200 000
	Testing: 2 400 000
	Validation: 2 400 000
	Number of attributes: 26
Wavelet	Mother wavelet: Morlet
	Input signals: axle speeds
LSTM	Hidden neurons per layer: 50, 90,60
	Activation: reLu, sigmoid, sigmoid
	Solver: Madam
	Adam regularisation: 0.05

7.5 Results and Discussion

In the wavelet analysis, an example of the computed $\varphi(t)$ is shown in Figure 7-5. Fast transition to slip mode shows a sharp increase and decrease of the φ values. The scalogram of the convolution of φ with the Morlet wavelet is shown in Figure 7-6. The normalised $C(\alpha, t)$ values, the scale, and the time are highlighted. It can be seen that high normalised $C(\alpha, t)$ values in yellow show irregularities in the speed signal, which represent wheel slip activities.

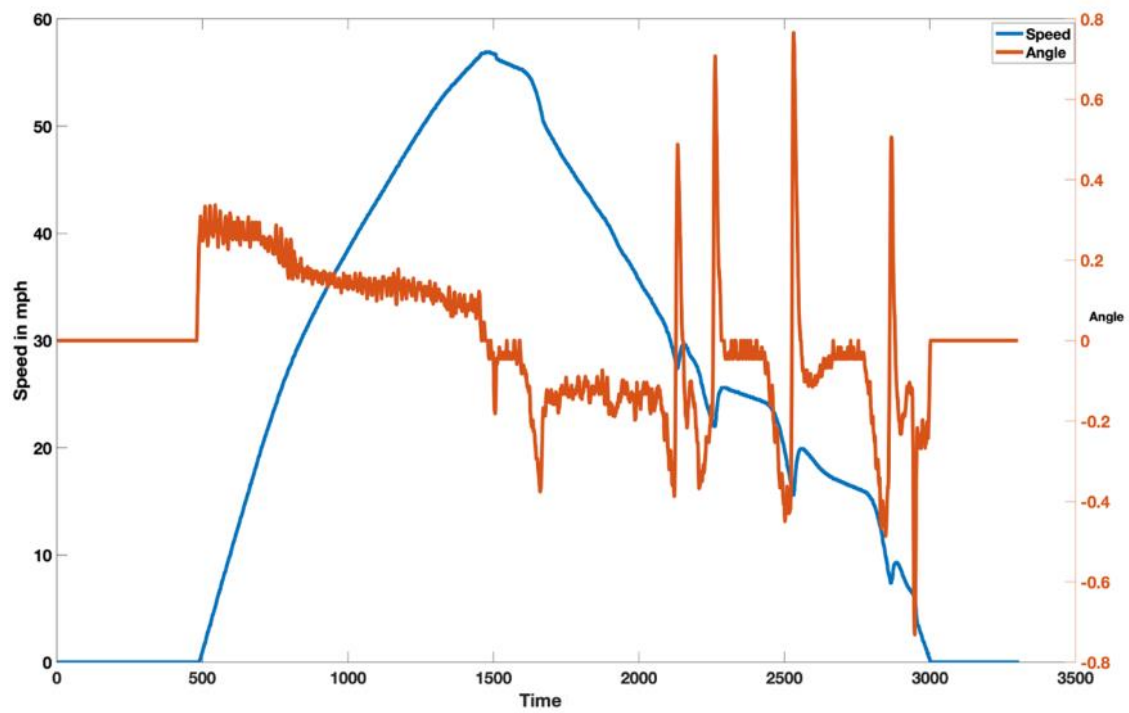


Figure 7-5 Angle value computed from the axles speed.

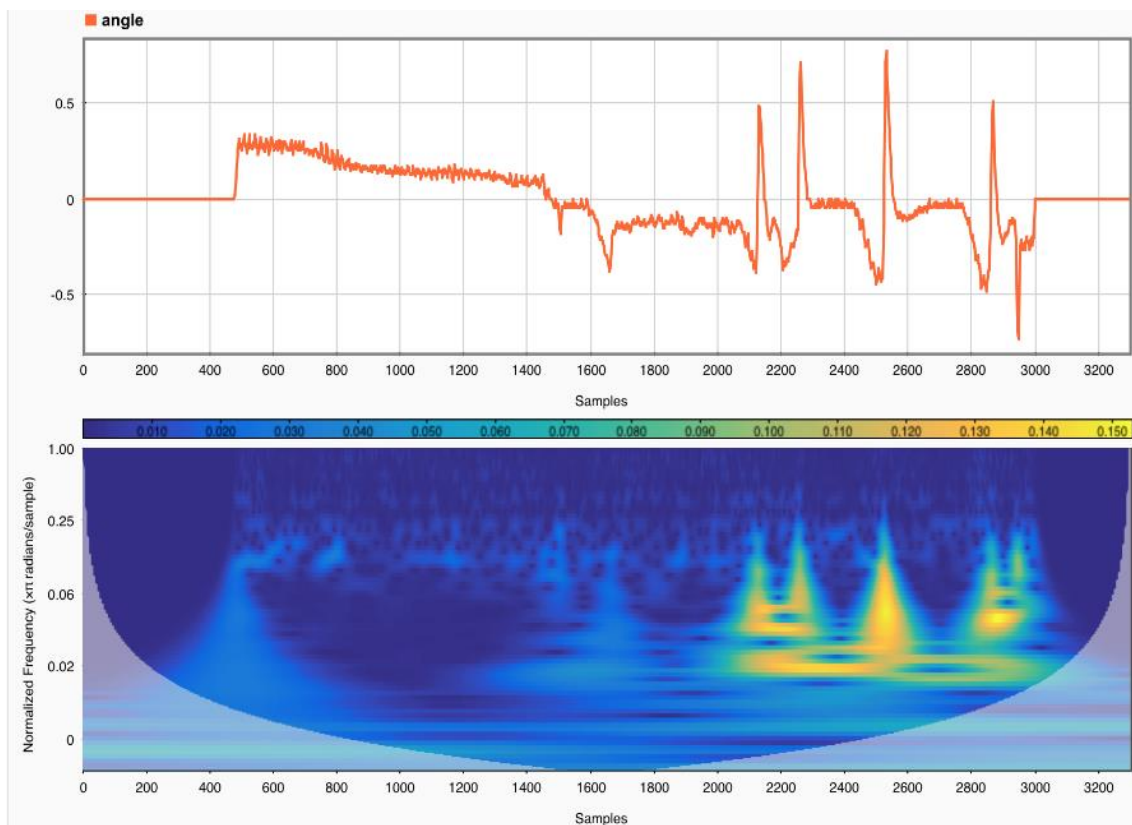


Figure 7-6 Scalogram of the wavelet.

Wavelet analysis shows efficient detection when wheel slip activity occurs during the speed deceleration phase. However, when it occurs during the acceleration phase, the speed profile may not be significantly affected. Figure 7-7 shows an example of this case. The LSTM method covers this deficiency through the learning of the multivariate attributes. An interesting observation of the LSTM results shows that the method is able to detect wheel slip condition several seconds before the slipping event starts. This can allow, for example, automatic activation controllers to soft release the brakes or inject more sand to reduce the slip activity.

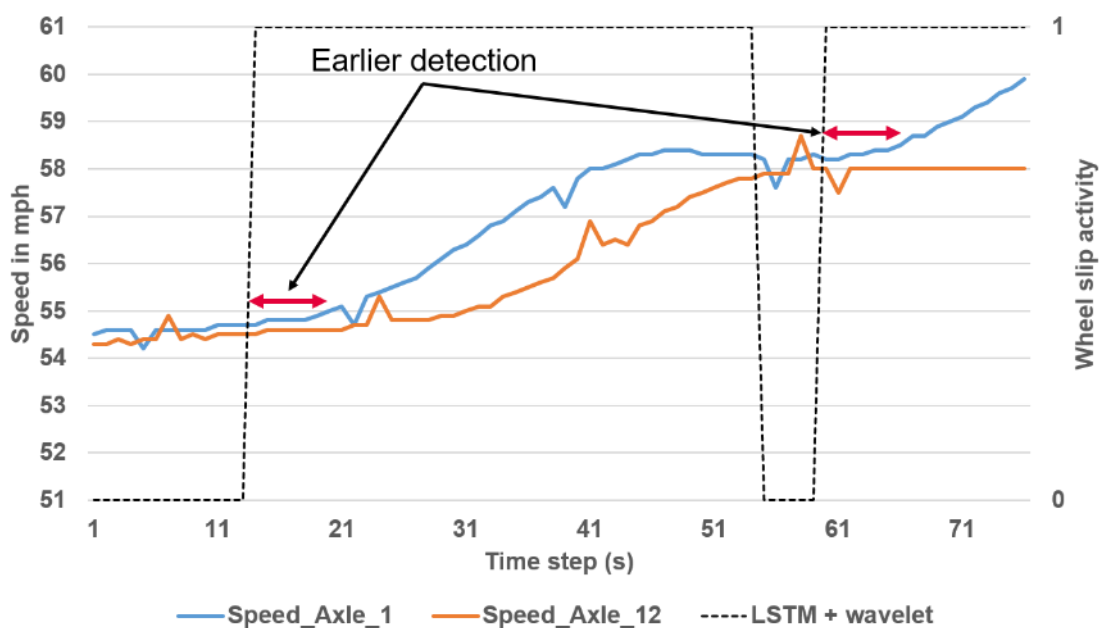


Figure 7-7 Wheel slip activity during an acceleration phase.

Table 7-2 below shows investigated methods of performance during the validation phase. The colour represents a visualisation of the performance for each metric, with green indicating a better performance. The Random Forest method shows the lowest performance in comparison to the remaining classifiers. It is followed by the Decision Tree and k-NN methods, which show similar F1 accuracy. Whilst the F1 score is higher (94% and 92% respectively), the AUC shows a low-performance score (50%). This is linked to the fact that the data are imbalanced and therefore the classifiers are unable to separate the classes very

well. Naïve Bayes, Logistic regression, and SVM methods show the AUC's best performance score in comparison to the random forest, k-NN, or decision tree methods. The highest performance score is shown by the proposed methods. LSTM+wavelets outperform all of the classifiers.

Table 7-2 Validation results of the classifications methods.

Model	AUC	F1	Precision	Recall
Decision tree	50	94	92	96
K-NN	50	93	92	95
Naive Bayes	86	86	87	85
Logistic regression	78	91	94	89
SVM	68	82	91	76
LSTM + Wavelet	98	98	99	97
Random forest	40	60	49	79

Limitations of the analysis are principally classifier parameter evaluation and the fact that the study relies mostly on the data. Numeric parameters have been tested empirically using a genetic algorithm to find the best parameters. However, as the genetic algorithm does not guarantee global maxima, the obtained parameters can be biased, leading to inefficient conclusions. Hyperparameters have been set throughout trials in order to distinguish the best performance. Moreover, although experts have annotated data, human-based annotations may have biases that influence the learning and the classification results. Additionally, the experiments carried out do not take into consideration information such as track and wheel condition, vehicle mass and traction force, and wheel conditions. The lack of such inputs limits the performance of the undertaken analysis.

7.6 Sanding system comparison

The developed methodology is used to detect wheel slip activities in two trains using the old and new types of the sand system. Recall, in the old sand system, is a constant drop of sand in the track (2kg/km) that allows an increase of the adherence between the track and the tyre. In the new system, instead of a constant drop, the sand drop velocity is proportional to the vehicle speed (maximum sand drops being 5kg/km at a vehicle speed of 60 miles/hour). Although wheel slip activities depend on the external non-monitored signals such

as the vehicle tyre, track quality, and the vehicle loads, the improvement may result in less wheel slip or short slip durations. The two trains run on the same track with an interval of 30 minutes. Figure 7-8 presents each train vehicle's slip number and the average slip length during a running three months period. It can be seen that the number of slips and the average slip length decreased in the new sand system; there was more than a 50% decrease in the number of slips and a 25% decrease in the slip lengths. These results therefore show that the new sanding system improves the wheel slip condition.

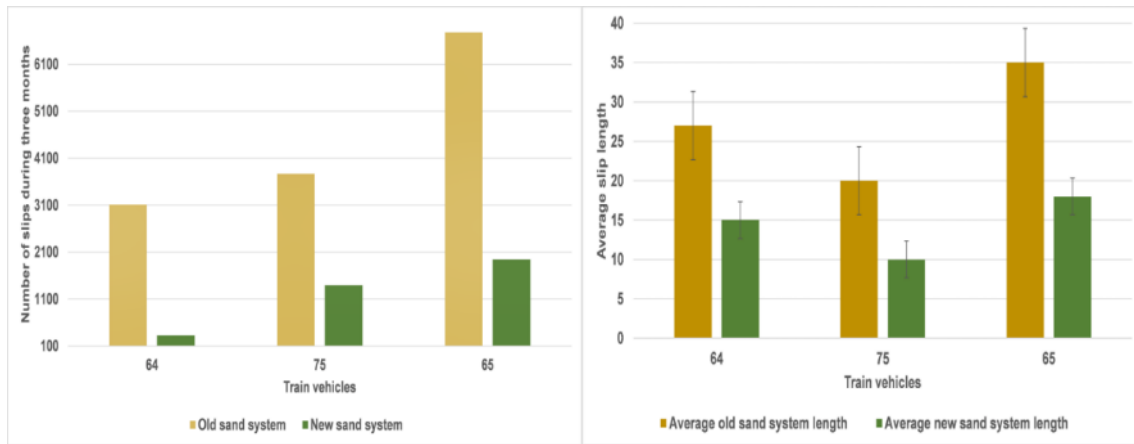


Figure 7-8 Old and new sand system slip number and length.

7.7 Conclusion

This chapter proposed an ensemble learning method for wheel slip activity recognition on urban trains. The proposed method was used to detect wheel slip activity with two different sand systems. Comparison of the sanding systems, using average wheel slip length and the number of slips, showed that the developed new sand system improved the slipping condition.

An ensemble data-driven method was proposed. This is a combination of LSTM networks and wavelet methods to classify the data gathered from real trains and annotated with TOC experts. Performance of the proposed methods was compared with alternative classifiers, namely Decision tree, K-NN, Naïve Bayes, Logistic Regression, SVM, and Random Forest. The proposed method

outperforms alternative classifiers with a 98% AUC score and 98% for the F1-score.

Limitations of the proposed method are the partial reliance on expert annotation and reliance on method parameters that have been tested empirically. Moreover, in this analysis, the absence of certain conditions such as weather information, track condition, traction load, and wheel condition limits analysis performance.

Areas for improvement include tackling the challenge mentioned and incorporating external data for more efficient diagnosis. Long-term work includes designing a distributed online diagnosis system with fault isolation, allowing similar train systems to share newly discovered fault characteristics.

8 DISCUSSION

8.1 Introduction

This chapter discusses the findings, limitations, and uncertainties of the investigation carried out. The study aim was to investigate the applicability of data mining methods to diagnose time series data-based faults to enable service support. To that end, three research objectives were set. Each of the research objectives are discussed in the following sections.

8.2 Tool Selection

The first research objective aimed to *investigate the literature to select appropriate big time series data tools to store and manage the vast amount of data generated from the train assets*. The main goal was to review the literature in order to select appropriate tools to build a streaming platform architecture.

The platform architecture that was built is shown in Figure 8-1. . It was found that given the temporal nature and the characteristics of the data gathered from the sensors and the different types of analysis available to be undertaken, a platform architecture built mainly upon Kafka, Apache Spark, and HDFS is suitable. The built architecture enables offline and online streaming analysis in different layers and is easily extensible to other temporal data management and analysis such as fault prognosis. Moreover, the selected software enables external tools, programming languages, and data mining library incorporation, which can allow fast customisation and development of the data mining techniques. Although broad data analysis platforms exist (Chawda and Thakur, 2016), the designed architecture is simply easier to implement in classical computers with low costs.

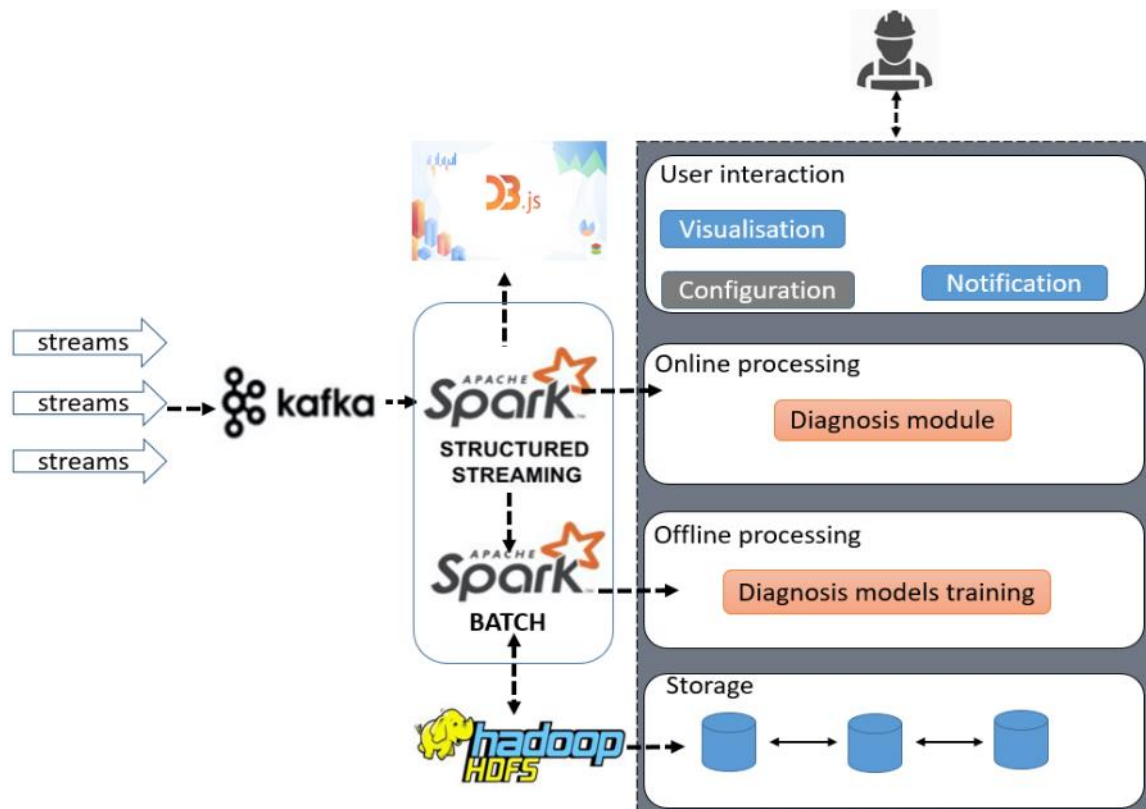


Figure 8-1 platform architecture.

The investigation showed that although fault diagnosis methods have grown over the years, the practical application of the diagnosis techniques and data fusion architectures, especially to temporal data, are limited due to the complex nature of the real-world problem. Therefore, in order to address this gap, a time series fusion architecture was proposed with application to real case studies presented in the following research objectives.

8.3 Time Series Architecture

The second research objective goal was to *develop an architecture for multivariate time series data analysis*. In this research objective, the aim was to provide an architectural framework for temporal data analysis to enable efficient decision making.

Figure 8-2 presents the proposed architecture. It can be seen as an abstract level of the implemented diagnosis module shown in Figure 8-1. It was found that there are essential steps, which include:

- Sensor level, which seeks to define sensor type, resolution, and distribution.
- Time series level, in which time series characteristics and properties are defined.
- Association level where observations are eventually aligned, pre-processed, and segmented, and key features are extracted with the support of the contextual knowledge of the system and data-mining techniques.
- Feature/pattern fusion level, in which, by means of system knowledge and knowledge discovery techniques, patterns or features extracted are combined for diagnosis, prognosis, or recommendation.
- Assessment level where the method built is validated through a benchmarking procedure or using relevant performance indicators.
- Decision level, in which important decisions are taken. Decisions can be either a new iteration to improve the performance of the model or administrations.
- Actions level, which is the implementation of the decisions.

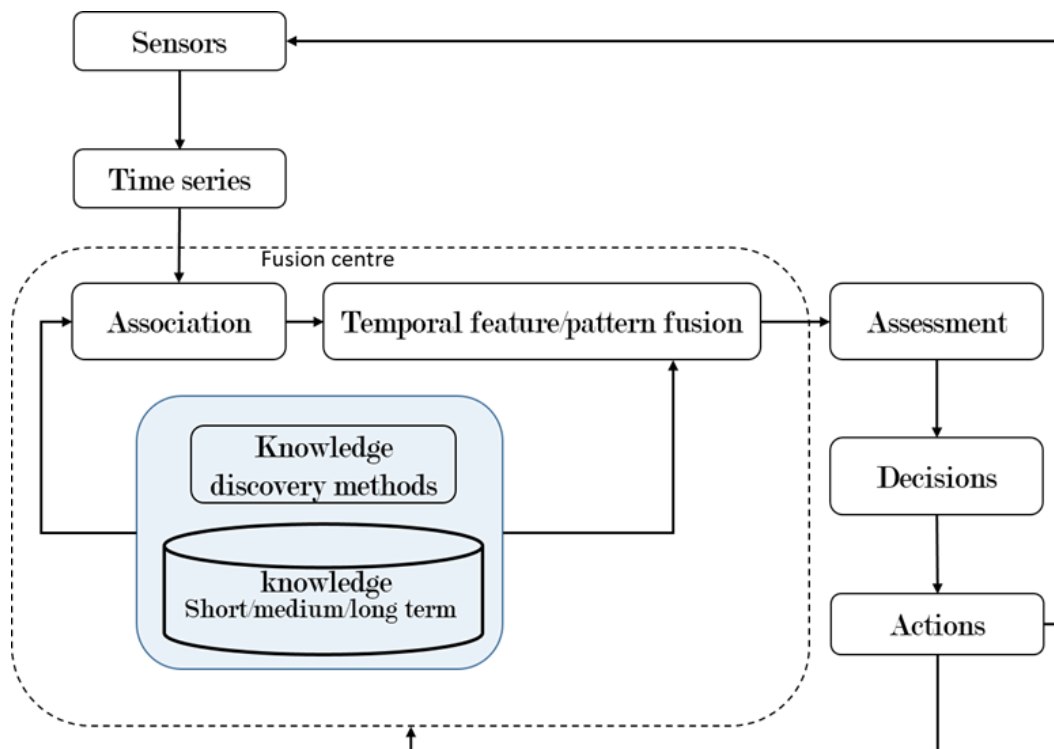


Figure 8-2 Proposed time series architecture.

Through investigations carried out in chapters 5, 6, and 7, using the example of the architecture, it was shown that the proposed time series fusion architecture could be employed in a diverse time series case scenario, which includes different contexts and domains. Existing widely-used architectures such as JDL or Boyd are too generic, in terms of steps. They are suitable in the military domain but present implementation difficulties in different fields, especially within a time series context. Time series often requires a different level of fusion, depending on the signal channels importance and relation. Furthermore, an online analysis may require prior offline training or parameter estimations. In addition, when the studied system evolves or is not well known, iterative analysis is critical to better perform the fusion with insights and knowledge obtained so far. The proposed architecture allows handling these contexts, which are appropriate for real-world time series processing.

8.4 Case Studies

The third research objective main goal was to design and investigate the applicability of methods to detect unusual changes in the train-based large time series data in order to enable maintenance operation service support. It aimed to provide suitable methodologies and methods for diagnosing potential faults on DMU engines, doors, and wheels.

The case studies encompass train engine data with multiple channel time series under different speeds and loads, train door data highlighting start-stop motion profiles, and wheel slip activity data showing fast slips transition under different driving conditions.

8.4.1 Engine Diagnosis

In the engine diagnosis case study, the experiments carried out investigate DMU engine fault diagnosis throughout a multiple channel time series clustering. The model showed high accuracy in comparison to alternative clustering methods. This model's advantage is that it is not only applicable to engine diagnosis but to any multichannel time series with the same number of dimensions and

observations. Moreover, it can be used to build a system structure with the annotation.

A Gaussian log-likelihood-based concentration matrix model is built under i.i.d assumptions. Moreover, the model also assumes that the data channels share the same precision matrix characteristics. In a long-time monitoring system, such assumptions may suffer from limitations. In fact, partial correlations of the feature often exist. In Figure 6-2 below, where the charge air pressure and the engine speed is plotted over time, within the interval [200,300] for instance, there is a linear correlation of the acceleration gradient features between the two attributes. This can create convergence issues in the precision matrix estimations as two columns with the same values in the covariance matrix make the matrix singular.

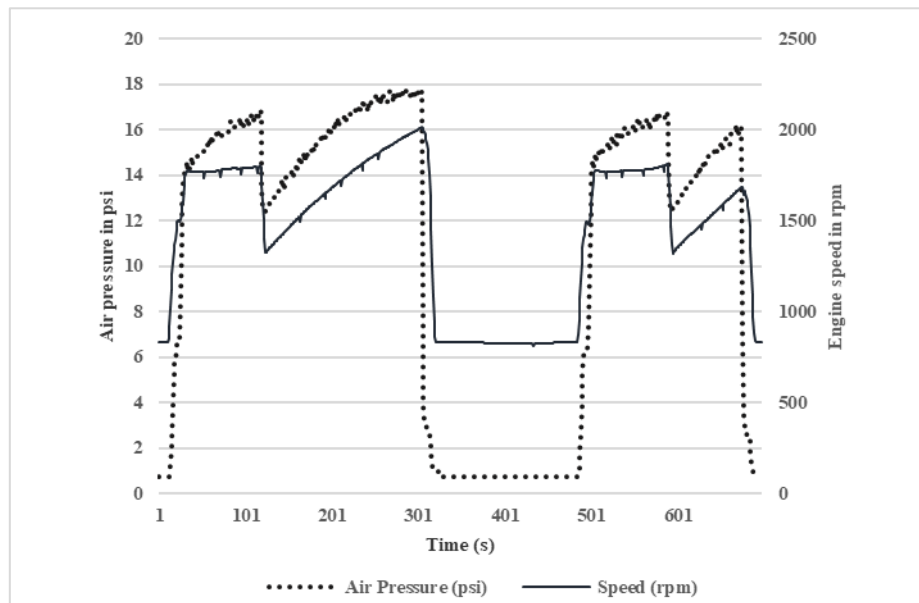


Figure 8-3 Engine frequency and associated charge air pressure over time.

Another limitation of the analysis undertaken is parameter estimations. Parameters have been empirically tested and set. Whilst these parameters provide efficient analysis results, they may not be optimal as the trial search strategy may have converged to local maxima.

In general, the model mostly relies on the data and therefore, uncertainties and analysis limitations of the previous stage (described above) may influence model parameters and bias the results.

8.4.2 Door Diagnosis

Door systems highlight two main motion profiles: opening and closing, also respectively known as extension and retraction. The analysis sought to detect and classify the degradation rate of the doors during extension and retraction phases. The k-means clustering method is first used in the condition indicators extracted through PCA analysis. PCA methods can result in important information loss that may be characteristic of faulty behaviours. Secondly, when faulty behaviour is detected and validated by the expert, modelling of its degradation behaviour is made to estimate the different degree of the fault stage for fault rate classification. The K-NN method reveals a high-performance score.

The advantage of the model is that it does not require door type specification to operate. It can also be used for second-order systems fault diagnosis such as actuators which highlight extension and retraction patterns.

The main drawback of such artificial degradation behaviour is that it may not reflect the real case scenario as there is no knowledge about the defective component. Additionally, there can be other component degradation such as the batteries influencing the current features used, which may entirely modify the data behaviour.

Like the engines, a limitation of the analysis includes the K-NN model issues, the compared methods parameter estimations, which have been done empirically, leading to analysis bias.

8.4.3 Wheel Slip Activity Diagnosis

In the wheel slip diagnosis, experiments carried out investigate wheel slip activity using DMU train data. A combination of the wavelets method and LSTM-based network is used to detect the wheel slip activities. The wavelet approach analysed the axle speeds seeking to fast slipping transitions of the wheels; whereas in LSTM networks, learning from some of the labelled multivariate time series observations is made before classification.

In wavelet analysis, the model is often unable to recognise an extended sleeping transition or sleeping during the acceleration phase as axle speed profiles do not necessarily highlight quickt-change profiles. Moreover, the hard braking condition creates fast speed changes leading to false detection with the wavelet methods.

The LSTM method solves this aforementioned wavelet challenges. However, it relies on human-data annotation, which may be inaccurate. Moreover, it is difficult to tell if all the wheel activity types are present in the data in order to diagnose all of the future sleeping conditions. Another uncertainty is the inability to comparatively associate hidden model states with physical wheel slip conditions to the HMM models.

In addition to the described pattern classification and clustering challenges, pre-processing steps also carry limitations and uncertainties. They are described in the next section.

In addition to the described pattern classification and clustering challenges, pre-processing steps also carries limitation and uncertainties. They are described in the next section.

8.5 Limitation of the pre-processing analysis: Assumptions, Limitations and Uncertainties

8.5.1 Data

Data analysed in the three case studies carry uncertainties. These uncertainties encompass missing data; imprecise sensor measurements; sampling frequency, which can lead to the loss of important system dynamics; the noise present in the measurement; data coherence; and ambiguous data, i.e., data exhibiting unexplained patterns. Although cleaning techniques and expert knowledge helped to reduce such uncertainties, they still remain, and may affect the accuracy and the precision of the analysis.

In terms of limitations, data collected from the sensors provide only a partial picture of the studied systems. For example, the engine and wheel data provide only several output measurements. The lack of input measurements limits the application of available techniques such as model-driven. Another limitation of the data is the lack of different fault types, and the inability to assess present fault severity. Moreover, faults are infrequent, as most of the data represent the normal working condition. Furthermore, the lack of monitored equipment specifications (for example engine class, wheel specification, type of actuators used for the doors) affect the comprehension of the data.

8.5.2 Data Synchronisation

Data synchronisation aims to correct time lag observed within variables or the delay between the input and the output variables due to sensor synchronisation issues or wrongly time-indexed values (failure to synchronise with network time protocol servers, for example). The analysis carried out used mainly dynamic time warping algorithms to realign the data. Desynchronisation often occurs randomly such that the applied methods may not entirely eliminate time lags. Additionally, the l1-norm used as the distance metric of the DTW algorithm is not robust to noise, which can introduce alignment bias.

8.5.3 Pre-processing

The pre-processing techniques include moving average for noise reduction after removing extreme values representing sensor errors. In terms of data imputation, for small gap windows (i.e., missing data length is less than 10 seconds), regression techniques are used to fill the gap. Large gap window datasets are employed with the DTW method to segment the data, as the uncertainty is high and can influence results when a regression method is used. However, these methods can add bias to the results. For example, the moving average methods smooth the signal in the acceleration and deceleration phase of engine signals or in the extension and the retraction phase of doors, which can lead to model deficiency.

8.5.4 Segmentation

Segmentation aims to break the data into known events that can help to extract sensitive features. For example, in engine analysis, the Bayesian online change detection approach is used to segment the data as investigations show that the method is efficient, in comparison to martingale approaches. Segmentation methods often suffer from end point issues as the end points often belong to the other segments. The inaccuracy of the segment end points in some context can lead to imprecise feature extraction, which can affect the model efficiencies.

8.5.5 Features

Features extracted have been seen to be sensitive to deviation from normalities. In both engine and door analysis, features such as gradients and steady-state variables have been used. Although these features are sensitive to fault conditions, they are also often sensitive to external conditions. For example, driver behaviour affects engine analysis and passenger actions, such as holding onto doors and blocking doors with their luggage, affect door analysis.

8.6 Conclusion

This chapter discussed the limitations and the uncertainties of the investigations carried out in this research. Limitation examples include:

- The data gathered, in general, contains uncertainties such as noise, sensor measurement inaccuracy, missing data rows and column values, redundant data, and explained behaviours.
- Data alignment techniques may create additional uncertainty such as time lag and the system dynamics in the output observations potentially being so specific that correlation or time alignment may not be accurate.
- The segmentation techniques used often suffer from endpoint problems. In fact, endpoints often belong to other segments due to segmentation inaccuracies. This can create a bias of the features extracted, resulting in analysis deficiencies.

- Features extracted in some case studies (doors and engines) analysis are revealed to be sensitive to either driver or passenger actions.

Although methodologies undertaken for each case scenario present some limitations and uncertainties and therefore need to be improved, the performances exhibited are promising and they can be used in broader domains.

9 CONCLUSION

9.1 Research Aim and Objectives

As mentioned in section 1.2, the research aim was to design and investigate the applicability of methods to detect unusual changes in the train-based large time series data in order to enable maintenance operation service support. To achieve the research aim, the following research objectives were set:

RO1: Investigate the literature to select appropriate big time series data tools to store and manage the vast amount of data generated from the train assets. It aims to review the literature and select appropriate tools in order to build platform architecture.

This objective has been achieved through the literature review undertaken in Chapter 2, which reviews main data analytic tools. In order to manage the massive gathered data, a system has been built using Apache Spark, Kafka, and HDFS for joint storage. The selection of these tools are mainly due to their maturity; their wide use, offering broad online support; the possibility of storing a wide range of data structures; the ability to process offline as well as streaming the data online; and finally their low costs. Moreover, they offer more flexibility by enabling the service exchange with external tools.

RO2: Develop an architecture for multivariate time series data analysis. In this research objective, the goal is to provide an architectural framework for temporal data analysis enabling efficient decision making.

This objective has been covered through chapter 4, where a novel time series architecture was introduced. The proposed architecture allows iterative temporal data fusion. Key steps include:

- Sensor identifications. This stage enables understanding the type of time series sensors used, the resolution and distribution, and the computational capabilities.

- Temporal properties identification. At this stage, the time series nature and properties are defined.
- Signal association. System knowledge and data-mining techniques are used in this step to align, pre-process, segment, and extract relevant features.
- Patterns or temporal features fusion. With the support of contextual knowledge of the system, the goal of this step is to devise techniques for the temporal pattern/ feature recognition.
- Assessment. It aims to evaluate and validate the fusion centre throughout relevant performance indicators and benchmarking.
- Decisions. The optimal decisions are taken considering the evaluation in the previous stage.
- Actions. Actions include either a new iteration to improve the performance or resource tasking.

RO3: Design, implement, and validate methodology using developed architecture in three train asset case studies that examine fault diagnosis of train doors, train engines, and train wheels. It aims to provide suitable methodologies and methods for potential fault diagnosis on multiple unit diesel train engines, doors, and wheels. The architecture developed in **RO2** will serve as a basis for the methodologies used for each case study.

This objective has been covered through chapters 5, 6, and 7, where the investigations allowed:

- Understanding key characteristics in the analysed datasets, which represent systems' physical dynamics and understanding the degradation behaviours.
- Determining normal and faulty conditions. Most of the faults found were actuator faults and sensor reading issues. This stage has been investigated with Unipart Rail and Instrumentel experts, who are the data providers.
- Proposal of a methodology for each case scenario to diagnose the faults. As data characteristics and fault behaviour are different for each case study, a different methodology is used to tackle each diagnosis case study.

- Applying existing methods as well as devising new methods for detecting the normal conditions and faults seen.

In the doors case study, the analysed data represent sequences of extension and retraction characteristics with discontinuities over time. The aim was to find normal and abnormal behaviours and then implement methods capable of diagnosing door degradation stages. The challenges were to find suitable techniques to synchronise, clean, extract features sensitive to faults, and then develop an appropriate classification method to detect degradation stages automatically. The investigation carried out showed the presence of actuator faults. Therefore, the degradation stages were estimated using ratios between the averaged normal door condition and the averaged final fault stage data. This artificial estimation was used along with the existing data for fault classification. The k-NN algorithm was retained as its analysis results showed the best performance in the data.

For the engine case study, a different characteristic of the time series is observed. The data are three-channel time series where each channel is a long run multivariate data expressing a diesel engine speed and load profiles. The engines are sharing the driver acceleration loads, hence the channels data share similarities. The aim was to diagnose faults present in such structure. Like the door data, the main challenges were to synchronise and align the data within each multivariate data as well as across the three channels; clean and segment the data; extract appropriate features; and devise a method to detect faults. The analysis showed that faults present in the data were mainly random engines not producing adequate power to satisfy driver demand, also known as low power events. Features sensitive to both normal and faulty conditions were extracted, and a graphical model-based method was built to capture the conditional dependency between channel attributes and then annotate the data through a clustering method. The annotation highlights similar observations with the same labels, allowing normal and faulty behaviour separation. This implemented

method showed the best performance indicator in comparison to traditional clustering methods such as k-means or GMM.

The last case study investigated related wheel slip activity. Wheel slip activity occurs under many conditions such as track condition (defective track parts, or leaves, snow, or grease are on the track); train vehicle axles and tyre condition; and driver braking activities. Data gathered encompass, for example, axle speeds, the brake pressures applied to the axles, and GPS data. Data contain fast change transitions from the sleep/slip condition to normality. The main goal was to develop efficient methods for wheel slip detection. The main issues are analogous to two previous case studies. Data have been aligned, cleaned, and, with the help of Unipart and Instrumentel experts, labelled. Then a hybrid method was designed by combining wavelet analysis in the speed and LSTM neural networks. The designed classification method showed the best performance in comparison to the other methods.

9.2 Contribution to Knowledge

This research's contribution can be stated as follows.

Study of the applicability of methods for sensitive and robust measurement of degradation in engineering systems. Three real-world time series case studies with different characteristics are analysed to seek faults. The first case study is related to multiple-unit train engines with different speeds and loads. The second case study is related to doors, which present start-stop characteristics with multiple comparable channels with discontinuities in the data. The third case study is wheel slip, which presents the fast transition from normal to slip condition in two conditions: acceleration and deceleration.

Novel aspects

- A novel methodology for door fault diagnosis is proposed. In the literature, no door fault diagnosis methods were covering a real case scenario. Instead of extracting traditional whole door profiles statistical properties as done in the literature (Lee et al., 2016), a focus is made on signal

segmentation and second-order system properties (overshoot, gradients, settlements) to better capture system behaviour prior to the fault classification stage.

- Multiple channel time series with shared structure annotations with application to DMU engines is a new development. Most of the diesel engine's fault studies focus on one unit only. In the present work, a new method was proposed to observe the variation across engines for fault diagnosis through concentration matrix estimations and clustering.
- A novel ensemble learning method for wheel slip activity diagnosis is created. The wheel slip activity diagnosis was model-driven as most of the analysis existing in the literature is experimental or simulation-based. Model-driven enables having and setting key input parameters such as external condition temperature and pressure, adhesion coefficients, and vehicle load. Moreover, the analysis undertaken is only in one axle. There was no investigation using multiple axles and exploring specifically the wavelet+LSTM method covered in the present work.
- Evaluation and comparison of Bayesian online change detection and martingale methods using different data characteristics. In the literature, there was no comparison and evaluation of these methods with different data characteristics.
- A new architecture for using methods and techniques for time series analysis. Existing data-fusion architectures are often too generic, too restrictive, or too specific for time temporal data fusion. The research proposes a flexible architecture for time series data fusion that served as the basis of the methodology in the case studies.

9.3 Future Work

9.3.1 Short-term

Short-term work includes many aspects:

- **Data and related knowledge.** More in-depth investigation can be carried out to understand unexplained behaviour seen in the data. This will help in the data-cleansing step and additional feature extractions to improve method accuracy. Furthermore, the data collected provide an incomplete picture of the monitored equipment. Adding additional sensors to capture key input measurements in the future will enable a more efficient diagnosis.
- **Temporal alignment and synchronisation.** Current literature significantly suffers from the lack of methods that are able to align time series with more than two dimensions. A path analysis could propose novel algorithms for multidimensional time series alignment.
- **Pre-processing.** Methods used to remove noise such as low pass filters often introduce local bias when a fast change occurs. Future work direction could therefore be developing new methods taking into account fast state transition. Moreover, gaps filling during the data imputation analysis often suffer from inaccuracy and also introduce additional uncertainty. Extensive work may include devising new methods that enable the use of cross-channel data for gap filling.
- **Features Extraction.** Although the extracted features are sensitive to equipment conditions, they are also often sensitive to external conditions. For example, in the case study with application to train door diagnosis, gradient features are often sensitive to passengers' actions such as holding the doors or getting their luggage stuck in the doors. Future work can include the analysis of the impact of external conditions in the data for efficient feature extraction.
- **Improvement of method performance.** In the engine case study, examples include parameter estimations for the graphical models and the

improvement of the concentration matrix estimators for global convergence instead of local convergence provided by expectation maximisation algorithms. Moreover, an in-depth investigation can be undertaken to enhance the robustness and the accuracy of the model and extend it to non-Gaussian analysis. In the case of the door's analysis, the improvement includes root-cause analysis and degradation prognosis. For the analysis of the wheels, future investigations may consist of detecting key characteristics of the wheel slip activity several seconds before in order to enable controllers' actions such as releasing the brakes.

The analysis carried out assumed one fault at a time, however, multiple faults can happen in a real case scenario. For example, a component can degrade as well as the associated sensors. Future work can include multiple fault separations and classifications.

9.3.2 Medium-term

Medium-term research that can be undertaken is fault prognosis. Predicting the failure of the equipment is critical in order to enhance maintenance activities and to reduce costs. Further work can be, for example, gathering more data and contextual system knowledge for real time failure rate distribution parameter analysis (often Weibull, normal, exponential, "Bath-tub curve" distribution), or estimating indicators such as remaining useful life, mean time to failure, or the current state of equipment.

Additionally, medium-term research that could be undertaken is cost-benefit analysis. Whilst methods applied for the three cases studies show good performance measure, there is a need to evaluate implementation costs as well as underlying costs generated by the false alarms, and then compare it with current maintenance activity costs in such a system. Implementation costs include sensors, transmissions, servers, and related costs, whereas false alarms may generate unnecessary maintenance or unseen fault costs.

Furthermore, with real time equipment life estimation, online maintenance costs can be assessed with optimal spare part orders, replacement quantities, and dates, as well as optimal maintenance activities.

9.3.3 Long-term

In line with the research aim and objectives, long-term research could address the need for building unified time series data-fusion tools that enable multi-streams analytics. The main weaknesses of traditional software tools such as Kafka, Spark, or Map-reduce are their inability to associate contextual knowledge and incorporate recent time-series analytics discoveries. Ontological modelling is key to combine analytical processes with domain knowledge for efficient decision making. Such time series tools should allow:

- Context/ontology modelling and association with the streaming.
- Structural analysis.
- Uncertainty handling (data level, analytics level, and knowledge level).
- Univariate, multivariate, multi-channel time series analysis.
- Various time series analytics utilisation including hybrid and ensemble learning approaches.
- A feedback system or human on the loop system to improve analytics performance over iteration.
- Online (real time, near time, automatic) as well as offline analysis.

REFERENCES

- Aggarwal, C.C. (2013) *Outlier Analysis*. New York, NY: Springer New York.
- Agrawal, R. and Srikant, R. (1994) 'Fast Algorithms for Mining Association Rules in Large Databases', *Journal of Computer Science and Technology*, 15(6), pp. 487–499.
- Ahuja, A.S., Ramteke, D.S. and Parey, A. (2020) 'Vibration-Based Fault Diagnosis of a Bevel and Spur Gearbox Using Continuous Wavelet Transform and Adaptive Neuro-Fuzzy Inference System', in , pp. 473–496.
- Anand, R. and Jeffrey, D.U. (2011) 'Mining of massive datasets', 2011-01-03]. [Http://Infolab. Stanford](http://Infolab.Stanford)
- Aronson, K.E., Murmanskyy, B.E., Murmanskii, I.B. and Brodov, Y.M. (2020) 'An expert system for diagnostics and estimation of steam turbine components' condition', *International Journal of Energy Production and Management*, 5(1), pp. 70–81.
- Arthur, D. and Vassilvitskii, S. (2007) 'K-Means++: The Advantages of Careful Seeding', *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. USA: Society for Industrial and Applied Mathematics. SODA '07, pp. 1027–1035.
- Backurs, A. and Tzamos, C. (2016) 'Improving Viterbi is Hard: Better Runtimes Imply Faster Clique Algorithms'
- Barna, G. (2011) 'Control System of Wheel Slide Protection Devices for Rail Vehicles Meeting the Requirements of European Normative Documents System Sterowania Układów', *Proceedings of 13th International Conference on Quality, Safety and Ecology in Transport*, , pp. 11–18.
- Bedworth, M. and O'Brien, J. (2000) 'Omnibus model: a new model of data fusion?', *IEEE Aerospace and Electronic Systems Magazine*, 15(4), pp. 30–36.
- Beggel, L., Kausler, B.X., Schiegg, M., Pfeiffer, M. and Bischl, B. (2018) 'Time series anomaly detection based on shapelet learning', *Computational Statistics*
- Belyaev, M.E., Gerasimov, D.N., Rymalis, M.R. and Semenov, S.A. (2018) 'Design of Diesel Engine Mathematical Model Oriented to Speed Control', *Journal of Computer and Systems Sciences International*, 57(4), pp. 626–639.

- Bishop, C.M. (2013) *Pattern Recognition and Machine Learning*.
- Bosso, N., Gugliotta, A., Magelli, M., Oresta, I.F. and Zampieri, N. (2019) 'Study of wheel-rail adhesion during braking maneuvers', *Procedia Structural Integrity*, 24, pp. 680–691.
- Boyd, J.R. (1976) 'Destruction and creation', *U.S. Army Command and General Staff College*
- Boyd, S. (2010) 'Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers', *Foundations and Trends® in Machine Learning*, 3(1), pp. 1–122.
- British Standards Institution (2010) '*BSI Standards Publication Maintenance — Maintenance terminology*', , p. 36.
- Cai, C., Zong, H. and Zhang, B. (2016) 'Ship diesel engine fault diagnosis based on the SVM and association rule mining', *Proceedings of the 2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2016*. IEEE, pp. 400–405.
- Chawda, R.K. and Thakur, G. (2016) 'Big data and advanced analytics tools', *2016 Symposium on Colossal Data Analysis and Networking, CDAN 2016*.
- Chen, J., Roberts, C. and Weston, P. (2008) 'Fault detection and diagnosis for railway track circuits using neuro-fuzzy systems', *Control Engineering Practice*, 16(5), pp. 585–596.
- CHEN, S.-X., NI, Y.-Q., LIU, J.-C. and YAO, N. (2019) 'Deep Learning-based Data Anomaly Detection in Rail Track Inspection', *Structural Health Monitoring 2019*. Lancaster, PA: DEStech Publications, Inc.
- Chiang, L.H., Russell, E.L. and Braatz, R.D. (2001) *Fault Detection and Diagnosis in Industrial Systems*. Springer-Verlag London.
- Chiatti, G., Chiavola, O. and Recco, E. (2014) 'Combustion diagnosis via block vibration signal in common rail diesel engine', *International Journal of Engine Research*, 15(6), pp. 654–663.
- Cui, S. and Sun, L. (2019) 'Sensor Fault Detection and Isolation of Electromechanical Actuator Based on Structural Residual Parity Space', in , pp. 2574–2585.

- Cui, X. and Lv, J. (2019) 'Design of Fault Diagnosis System for Balise Cable Based on Machine Learning', in , pp. 49–53.
- Danaher, P., Wang, P. and Witten, D.M. (2014) 'The joint graphical lasso for inverse covariance estimation across multiple classes', *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 76(2), pp. 373–397.
- Darong, H., Lanyan, K., Xiaoyan, C., Ling, Z. and Bo, M. (2018) 'Fault diagnosis for the motor drive system of urban transit based on improved Hidden Markov Model', *Microelectronics Reliability*, 82, pp. 179–189.
- Dassanayake, H., Roberts, C., Goodman, C.J. and Tobias, A.M. (2009) 'Use of parameter estimation for the detection and diagnosis of faults on electric train door systems', *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 223(4), pp. 271–278.
- Daugman, J.G. (1985) 'Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters', *Journal of the Optical Society of America A*, 2(7), p. 1160.
- Dong, J., Zhuang, D., Huang, Y. and Fu, J. (2009) 'Advances in multi-sensor data fusion: Algorithms and applications', *Sensors*, 9(10), pp. 7771–7784.
- Dong, M., He, D., Banerjee, P. and Keller, J. (2006) 'Equipment health diagnosis and prognosis using hidden semi-Markov models', *The International Journal of Advanced Manufacturing Technology*, 30(7–8), pp. 738–749.
- Dong, T., Zhang, F., Liu, B. and An, X. (2015) 'Model-Based State Feedback Controller Design for a Turbocharged Diesel Engine with an EGR System', *Energies*, 8(6), pp. 5018–5039.
- Dong, X.T., Nguyen, M.H. and Lee, D. (2020) 'Algorithm development for acoustic emission measurement in high-frequency ranges and its application on a large two-stroke marine diesel engine', *Measurement*, 162, p. 107907.
- Du, H., Li, Z., Chen, R., Yin, Z., Fu, Z., Zhang, Q., Xiao, X., Luo, M. and Bao, F. (2019) 'Dynamic Time Warping and Spectral Clustering Based Fault Detection and Diagnosis of Railway Point Machines', *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, pp. 595–600.

Dubuc, D., Sename, O., Bresch-Pietri, D., Halle, A. and Gauthier, C. (2018) 'Exhaust pressure LPV observer for turbocharged diesel engine on-board diagnosis', *IFAC-PapersOnLine*, 51(24), pp. 389–394.

Dykas, B. and Harris, J. (2017) 'Acoustic emission characteristics of a single cylinder diesel generator at various loads and with a failing injector', *Mechanical Systems and Signal Processing*, 93, pp. 397–414.

Esteban, J., Starr, A., Willetts, R., Hannah, P. and Bryanston-Cross, P. (2005) 'A Review of data fusion models and architectures: towards engineering guidelines', *Neural Computing and Applications*, 14(4), pp. 273–281.

Fassois, S.D. and Sakellariou, J.S. (2007) 'Time-series methods for fault detection and identification in vibrating structures', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851), pp. 411–448.

Fink, O., Zio, E. and Weidmann, U. (2014) 'Predicting component reliability and level of degradation with complex-valued neural networks', *Reliability Engineering and System Safety*, 121, pp. 198–206.

Flink, A. (2013) *Apache Flink: Introducing Flink Streaming*. Available at: <https://flink.apache.org/news/2015/02/09/streaming-example.html> (Accessed: 4 May 2018).

Fundation, A. (2016) *MapReduce Tutorial*. Available at: https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html (Accessed: 4 May 2018).

Gandomi, A. and Haider, M. (2015) 'Beyond the hype: Big data concepts, methods, and analytics', *International Journal of Information Management*, 35(2) Elsevier Ltd, pp. 137–144.

Garg, N. (2013) *Apache Kafka*. Available at: 10.1017/CBO9781107415324.004 (Accessed: 6 May 2018).

Garnham, J.E. and Davis, C.L. (2009) Rail materials *Wheel–Rail Interface Handbook*. Elsevier,

Gertler, J.J. (2017) *Fault detection and diagnosis in engineering systems*.

H2O (2016a) *Deep Learning with H2O*. Available at: <http://docs.h2o.ai/h2o->

- tutorials/latest-stable/tutorials/deeplearning/index.html?_ga=1.25172472.419461528.1466391851%0Ahttp://files/1239/index.html (Accessed: 4 May 2018).
- H2O (2016b) *H2O.ai*. Available at: <https://www.h2o.ai/h2o-old/machine-learning/> (Accessed: 4 May 2018).
- Hajam, P. (2014) *Mech diesel*. Available at: <https://mechdiesel.blogspot.com/> (Accessed: 9 August 2020).
- Hall, D.L. and James, L. (2001) *Handbook of Multisensor Data Fusion*. CRC Press.
- Hall, D.L. and Llinas, J. (1997) 'An introduction to multisensor data fusion', *Proceedings of the IEEE*, 85(1), pp. 6–23.
- Hallac, D., Vare, S., Boyd, S. and Leskovec, J. (2018) 'Toeplitz Inverse Covariance-based Clustering of Multivariate Time Series Data', *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. California: International Joint Conferences on Artificial Intelligence Organization, Vol.2018-July, pp. 5254–5258.
- Ham, S., Han, S.-Y., Kim, S., Park, H.J., Park, K.-J. and Choi, J.-H. (2019) 'A Comparative Study of Fault Diagnosis for Train Door System: Traditional versus Deep Learning Approaches', *Sensors*, 19(23), p. 5160.
- Hochreiter, S. and Schmidhuber, J. (1997) 'Long Short-Term Memory', *Neural Computation*, 9(8), pp. 1735–1780.
- Hong Li (1998) 'Modelling and analysis of a railway wheelset for active control', *UKACC International Conference on Control (CONTROL '98)*. IEE, Vol.1998, pp. 1289–1293.
- Hou, P.K., Shi, X.Z. and Lin, L.J. (2000) 'Generic blackboard based architecture for data fusion', *IECON Proceedings (Industrial Electronics Conference)*. IEEE, Vol.1, pp. 864–869.
- Hsieh, C.J., Sustik, M.A., Dhillon, I.S. and Ravikumar, P. (2011) 'Sparse inverse covariance matrix estimation using quadratic approximation', *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011*
- Hu, W., Liu, Z., Liu, D. and Hai, X. (2017) 'Fatigue failure analysis of high speed train

gearbox housings', *Engineering Failure Analysis*, 73, pp. 57–71.

Ince, T., Kiranyaz, S., Eren, L., Askar, M. and Gabbouj, M. (2016) 'Real-Time Motor Fault Detection by 1-D Convolutional Neural Networks', *IEEE Transactions on Industrial Electronics*, 63(11), pp. 7067–7075.

Indraratna, B. and Nimbalkar, S. (2013) 'Stress-strain degradation response of railway ballast stabilized with geosynthetics', *Journal of Geotechnical and Geoenvironmental Engineering*, 139(5), pp. 684–700.

Iwnicki, S. (2006) *Handbook of Railway Vehicle Dynamics*. Iwnicki, S. and Iwnicki, S. (eds.) CRC Press.

James, A., Jie, W., Xulei, Y., Chenghao, Y., Ngan, N.B., Yuxin, L., Yi, S., Chandrasekhar, V. and Zeng, Z. (2018) 'TrackNet - A Deep Learning Based Fault Detection for Railway Track Inspection', *2018 International Conference on Intelligent Rail Transportation (ICIRT)*. IEEE, pp. 1–5.

Jesussek, M. and Ellermann, K. (2014) 'Fault detection and isolation for a full-scale railway vehicle suspension with multiple Kalman filters', *Vehicle System Dynamics*, 52(12), pp. 1695–1715.

Ji, H., He, X., Shang, J. and Zhou, D. (2017) 'Incipient fault detection with smoothing techniques in statistical process monitoring', *Control Engineering Practice*, 62, pp. 11–21.

Jia, S. and Dhanasekar, M. (2007) 'Detection of rail wheel flats using wavelet approaches', *Structural Health Monitoring*, 6(2), pp. 121–131.

Jiao, J., Zhao, M., Lin, J. and Liang, K. (2020) 'A comprehensive review on convolutional neural network in machine fault diagnosis'

Jin, X., Fan, J. and Chow, T.W.S. (2019) 'Fault Detection for Rolling-Element Bearings Using Multivariate Statistical Process Control Methods', *IEEE Transactions on Instrumentation and Measurement*, 68(9), pp. 3128–3136.

Jones, N.B. and Li, Y.H. (2000) 'A review of condition monitoring and fault diagnosis for diesel engines', *Tribo Test*, 6(3), pp. 267–291.

Jung, D. and Sundstrom, C. (2017) 'A Combined Data-Driven and Model-Based Residual

Selection Algorithm for Fault Detection and Isolation', *IEEE Transactions on Control Systems Technology*, , pp. 1–15.

Keogh, E. and Ratanamahatana, C.A. (2005) 'Exact indexing of dynamic time warping', *Knowledge and Information Systems*, 7(3), pp. 358–386.

Khairy A.H.Kobbacy, D.N.P.M. (2008) *Complex System Maintenance Handbook*.Murthy, K. A. H. K. • D. N. P. (ed.) Engineering, Springer Series in Reliability.

Klein, L.A. (1999) *Sensor and Data Fusion Concepts and Applications*.

Kobayashi, T., Naganuma, Y. and Tsunashima, H. (2013) 'Condition monitoring of Shinkansen tracks based on inverse analysis', *Chemical Engineering Transactions*, 33, pp. 703–708.

Komalan, N. and Chauhan, A. (2017) 'A survey on cluster based routing, data aggregation and fault detection techniques for locating real time faults in modern metro train tracks using wireless sensor network', *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. IEEE, pp. 116–123.

Krishnaswami, V., Luh, G. and Rizzoni, G. (1995) 'Nonlinear parity equation based residual generation for diagnosis of automotive engine faults', *Control Engineering Practice*, 3(10), pp. 1385–1392.

Kumar, R. (2019) *Research Methodology: A step-by-step guide for beginners*.3rd editio. SAGE.

Lavielle, M. (1999) 'Detection of multiple changes in a sequence of dependent variables', *Stochastic Processes and their Applications*, 83(1), pp. 79–102.

Lee, H., Lee, J. and Sunwoo, M. (2014) 'Fault diagnosis of exhaust gas recirculation and variable geometry turbocharger systems in a passenger car diesel engine based on a sliding mode observer for air system states estimation', *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, 136(3)

Lee, H.H.Y. and Scott, D. (2009) 'Overview of maintenance strategy, acceptable maintenance standard and resources from a building maintenance operation perspective', *Journal of Building Appraisal*, 4(4), pp. 269–278.

Lee, J., Choi, H., Park, D., Chung, Y., Kim, H.-Y. and Yoon, S. (2016) 'Fault Detection

and Diagnosis of Railway Point Machines by Sound Analysis', *Sensors*, 16(4), p. 549.

Lewis, R. and Olofsson, U. (2009) *Wheel—rail interface handbook* Woodhead Publishing Limited,

Liang, B., Iwnicki, S., Feng, G., Ball, A., Tran, V.T. and Cattley, R. (2013) 'Railway wheel flat and rail surface defect detection by time-frequency analysis', *Chemical Engineering Transactions*, 33, pp. 745–750.

Lin-Hai, Z., Jian-Ping, W. and Yi-Kui, R. (2012) 'Fault diagnosis for track circuit using AOK-TFRs and AGA', *Control Engineering Practice*, 20(12), pp. 1270–1280.

Lin, C.-J., Chu, W.-L., Wang, C.-C., Chen, C.-K. and Chen, I.-T. (2019) 'Diagnosis of ball-bearing faults using support vector machine based on the artificial fish-swarm algorithm', *Journal of Low Frequency Noise, Vibration and Active Control*, , p. 146134841986182.

Liu, R., Yang, B., Zio, E. and Chen, X. (2018) 'Artificial intelligence for fault diagnosis of rotating machinery: A review', *Mechanical Systems and Signal Processing*, 108, pp. 33–47.

Llinas, J., Bowman, C., Rogova, G., Steinberg, A., Waltz, E. and White, F. (2004) 'Revisiting the JDL Data Fusion Model II',

Lu, C. and Shen, Y. (2020) 'Research on location detection of railway track spike based on neural network', *IOP Conference Series: Materials Science and Engineering*, 711, p. 012019.

Lu, S., He, Q., Hu, F. and Kong, F. (2014a) 'Sequential Multiscale Noise Tuning Stochastic Resonance for Train Bearing Fault Diagnosis in an Embedded System', *IEEE Transactions on Instrumentation and Measurement*, 63(1), pp. 106–116.

Lu, Y., Peng, Z., Miller, A., Zhao, T. and Johnson, C. (2014b) 'Timed fault tree models of the China Yongwen railway accident', *Proceedings - Asia Modelling Symposium 2014: 8th Asia International Conference on Mathematical Modelling and Computer Simulation, AMS 2014*. IEEE, pp. 128–133.

Lukoševičius, M. and Jaeger, H. (2009) 'Reservoir computing approaches to recurrent neural network training', *Computer Science Review*, 3(3), pp. 127–149.

- Manning, J. (2004) *Apache Storm*. Available at: <http://storm.apache.org/about/integrates.html> (Accessed: 4 May 2018).
- Maraini, D. and Nataraj, C. (2015) 'Freight Car Roller Bearing Fault Detection Using Artificial Neural Networks and Support Vector Machines', in , pp. 663–672.
- Martinez Guerra, R. and Mata Machuca, J.L. (2005) *Fault Detection and Diagnosis in Nonlinear Dynamical Systems*.
- Miao, Q. and Makis, V. (2007) 'Condition monitoring and classification of rotating machinery using wavelets and hidden Markov models', *Mechanical Systems and Signal Processing*, 21(2), pp. 840–855.
- Misiti, M., Oppenheim, G., Poggi, J.-M. and Misiti, Y. (2006) *Wavelets and Their Applications (Digital Signal and Image Processing Series)*. ISTE.
- Mobley, R.K. (2002) *An Introduction to Predictive Maintenance (Second Edition)*.
- Mohd Amiruddin, A.A.A., Zabiri, H., Taqvi, S.A.A. and Tufa, L.D. (2020) 'Neural network applications in fault diagnosis and detection: an overview of implementations in engineering-related systems', *Neural Computing and Applications*, 32(2), pp. 447–472.
- Mwangi, J.K., Lee, W.-J., Chang, Y.-C., Chen, C.-Y. and Wang, L.-C. (2015) 'An overview: Energy saving and pollution reduction by using green fuel blends in diesel engines', *Applied Energy*, 159, pp. 214–236.
- Namoano, B., Emmanouilidis, C., Ruiz-carcel, C. and Starr, A.G. (2020a) 'Change detection in streaming data analytics : A comparison of Bayesian online and martingale approaches (accepted)', 2020, C. 10-11 S. (ed.) *4th international workshop on Advances in Maintenance, Engineering, Services, and Technology: Maintenance in the Digital Era*. Cambridge 10-11 September 2020, pp. 1–8.
- Namoano, B., Starr, A., Emmanouilidis, C. and Cristobal, R.C. (2019) 'Online change detection techniques in time series: An overview', *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, pp. 1–10.
- Namoano, B., Starr, A., Emmanouilidis, C. and Ruiz Carcel, C. (2020b) *Doors datasets* Cranfield Online Research Data (CORD),
- Namoano, B., Starr, A., Emmanouilidis, C. and Ruiz Carcel, C. (2020c) *Engines datasets*

Cranfield Online Research Data (CORD),

Network, T. (2017) *Rail bounces back with record passenger numbers*. Available at: <https://www.transport-network.co.uk/Rail-bounces-back-with-record-passenger-numbers/15933> (Accessed: 29 October 2019).

Novikov, L. V. (2006) 'Modified wavelets and their applications', *Journal of Communications Technology and Electronics*, 51(11), pp. 1261–1270.

Nyberg, M. and Frisk, E. (2019) *Model Based Diagnosis of Technical Processes*. Linköping. Sweden: Course material, Linköpings Universitet.

Nyberg, R.G., Gupta, N.K., Yella, S. and Dougherty, M.S. (2014) 'Machine vision for condition monitoring vegetation on railway embankments', *IET Conference Publications*, 2014(CP631)

Olofsson, U. and Lewis, R. (2006) *Tribology of the Wheel – Rail Contact Handbook of Railway Vehicle Dynamics*. CRC Press,

P., D. and Ahmed, K. (2016) 'A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools', *International Journal of Advanced Computer Science and Applications*, 7(2)

Pascanu, R., Mikolov, T. and Bengio, Y. (2012) 'On the difficulty of training Recurrent Neural Networks', *30th International Conference on Machine Learning, ICML 2013*, (PART 3), pp. 2347–2355.

Patton, R.J. and Chen, J. (1991) 'A Review of Parity Space Approaches to Fault Diagnosis', *IFAC Proceedings Volumes*, 24(6), pp. 65–81.

Peng, W., Liu, E., Chen, L. and Zhai, D. (2019) 'Modeling and analysis of urban rail plug door system based on Petri net and SDG diagram', *MATEC Web of Conferences*, 272, p. 01037.

Pennacchi, P., Ricci, R., Chatterton, S. and Borghesani, P. (2011) 'Effectiveness of MED for fault diagnosis in roller bearings', in *Springer Proceedings in Physics*. , pp. 637–642.

Pestana-Viana, D., Gutiérrez, R.H.R., de Lima, A.A., e Silva, F.L., Vaz, L., de M. Prego, T. and Monteiro, U.A. (2019) 'Application of Machine Learning in Diesel Engines Fault Identification', in *Mechanisms and Machine Science*. , pp. 74–89.

Rabiner, L.R. (1989) 'A tutorial on hidden Markov models and selected applications in speech recognition', *Proceedings of the IEEE*, 77(2), pp. 257–286.

Railsystem (2015) *Sliding Plug Doors*. Available at: <http://www.railsystem.net/plug-doors/> (Accessed: 12 August 2020).

Rajagopalan, B. and Krovi, R. (2015) *Data Mining Algorithms Explained using R*.

Ren, D., Zeng, H., Wang, X., Pang, S. and Wang, J. (2020) 'Fault Diagnosis of Diesel Engine Lubrication System Based on Bayesian Network', *2020 6th International Conference on Control, Automation and Robotics, ICCAR 2020*. IEEE, pp. 423–429.

Roberts, C., Dassanayake, H.P.B., Lehasab, N. and Goodman, C.J. (2002) 'Distributed quantitative and qualitative fault diagnosis: Railway junction case study', *Control Engineering Practice*, 10(4), pp. 419–429.

Rowshandel, H., Nicholson, G.L., Shen, J.L. and Davis, C.L. (2018) 'Characterisation of clustered cracks using an ACFM sensor and application of an artificial neural network', *NDT & E International*, 98, pp. 80–88.

Ruiz-Carcel, C. and Starr, A. (2018) 'Data-Based Detection and Diagnosis of Faults in Linear Actuators', *IEEE Transactions on Instrumentation and Measurement*, 67(9), pp. 2035–2047.

Samé, A., Chamroukhi, F., Govaert, G. and Aknin, P. (2011) 'Model-based clustering and segmentation of time series with changes in regime', *Advances in Data Analysis and Classification*, 5(4), pp. 301–321.

Sammouri, W., Côme, E., Oukhellou, L. and Aknin, P. (2013) 'Mining Floating Train Data Sequences for Temporal Association Rules within a Predictive Maintenance Framework', in , pp. 112–126.

SAMOA (2017) *Apache SAMOA Documentation*. Available at: <https://samoa.incubator.apache.org/documentation/Home.html> (Accessed: 4 May 2018).

Sanfeliu, A., Andrade-Cetto, J., Barbosa, M., Bowden, R., Capitán, J., Corominas, A., Gilbert, A., Illingworth, J., Merino, L., Mirats, J.M., Moreno, P., Ollero, A., Sequeira, J. and Spaan, M.T.J. (2010) 'Decentralized sensor fusion for ubiquitous networking robotics in urban areas', *Sensors*, 10(3), pp. 2274–2314.

- Sankararaman, S. and Mahadevan, S. (2013) 'Bayesian methodology for diagnosis uncertainty quantification and health monitoring', *Structural Control and Health Monitoring*, 20(1), pp. 88–106.
- Santur, Y., Karakose, M. and Akin, E. (2018) 'An adaptive fault diagnosis approach using pipeline implementation for railway inspection', *TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES*, 26(2), pp. 987–998.
- Seryasat, O.R., Aliyari shoorehdeli, M., Honarvar, F. and Rahmani, A. (2010) 'Multi-fault diagnosis of ball bearing based on features extracted from time-domain and multi-class support vector machine(MSVM)', *2010 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, pp. 4300–4303.
- Shenton, M.J. (1985) 'Ballast Deformation and Track Deterioration.', in *TRACK TECHNOLOGY*. Thomas Telford Publishing, pp. 253–265.
- Shuai, L., Limin, J., Yong, Q., Bo, Y. and Yanhui, W. (2014) 'Research on Urban Rail Train Passenger Door System Fault Diagnosis Using PCA and Rough Set', *The Open Mechanical Engineering Journal*, 8(1), pp. 340–348.
- Simani, S. (2013) 'Residual generator fuzzy identification for automotive diesel engine fault diagnosis', *International Journal of Applied Mathematics and Computer Science*, 23(2), pp. 419–438.
- Spark (2017) *Configuration - Spark 2.3.0 Documentation*. Available at: <https://spark.apache.org/docs/latest/configuration.html#dynamic-allocation> (Accessed: 4 May 2018).
- Spark (2016) *Machine Learning Library (MLlib) Programming Guide - Spark 1.2.0 Documentation*. Available at: <https://spark.apache.org/docs/1.2.0/mllib-guide.html> (Accessed: 4 May 2018).
- Steinberg, A.N., Bowman, C.L. and White, F.E. (1999) 'Revisions to the JDL data fusion model', Dasarathy, B. V. (ed.) *Sensor Fusion: Architectures, Algorithms, and Applications III.*, Vol.3719, p. 430.
- Strano, S. and Terzo, M. (2019) 'Review on model-based methods for on-board condition monitoring in railway vehicle dynamics', *Advances in Mechanical Engineering*, 11(2), p. 168781401982679.

Taghizadeh-Alisaraei, A. and Mahdavian, A. (2019) 'Fault detection of injectors in diesel engines using vibration time-frequency analysis', *Applied Acoustics*, 143, pp. 48–58.

TensorFlow (2017) *TensorFlow*. Available at: <https://www.tensorflow.org/tutorials/> (Accessed: 6 May 2018).

Thomopoulos, S.C.A. (1990) 'Sensor Integration And Data Fusion', Schenker, P. S. (ed.) *Sensor Fusion II: Human and Machine Strategies.*, Vol.1198, p. 178.

Toliat, H.A., Abbaszadeh, K., Rahimian, M.M. and Olson, L.E. (2003) 'Rail defect diagnosis using wavelet packet decomposition', *IEEE Transactions on Industry Applications*, 39(5), pp. 1454–1461.

Trachi, Y., Elbouchikhi, E., Choqueuse, V. and Benbouzid, M.E.H. (2016) 'Induction Machines Fault Detection Based on Subspace Spectral Estimation', *IEEE Transactions on Industrial Electronics*, 63(9), pp. 5641–5651.

Triantaphyllou, E., Kovalerchuk, B., Mann, L. and Knapp, G.M. (1997) 'Determining the most important criteria in maintenance decision making', *Journal of Quality in Maintenance Engineering*, 3(1), pp. 16–28.

UIC (2005) *UIC Code 541-5 - Brakes - Specifications for the construction of various brake parts - Wheel Slide Protection device (WSP)*.3rd Editio.

Vagnoli, M., Remenyte-Prescott, R. and Andrews, J. (2018) 'Railway bridge structural health monitoring and fault detection: State-of-the-art methods and future challenges', *Structural Health Monitoring*, 17(4), pp. 971–1007.

Venkatasubramanian, V., Rengaswamy, R. and Kavuri, S.N. (2003) 'A review of process fault detection and diagnosis part II: Qualitative models and search strategies', *Computers and Chemical Engineering*, 27(3), pp. 313–326.

Venkatasubramanian, V., Rengaswamy, R., Yin, K. and Kavuri, S.N. (2003) 'A review of process fault detection and diagnosis: Part I: Quantitative model-based methods', *Computers & Chemical Engineering*, 27(3), pp. 293–311.

Verbert, K., Babuška, R. and De Schutter, B. (2017) 'Bayesian and Dempster–Shafer reasoning for knowledge-based fault diagnosis—A comparative study', *Engineering Applications of Artificial Intelligence*, 60, pp. 136–150.

- Vinnakota, B. (1996) 'Monitoring power dissipation for fault detection', *Proceedings of the IEEE VLSI Test Symposium*. IEEE Comput. Soc. Press, pp. 483–488.
- Wang, W., Wu, L., Gong, H., Fan, P., Wu, W., Zhou, Y. and Zhang, Z. (2019) 'Deformation Monitoring for High-Voltage Transmission Lines Using Sentinel-1A Data', *IOP Conference Series: Earth and Environmental Science*, 252(3), p. 032033.
- Wang, Y.W., Ni, Y.Q. and Wang, X. (2020) 'Real-time defect detection of high-speed train wheels by using Bayesian forecasting and dynamic model', *Mechanical Systems and Signal Processing*, 139, p. 106654.
- White, T. (2015) *Hadoop: The definitive guide*.
- Witten, D.M. and Tibshirani, R. (2009) 'Covariance-regularized regression and classification for high dimensional problems', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3), pp. 615–636.
- Woodfort, C. (2014) *Explain That Stuff*. Available at: <http://www.explainthatstuff.com/luminescence.html> (Accessed: 9 August 2020).
- Xian, G.-M. and Zeng, B.-Q. (2009) 'An intelligent fault diagnosis method based on wavelet packer analysis and hybrid support vector machines', *Expert Systems with Applications*, 36(10), pp. 12131–12136.
- Xie, S.F. and Li, Q.Z. (2002) 'Application of expert system based on mixing reasoning in traction substation fault diagnosis', *Proceedings - 2nd International Workshop on Autonomous Decentralized System, IWADS 2002*. IEEE Comput. Soc, pp. 229–232.
- Xie, Y., Liu, Y. and Valdar, W. (2016) 'Joint estimation of multiple dependent Gaussian graphical models with applications to mouse genomics', *Biometrika*, 103(3), pp. 493–511.
- Xin, G., Hamzaoui, N. and Antoni, J. (2018) 'Semi-automated diagnosis of bearing faults based on a hidden Markov model of the vibration signals', *Measurement*, 127, pp. 141–166.
- Xu, L. and Zhou, Y. (2016) 'Fault diagnosis for BLDCM system used FFT algorithm and support vector machines', *2016 IEEE International Conference on Aircraft Utility Systems (AUS)*. IEEE, pp. 384–387.

- Xu, M.H. and Shao, H. (2012) 'Solving the Matrix Nearness Problem in the Maximum Norm by Applying a Projection and Contraction Method', *Advances in Operations Research*, 2012, pp. 1–15.
- Xu, Q., Liu, Z. and Zhao, H. (2018) 'Method of Turnout Fault Diagnosis Based on Hidden Markov Model', *Tiedao Xuebao/Journal of the China Railway Society*, 40(8), pp. 98–106.
- Xu, X., Lu, N., Yong, J. and Jiang, B. (2016) 'Fault propagation analysis of IGBT fault in CRH5 traction system based on signed directed graph', *2016 Prognostics and System Health Management Conference (PHM-Chengdu)*. IEEE, pp. 1–6.
- Xu, X., Yan, X., Sheng, C., Yuan, C., Xu, D. and Yang, J. (2017) 'A Belief Rule-Based Expert System for Fault Diagnosis of Marine Diesel Engines', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, , pp. 1–17.
- Yan, J. and Lee, J. (2005) 'Degradation Assessment and Fault Modes Classification Using Logistic Regression', *Journal of Manufacturing Science and Engineering*, 127(4), pp. 912–914.
- Yang, T., Wu, L., Li, X., Zhu, M., Brennan, M.J. and Liu, Z. (2020) 'Active Vibration Isolation of a Diesel Generator in a Small Marine Vessel: An Experimental Study', *Applied Sciences*, 10(9), p. 3025.
- Yeh, C.-C.M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H.A., Zimmerman, Z., Silva, D.F., Mueen, A. and Keogh, E. (2018) 'Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile', *Data Mining and Knowledge Discovery*, 32(1), pp. 83–123.
- Yeh, C.C.M., Kavantzaz, N. and Keogh, E. (2017) 'Matrix profile VI: Meaningful multidimensional motif discovery', *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2017-Novem, pp. 565–574.
- Yin, S. and Huang, Z. (2015) 'Performance Monitoring for Vehicle Suspension System via Fuzzy Positivistic C-Means Clustering Based on Accelerometer Measurements', *IEEE/ASME Transactions on Mechatronics*, 20(5), pp. 2613–2620.
- Yoon, S., Sa, J., Chung, Y., Park, D. and Kim, H. (2016) 'Fault diagnosis of railway point machines using dynamic time warping', *Electronics Letters*, 52(10), pp. 818–819.
- Yuan, Z., Zhu, S. and Zhai, W. (2020) 'Damage Detection of Rail Fastening System

Through Deep Learning and Vehicle-Track Coupled Dynamics', in , pp. 148–153.

Zadeh, L.A. (1965) 'Fuzzy sets', *Information and Control*, 8(3), pp. 338–353.

Zhang, X., Grube, R., Shin, K.-K., Salman, M. and Conell, R.S. (2011) 'Parity-relation-based state-of-health monitoring of lead acid batteries for automotive applications', *Control Engineering Practice*, 19(6), pp. 555–563.

Zhang, X., Liang, Y., Zhou, J. and Zang, Y. (2015) 'A novel bearing fault diagnosis model integrated permutation entropy, ensemble empirical mode decomposition and optimized SVM', *Measurement*, 69, pp. 164–179.

Zhi-Ling, Y., Bin, W., Xing-Hui, D. and Hao, L. (2012) 'Expert System of Fault Diagnosis for Gear Box in Wind Turbine', *Systems Engineering Procedia*, 4, pp. 189–195.

Zhong, M., Song, Y. and Ding, S.X. (2015) 'Parity space-based fault detection for linear discrete time-varying systems with unknown input', *Automatica*, 59, pp. 120–126.

Zweiri, Y.H., Whidborne, J.F. and Seneviratne, L.D. (2001) 'Detailed analytical model of a single-cylinder diesel engine in the crank angle domain', *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 215(11), pp. 1197–1216.